



**ÜRETKEN YAPAY ZEKA DESTEKLİ
PROGRAMLAMA EĞİTİMİNİN ORTAOKUL
ÖĞRENCİLERİNİN BİLGİ İŞLEMSEL DÜŞÜNME
BECERİLERİ, PROGRAMLAMA BAŞARILARI VE
TUTUMLARINA ETKİSİ**

Volkan KAYA

Yüksek Lisans Tezi

**Bilgisayar ve Öğretim Teknolojileri Eğitimi Ana Bilim
Dalı**

2025

(Her hakkı saklıdır.)

T.C.
ATATÜRK ÜNİVERSİTESİ
EĞİTİM BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ ANA BİLİM DALI
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ BİLİM DALI

**ÜRETKEN YAPAY ZEKA DESTEKLİ PROGRAMLAMA EĞİTİMİNİN
ORTAOKUL ÖĞRENCİLERİNİN BİLGİ İŞLEMSEL DÜŞÜNME BECERİLERİ,
PROGRAMLAMA BAŞARILARI VE TUTUMLARINA ETKİSİ**

Investigating the Impact of Generative AI-Supported Programming Instruction on Middle School Students' Computational Thinking, Programming Performance, and Attitudes

YÜKSEK LİSANS TEZİ

Volkan KAYA

Danışman: Prof. Dr. Sevda KÜÇÜK

Erzurum
Mayıs, 2025

KABUL VE ONAY TUTANAĞI

Volkan Kaya tarafından hazırlanan “Üretken Yapay Zeka Destekli Programlama Eğitiminin Ortaokul Öğrencilerinin Bilgi İşlemsel Düşünme Becerileri, Programlama Başarıları Ve Tutumlarına Etkisi” başlıklı çalışması 29 / 04 / 2025 tarihinde yapılan tez savunma sınavı sonucunda başarılı bulunarak jürimiz tarafından Bilgisayar ve Öğretim Teknolojileri Ana Bilim Dalı, Bilgisayar ve Öğretim Teknolojileri Eğitimi Bilim Dalında yüksek lisans tezi olarak kabul edilmiştir.

Jüri Başkanı:	Prof. Dr. İrfan ŞİMŞEK <i>İstanbul Üniversitesi-Cerrahpaşa</i>	Aslı ıslak imzalıdır
Danışman:	Prof. Dr. Sevdâ KÜÇÜK <i>Atatürk Üniversitesi</i>	Aslı ıslak imzalıdır
Jüri Üyesi:	Dr. Öğr. Üyesi Ömer ARPACIK <i>Atatürk Üniversitesi</i>	Aslı ıslak imzalıdır

Bu tezin Atatürk Üniversitesi Lisansüstü Eğitim ve Öğretim Yönetmeliği'nin ilgili maddelerinde belirtilen şartları yerine getirdiğini onaylarım.

16 / 05 / 2025

Aslı ıslak imzalıdır

Prof. Dr. Refik DİLBER

Enstitü Müdürü

ETİK VE BİLDİRİM SAYFASI

Yüksek lisans Tezi olarak sunduğum “Üretken Yapay Zeka Destekli Programlama Eğitiminin Ortaokul Öğrencilerinin Bilgi İşlemsel Düşünme Becerileri, Programlama Başarıları Ve Tutumlarına Etkisi” başlıklı çalışmanın tarafımdan bilimsel etik ilkelere uyularak yazıldığını ve yararlandığım eserleri kaynakçada gösterdiğimi beyan ederim.

29/ 04 / 2025

Aslı ıslak imzalıdır

Volkan KAYA

☐ Tezle ilgili patent başvurusu yapılması / patent alma sürecinin devam etmesi sebebiyle Enstitü Yönetim Kurulunun/.../.... tarih ve sayılı kararı ile teze erişim 2 (iki) yıl süreyle engellenmiştir.

☐ Enstitü Yönetim Kurulunun/.../.... tarih ve sayılı kararı ile teze erişim 6 (altı) ay süreyle engellenmiştir.

TEŞEKKÜR

Bu tez çalışmasının ortaya çıkmasında bilgi birikimi, akademik rehberliği ve her aşamada sunduğu destekle yolumu aydınlatan çok değerli danışmanım Prof. Dr. Sevda Küçük'e en içten şükranlarımı sunarım. Kendisinin yönlendirmeleri, titiz değerlendirmeleri ve sabırlı yaklaşımı sayesinde bu çalışmayı akademik ölçütlere uygun şekilde tamamlayabildim. Ayrıca, çalışmanın gelişim sürecinde değerli katkılarda bulunan, görüş ve önerileriyle ufkumu genişleten Dr. Öğr. Üyesi Ömer Arpacık ve değerli jüri üyesi Prof. Dr. İrfan Şimşek'e teşekkür ederim. Akademik yolculuğum boyunca gösterdikleri ilgi, destek ve içtenlikleri için kendilerine minnettarım.

Araştırmanın uygulama sürecinde gönüllü olarak katılan sevgili öğrencilerime, araştırmaya gösterdikleri ilgi, samimiyet ve katkı için teşekkür ederim. Onların katılımı, bu çalışmanın anlamını ve değerini artırmıştır.

Bu süreçte yanımda olan, bana inanan, motivasyonumu kaybettiğim anlarda destekleriyle yeniden güç bulmamı sağlayan tüm sevdiklerime ve varlıklarıyla bana güç veren herkese sonsuz teşekkür ederim.

Bu tez, yalnızca akademik bir çalışma değil, aynı zamanda birlikte yürüdüğüm yol arkadaşlarımın, destekçilerimin ve inananların emeğini taşıyan bir eserdir.

Volkan KAYA

ÖZ

YÜKSEK LİSANS TEZİ

ÜRETKEN YAPAY ZEKA DESTEKLİ PROGRAMLAMA EĞİTİMİNİN ORTAOKUL ÖĞRENCİLERİNİN BİLGİ İŞLEMSEL DÜŞÜNME BECERİLERİ, PROGRAMLAMA BAŞARILARI VE TUTUMLARINA ETKİSİ

Volkan KAYA

Mayıs 2025, 152 Sayfa

Amaç: Bu çalışmanın amacı, bilişim teknolojileri ve yazılım dersinde üretken yapay zekâ destekli öğrenme modülünün kullanımının ortaokul öğrencilerinin bilgi işlemsel düşünme becerileri, programlama başarıları ve programlamaya yönelik tutumları üzerindeki etkilerini belirlemek ve bu süreçte elde edilen deneyimleri detaylı bir şekilde incelemektir.

Yöntem: Araştırma, karma araştırma yöntemi kapsamında açıklayıcı sıralı desen kullanılarak gerçekleştirilmiştir. Çalışmanın nicel boyutu, ön test-son test kontrol gruplu yarı deneysel desen ile yürütülmüş, nitel boyutu ise durum çalışması yaklaşımına dayandırılmıştır. Araştırma örneklemini, Bitlis Merkez ilçesindeki bir ortaokulda öğrenim gören toplam 34 yedinci sınıf öğrencisi oluşturmuştur. Öğrenciler rastgele atama yöntemiyle deney (17 öğrenci) ve kontrol (17 öğrenci) gruplarına ayrılmıştır. Deney grubu öğrencileri, üretken yapay zekâ destekli öğrenme modülünü kullanarak Python programlama dilini öğrenmiş, kontrol grubu ise geleneksel yöntemlerle eğitim almıştır. Veri toplama aracı olarak Bilgi İşlemsel Düşünme Öz-Değerlendirme Ölçeği, Bebras Bilgi İşlemsel Düşünme Testi, Programlama Başarı Testi, Programlamaya Yönelik Tutum Ölçeği ve Yarı Yapılandırılmış Görüşme Formu kullanılmıştır. Nicel veriler, bağımsız ve bağımlı örneklem t-testleri ile analiz edilmiş; nitel veriler ise betimsel ve içerik analizi yöntemleriyle değerlendirilmiştir.

Bulgular: Araştırma bulgularına göre, deney grubundaki öğrencilerin problem çözme becerilerinde anlamlı bir artış gözlemlenirken, kontrol grubunda bu artış anlamlı değildir; ancak iki grubun son test puanları arasında anlamlı bir fark bulunmamıştır. Her iki grupta da bilgi işlemsel düşünme becerileri gelişmiş, fakat son test puanları açısından yine anlamlı bir fark oluşmamıştır. Programlama başarıları açısından deney grubu lehine anlamlı bir artış görülmüştür. Programlamaya yönelik tutumlarda her iki grupta olumlu değişim yaşansa da aralarında anlamlı bir fark tespit edilmemiştir. Nitel bulgular ise öğrencilerin üretken yapay zekâ destekli etkinlikleri eğlenceli bulduklarını ve öğrenme sürecine olumlu katkılar sağladığını ifade ettiklerini ortaya koymuştur.

Sonuçlar: Araştırma sonuçlarına göre, üretken yapay zekâ destekli öğrenme modüllerinin, öğrencilerin bilgi işlemsel düşünme, problem çözme becerileri ve programlama başarıları üzerinde olumlu etkiler yarattığı görülmüştür.

Anahtar Kelimeler: üretken yapay zekâ, programlama eğitimi, problem çözme becerileri, bilgi işlemsel düşünme, ortaokul öğrencileri

ABSTRACT

MASTER'S THESIS

INVESTIGATING THE IMPACT OF GENERATIVE AI-SUPPORTED PROGRAMMING INSTRUCTION ON MIDDLE SCHOOL STUDENTS' COMPUTATIONAL THINKING, PROGRAMMING PERFORMANCE, AND ATTITUDES

Volkan KAYA

May 2025, 152 Pages

Purpose: The aim of this study is to examine the effects of a generative artificial intelligence (AI)-supported learning module and traditional programming methods on middle school students' computational thinking skills, problem-solving skills, programming achievements, and attitudes towards programming. Additionally, the study aims to analyze the experiences gained during this process in detail.

Method: The research was conducted using a sequential explanatory design within the framework of a mixed methods approach. The sample consisted of 34 seventh-grade students from a middle school in Bitlis, Turkey. Students were randomly assigned to experimental (17 students) and control (17 students) groups. The experimental group used a generative AI-supported learning module to learn Python programming, while the control group received instruction through traditional methods. Data collection instruments included the Programming Achievement Test, the Computational Thinking Self-Assessment Scale, the Attitude Towards Programming Scale, and Semi-Structured Interview Forms. Quantitative data were analyzed using independent and paired-sample t-tests, while qualitative data were evaluated using descriptive and content analysis methods.

Findings: According to the research findings, the experimental group showed a significant increase in problem-solving skills, while the control group did not; however, no significant difference was found between their post-test scores. Both groups improved in computational thinking skills, yet without a significant difference in outcomes. The experimental group achieved significantly higher programming success compared to the control group. Although both groups showed positive changes in attitudes toward programming, the difference was not significant. Qualitative data indicated that students found generative AI-supported activities enjoyable and helpful, and expressed a desire for their continuation.

Conclusions: According to the research results, the learning modules supported by generative artificial intelligence had positive effects on students' computational thinking, problem-solving skills, and programming achievement.

Keywords: generative artificial intelligence, programming education, problem-solving skills, computational thinking, middle school students

İÇİNDEKİLER

KABUL VE ONAY TUTANAĞI.....	iii
ETİK VE BİLDİRİM SAYFASI.....	ii
TEŞEKKÜR	iii
ÖZ.....	iv
ABSTRACT	v
İÇİNDEKİLER.....	vi
TABLolar DİZİNİ.....	viii
ŞEKİLLER DİZİNİ.....	ix
KISALTMALAR VE SİMGELER DİZİNİ.....	ix
BİRİNCİ BÖLÜM.....	1
Giriş	1
Araştırmanın Amacı	7
Araştırma Soruları.....	8
Araştırmanın Önemi ve Gerekçesi	8
Araştırmanın Sınırlılıkları	10
Varsayımlar	10
Terim ve Tanımlar.....	10
İKİNCİ BÖLÜM	12
Kuramsal Çerçeve ve İlgili Araştırmalar	12
Bilgi İşlemsel Düşünme	12
Programlama Eğitimi	18
Dünya’da Programlama Eğitimi	21
Avrupa’da Programlama Eğitimi.....	21
Amerika Birleşik Devletleri’nde Programlama Eğitimi	21
Asya’da Programlama Eğitimi	22
Türkiye’de Programlama Eğitimi	23
Yapay Zeka	24
Eğitimde Yapay Zeka Uygulamaları.....	25
Eğitimde Üretken Yapay Zeka.....	29

Üretken Yapay Zeka ile Programlama Eğitimi.....	32
İlgili Çalışmalar.....	37
ÜÇÜNCÜ BÖLÜM.....	42
Yöntem	42
Araştırma Yöntemi.....	42
Araştırma Deseni.....	42
Nicel Bölüm: Yarı Deneysel Desen	43
Nitel Bölüm: Durum Çalışması.....	44
Katılımcılar	44
Evren	44
Örnekleme.....	44
Veri Toplama Araçları	46
Bilgi İşlemsel Düşünme Becerisi	46
Bilge Kunduz ve Bebras Testi.....	46
Bilgi İşlemsel Düşünme Öz Değerlendirme Ölçeği.....	47
Programlamaya Yönelik Tutum Ölçeği	48
Programlama Başarı Testi (Rubrik)	50
Yarı Yapılandırılmış Görüşme Formu	51
Uygulama	52
Analiz Basamağı	52
Tasarım Basamağı.....	53
Geliştirme Basamağı	64
Uygulama Basamağı	65
Değerlendirme Basamağı	67
Nitel Veri Toplama ve Sonuçlandırma	68
Nitel Veri Toplama Süreci	68
Verilerin Analizi.....	69
Nicel Verilerin Analizi	69
Nitel Verilerin Analizi.....	70
Araştırmacı Rolü	72
Geçerlik ve Güvenirlik.....	73
Geçerlik.....	73
Güvenirlik	74
DÖRDÜNCÜ BÖLÜM	75
Bulgular	75

Normallik Analizi Sonuçları	75
Bilgi İşlemsel Düşünmeye İlişkin Bulgular	76
Deney Grubu Bilgi İşlemsel Düşünme Ön-test Son-test Puanlarının Karşılaştırılması..	76
Kontrol Grubu Bilgi İşlemsel Düşünme Ön Test Son Test Verilerine İlişkin t-testi Puanları	77
Deney ve Kontrol Gruplarının Bilgi İşlemsel Düşünme Son Test Puanlarının Karşılaştırılması	79
Bebras Testine İlişkin Bulgular.....	80
Deney Grubu Bebras Ön Test Son Test Puanlarının Karşılaştırılması	80
Kontrol Grubu Bebras Ön Test Son Test Puanlarının Karşılaştırılması	81
Deney ve Kontrol Gruplarının Bebras Son Test Puanlarının Karşılaştırılması	82
Programlamaya Yönelik Tutum Ölçeğine İlişkin Bulgular	83
Deney Grubu Programlamaya Yönelik Tutum Ön Test Son Test Puanlarının Karşılaştırılması	84
Kontrol Grubu Programlamaya Yönelik Tutum Ön Test Son Test Puanlarının Karşılaştırılması	85
Deney ve Kontrol Gruplarının Programlamaya Yönelik Tutum Son Test Puanlarının Karşılaştırılması	86
Programlama Başarı Testine İlişkin Bulgular	87
Deney ve Kontrol Gruplarının Başarı Son Test Puanlarının Karşılaştırılması	87
ÜYZ Tabanlı Programlama Modülünün Öğrenciler Üzerindeki Etkilerine İlişkin Nicel ve Nitel Bulguların Entegre Edilmiş Sonuçları.....	88
ÜYZ Tabanlı Programlama Modülünün Programlamaya Yönelik Etkisine İlişkin Karma Yöntem Entegre Sonuçları	93
Programlama Başarısı Teması.....	96
BEŞİNCİ BÖLÜM	98
Tartışma ve Sonuç	98
Öğrencilerinin Bilgi İşlemsel Düşünmelerine Yönelik Sonuçları ve Tartışma	98
Bebras Bilge Kunduz Testi Sonuçlarına İlişkin ve Tartışma	101
Öğrencilerinin Programlamaya Yönelik Tutumlarına İlişkin Sonuçları ve Tartışma ...	103
Programlama Başarı Testine İlişkin Sonuç ve Tartışma	105
Genel Sonuç ve Değerlendirme	107
Öneriler.....	108
Uygulayıcılara (Eğitimcilere) Yönelik Öneriler	108
Gelecek Araştırmalara Yönelik Öneriler	109

Geliştiricilere ve Sistem Tasarımcılarına Yönelik Öneriler	109
KAYNAKÇA	111
EKLER	120
EK-1. Programlama Başarı Testi	120
EK-2. Bilge Kunduz Bebras Testi.....	122
EK-3. Bilgi İşlemsel Düşünme Öz Değerlendirme Ölçeği	124
EK-4. Programlamaya Yönelik Tutum Ölçeği.....	125
EK-5. Yarı Yapılandırılmış Görüşme Formu	126
EK-6. Bireysel Bilgi Formu	127
EK-7. Tanıtım Materyalleri.....	128
EK-8. Etkinlik Yaprakları	129
EK-9. Etik Kurul İzni.....	131
EK-10. İl Milli Eğitim Müdürlüğü Uygulama İzin Belgesi	132
ÖZ GEÇMİŞ.....	137

TABLÖLAR DİZİNİ

Tablo 1. <i>İlgili Araştırmalar</i>	37
Tablo 2. <i>Programlama Başarı Testi Soru Kazanım Tablosu</i>	51
Tablo 3. <i>Uygulama Süreci Kazanım ve Etkinlik Tablosu</i>	66
Tablo 4. <i>Değişkenlerin Normallik Analizi Puanları</i>	75
Tablo 5. <i>Deney Grubu Bilgi İşlemsel Düşünme Ön-test Son-test Puanlarına İlişkin t-testi Puanları</i>	76
Tablo 6. <i>Kontrol Grubu Bilgi İşlemsel Düşünme Ön Test Son Test Verilerine İlişkin t-testi Puanları</i>	77
Tablo 7. <i>Deney ve Kontrol Gruplarının Bilgi İşlemsel Düşünme Son Test Puanları</i>	79
Tablo 8. <i>Deney Grubu Bebras Ön Test Son Test Puanları</i>	81
Tablo 9. <i>Kontrol Grubu Bebras Ön Test Son Test Puanları</i>	82
Tablo 10. <i>Deney ve Kontrol Gruplarının Bebras Son Test Puanları</i>	83
Tablo 11. <i>Deney Grubu Programlamaya Yönelik Tutum Ön Test Son Test Puanları</i>	84
Tablo 12. <i>Kontrol Grubu Programlamaya Yönelik Tutum Ön Test Son Test Puanları</i>	85
Tablo 13. <i>Deney ve Kontrol Gruplarının Programlamaya Yönelik Tutum Son Test Puanları</i>	86
Tablo 14. <i>Deney ve Kontrol Gruplarının Programlama Başarı Son Test Puanları</i>	87
Tablo 15. <i>BİD Algoritmik Düşünme Alt Boyutu Öğrenci Görüşleri</i>	89
Tablo 16. <i>BİD Paralleştirme Alt Boyutu Öğrenci Görüşleri</i>	90
Tablo 17. <i>BİD Ayırıştırma Alt Boyutu Öğrenci Görüşleri</i>	91
Tablo 18. <i>BİD Soyutlama Alt Boyutu Öğrenci Görüşleri</i>	92
Tablo 19. <i>BİD Otomasyon Alt Boyutu Öğrenci Görüşleri</i>	93
Tablo 20. <i>Programlamaya Yönelik Tutum İstek Alt Boyutu Öğrenci Görüşleri</i>	94
Tablo 21. <i>Programlamaya Yönelik Tutum İnanç Alt Boyutu Öğrenci Görüşleri</i>	95
Tablo 22. <i>Programlamaya Yönelik Tutum İlgi Alt Boyutu Öğrenci Görüşleri</i>	96
Tablo 23. <i>Programlama Başarısı Öğrenci Görüşleri</i>	97

ŞEKİLLER DİZİNİ

Şekil 1. Bilgi İşlemsel Düşünmenin Alt Boyutları	14
Şekil 2. Araştırmanın Bütüncül Yöntem Süreci	43
Şekil 3. ADDIE Modeli Basamakları	52
Şekil 4. Öğrenci Giriş Sayfası	56
Şekil 5. Öğrenci Paneli Sayfası	57
Şekil 6. Python Kod Çalıştır Sayfası	58
Şekil 7. Python Kod Çalıştır Sayfası	58
Şekil 8. ÜYZ İle Kod Yazma Arayüzü Sayfası	59
Şekil 9. ÜYZ'ye Kod Sor Sayfası	60
Şekil 10. ÜYZ'ye Kod Sor Sayfası	60
Şekil 11. ÜYZ İle Kod Açıklaması Al Sayfası	61
Şekil 12. ÜYZ İle Kod Açıklaması Al Sayfası	62
Şekil 13. ÜYZ İle Hata Bul Sayfası	63
Şekil 14. ÜYZ İle Hata Bul Sayfası	63

KISALTMALAR VE SİMGELER DİZİNİ

BİD: Bilgi İşlemsel Düşünme

BTY: Bilişim Teknolojileri ve Yazılımı

MEB: Milli Eğitim Bakanlığı

PBT: Programlama Başarı Testi

PÇB: Problem Çözme Becerileri

ÜYZ: Üretken Yapay Zeka



BİRİNCİ BÖLÜM

Giriş

Günümüzde, dijital dönüşüm sürecinin hızlanmasıyla birlikte programlama eğitimi, bireylerin problem çözme, analitik düşünme ve algoritmik beceriler geliştirmesi için kritik bir öneme sahip olmuştur (Wing, 2006). Bu becerilerin erken yaşlardan itibaren kazandırılması, bireylerin gelecekteki akademik, mesleki ve sosyal yaşamlarına güçlü bir temel oluşturacaktır. Eğitimde yenilikçi yaklaşımların artmasıyla, yapay zeka (YZ) teknolojilerinin öğrenme süreçlerine entegrasyonu, programlama eğitiminin hem etkililiğini hem de erişilebilirliğini artırma potansiyeline sahiptir. (Ertmer vd., 2010). Özellikle, üretken yapay zeka (ÜYZ) araçları, öğrencilerin öğrenme deneyimlerini kişiselleştirme, kodlama süreçlerini kolaylaştırma ve hata tespitinde rehberlik etme gibi özellikleriyle programlama eğitiminde yeni bir dönemi başlatmaktadır. Ancak, bu alandaki uygulamaların ortaokul düzeyinde etkili bir şekilde hayata geçirilmesine yönelik araştırmalar sınırlıdır. Bu tez, ortaokul düzeyinde programlama eğitimi bağlamında üretken yapay zeka destekli bir öğrenme modülünün öğrencilerin bilgi işlemsel düşünme becerilerini, programlama başarıları ve tutumlarının değerlendirilmesini hedeflemektedir.

Ortaokul düzeyinde programlama eğitimi, öğrencilerin problem çözme, yaratıcı düşünme ve üst düzey bilişsel beceriler kazanmasını sağlamaktadır (Ekinci vd., 2023). Kodlama eğitiminde bilgi işlemsel düşünme becerilerinin gelişimi, öğrencilerin algoritmik düşünme, işbirlikçi çalışma ve eleştirel düşünme yeteneklerini desteklemektedir. Ayrıca, görsel tabanlı blok programlama araçlarının kullanımı, öğrenim sürecini daha erişilebilir ve motive edici hale getirmektedir (Horn & Bers, 2019). Bunun yanı sıra, araştırmalar, kodlama eğitiminin öğrencilerin derslere yönelik tutumlarını ve bilişim teknolojilerine karşı ilgilerini olumlu yönde etkilediğini ortaya koymaktadır (Erol, 2015; Çavdar vd., 2022).

Programlama eğitimi, bireylerin yalnızca yazılım geliştirme becerilerini değil, aynı zamanda problem çözme, eleştirel düşünme ve algoritmik düşünme gibi üst düzey bilişsel becerilerini geliştirmek için temel bir araçtır. Eğitimde programlama öğretimi, dijitalleşen dünyanın taleplerine cevap verebilecek bireyler yetiştirme açısından önemli bir role sahiptir. Bu süreçte, öğrencilerin analitik düşünme becerilerini geliştirmek, algoritmik yapıların temel prensiplerini anlamalarını sağlamak ve teknolojiyi yaratıcı bir şekilde kullanmalarına olanak tanımak amaçlanmaktadır (Wing, 2006).

Programlama eğitimi, çağımızın dijital dönüşüm sürecinde önemli bir yer tutmaktadır. Özellikle bilgi ve iletişim teknolojilerinin hızla gelişmesi, programlama bilgisine sahip bireylerin iş gücü piyasasında daha fazla talep görmesine neden olmaktadır. Programlama eğitimi, öğrencilere analitik düşünme ve problem çözme becerileri kazandırmaktadır. Bunun yanı sıra, programlama öğrenen bireylerin yaratıcı düşünme kapasiteleri artırmakta ve karmaşık problemleri daha etkili bir şekilde çözebilmektedirler (Lee vd., 2011). Ayrıca, programlama eğitiminin erken yaşlarda başlaması, öğrencilerin dijital okuryazarlık seviyelerini yükseltmektedir. Bu durum, onların gelecekteki akademik ve mesleki başarılarını olumlu yönde etkilemektedir (Çetin & Sönmez, 2009). Ek olarak, programlama eğitimi yaratıcı düşünme ve yenilikçi çözümler üretme kapasitesini artırmaktadır. Öğrenciler, programlama projeleri aracılığıyla kendi fikirlerini hayata geçirebilmekte ve bu süreçte yaratıcılıklarını geliştirebilmektedirler (Resnick vd., 2009). Bu durum, geleceğin inovasyon liderlerinin yetişmesine katkı sağlamaktadır. Programlama eğitiminin bir diğer önemli faydası, iş gücü piyasasında bireylere sağladığı avantajlardır. Programlama bilgisi, bireylerin iş bulma ve kariyerlerinde ilerleme şanslarını artıracaktır. Dijital dünyada, programlama bilgisine sahip bireyler daha fazla talep görmekte ve daha yüksek maaşlar almaktadır. Ayrıca, programlama bilgisi, bireylerin girişimcilik yeteneklerini de geliştirecek ve yeni iş fikirleri oluşturma kapasitesini artıracaktır (Durak vd., 2017).

Erken yaşta programlama öğretimi, bireylerin yaratıcı düşünme, problem çözme ve üst düzey bilişsel beceriler geliştirme süreçlerinde kritik bir araç olarak değerlendirilmektedir (Durak vd., 2017). Araştırmalar, özellikle K12 düzeyindeki öğrenciler arasında programlama eğitiminin etkisini artıran araç ve yöntemlerin kullanımına odaklanmaktadır. Bu bağlamda, Scratch gibi blok tabanlı programlama ortamlarının, öğrencilerin hem öğrenme motivasyonunu artırmada hem de bilişsel bariyerleri azaltmada etkili olduğu ifade edilmektedir.

Araştırmalar, programlama eğitiminin etkili olabilmesi için pedagojik yaklaşımların ve araçların büyük önem taşıdığını göstermektedir (Bers vd., 2014). Özellikle blok tabanlı programlama araçları (Scratch, Blockly gibi), çocukların programlama öğrenimini daha erişilebilir ve eğlenceli hale getirirken, geleneksel metin tabanlı programlama dillerine kıyasla öğrenme sürecindeki engelleri azaltmaktadır (Grover & Pea, 2013). Bunun yanı sıra, FeTeMM (STEM) odaklı eğitim programlarında programlama eğitimi, disiplinlerarası düşünmeyi teşvik etmekte ve öğrencilerin bilim, teknoloji, mühendislik ve matematik alanlarındaki becerilerini artırmaktadır (Bers vd., 2014).

Programlama eğitimi yalnızca teknik beceriler kazandırmakla kalmamakta, aynı zamanda bireylerin sosyal becerilerini ve işbirliği yeteneklerini de geliştiren bir ortam

sunmaktadır. Örneğin, projeye dayalı öğrenme ve grup çalışmaları gibi yöntemlerle öğrenciler, programlama yoluyla karmaşık problemleri ekip çalışmasıyla çözmeyi öğrenmektedirler (Papert, 1980). Bu özellikleriyle programlama eğitimi, bireylerin hem bireysel hem de toplumsal gelişimine katkı sağlamaktadır.

Programlama eğitimi, bilgi-işlemsel düşünme becerilerinin kazandırılması ve bireylerin dijital çağın gerekliliklerine uyum sağlaması amacıyla dünya genelinde giderek daha fazla önem kazanmaktadır. 21. yüzyılda birçok ülke, eğitim müfredatlarında programlama derslerine yer vererek, erken yaşlardan itibaren bireylerin teknoloji okuryazarlığını artırmayı hedeflemiştir (Balanskat & Engelhardt, 2015). Örneğin, İngiltere 2014 yılında, "Kodlama Yılı" ilan ederek, ilkokuldan itibaren programlama eğitimini zorunlu hale getirmiştir. Bu kapsamda, Scratch ve Python gibi araçlar ile algoritmik düşünme ve problem çözme becerileri geliştirilmesi hedeflenmiştir (Hubwieser vd., 2015). Günümüzde hızla gelişen teknolojinin etkisiyle, programlama eğitimi küresel ölçekte önem kazanmıştır. Birçok ülke, bilişim teknolojilerinin gerektirdiği yazılım geliştirme becerilerini bireylere kazandırmak amacıyla müfredatlarında köklü değişiklikler yapmıştır. Örneğin, Estonya 2012 yılında ilkokuldan itibaren programlama eğitimini zorunlu hale getirirken, Güney Kore 2017 yılından itibaren aşamalı olarak yazılım derslerini eğitim sistemine entegre etmiştir (Demirer & Sak, 2016)

Avrupa Birliği'nin başlattığı "Avrupa Kod Haftası" etkinlikleri, programlama eğitiminin yaygınlaştırılmasında önemli bir rol oynamaktadır. Bu çerçevede düzenlenen yüzlerce çalıştayda binlerce öğrenciye ulaşılmıştır. Amerika Birleşik Devletleri'nde ise "Hour of Code" gibi girişimler programlama eğitiminin herkes için erişilebilir olmasını sağlamayı amaçlamıştır (Demirer & Sak, 2016). Bu tür etkinlikler ve projeler, programlama eğitiminin yalnızca teknik beceriler kazandırmanın ötesinde, problem çözme, analitik düşünme ve yaratıcılık gibi üst düzey bilişsel becerilerin gelişimine de katkı sağladığını göstermektedir. Bununla birlikte, görsel programlama araçlarının kullanımı, programlama eğitiminin daha erişilebilir hale getirilmesinde etkili olmuştur. Scratch, Microsoft Small Basic ve Alice gibi araçlar, programlama becerilerinin eğlenceli ve kolay bir şekilde öğrenilmesini sağlayarak öğrencilerin teknolojiye yönelik ilgisini artırmıştır. Bu yaklaşımlar, öğrencilerin bireysel projeler geliştirme ve uygulama tasarımı yapma gibi beceriler kazanmasına olanak tanımaktadır.

Amerika Birleşik Devletleri, Code.org gibi platformlar aracılığıyla programlama eğitimini yaygınlaştırma konusunda öncü olmuştur. Bu girişimler, öğrencilerin bilgisayar bilimlerine yönelik ilgisini artırmayı ve özellikle dezavantajlı grupların bu alanlara erişimini sağlamayı amaçlamaktadır (Grover & Pea, 2013). Ayrıca, STEM odaklı programlar aracılığıyla öğrencilerin disiplinlerarası beceriler kazanması desteklenmiştir.

Avrupa'da, Finlandiya ve Estonya gibi ülkeler, programlama eğitiminde öncü uygulamalar geliştirmiştir. Finlandiya, 2016 yılında ulusal müfredatında programlama derslerini zorunlu hale getirerek, algoritmik düşünmeyi temel eğitim hedeflerinden biri olarak benimsemiştir (Heintz vd., 2017). Estonya, ProgeTiger adlı bir girişim başlatarak, ilkokuldan itibaren öğrencilere programlama ve robotik beceriler kazandırmaya odaklanmıştır.

Asya ülkeleri de programlama eğitimine büyük önem vermektedir. Örneğin, Japonya 2020 yılında programlama eğitimini zorunlu ders olarak müfredatına eklemiş ve öğrencilerin yapay zeka ve otomasyon teknolojilerini anlamalarını sağlamayı hedeflemiştir (Nakajima, 2020). Bu örnekler, dünya genelinde programlama eğitiminin sadece teknik beceriler kazandırmanın ötesine geçtiğini ve öğrencilerin yaratıcı düşünme, problem çözme ve teknolojiye yönelik olumlu tutumlar geliştirmesine odaklandığını göstermektedir. Ayrıca, küresel düzeyde eğitim politikalarının, ekonomik büyüme ve inovasyonu desteklemek amacıyla programlama eğitimi stratejik bir araç olarak gördüğü anlaşılmaktadır.

Türkiye'de programlama eğitimi, Millî Eğitim Bakanlığı'nın (MEB) son yıllarda yürürlüğe koyduğu yeni müfredat çalışmalarıyla birlikte, özellikle ortaokul düzeyinde Bilişim Teknolojileri ve Yazılım dersi kapsamında sistematik olarak uygulanmaya başlanmıştır. Bu derslerde öğrencilerin algoritmik düşünme, problem çözme ve temel programlama becerilerini geliştirmeleri hedeflenmektedir (MEB, 2018). Uygulamalarda öncelikli olarak Scratch gibi blok tabanlı görsel programlama araçları tercih edilmekte, öğrencilerin somut görseller üzerinden kodlama mantığını kavramaları sağlanmaktadır (Akgün & Topal, 2015). Bunun yanı sıra, 7. ve 8. sınıf seviyelerinde Python gibi metin tabanlı programlama dillerine de giriş yapılmakta, öğrencilerin gerçek kod yapılarıyla tanışmaları sağlanmaktadır (Korkmaz & Altun, 2017). Türkiye'deki bu yaklaşım, erken yaşlarda programlama eğitiminin yaygınlaştırılmasını ve dijital okuryazarlığın artırılmasını amaçlayan küresel eğilimlerle paralellik göstermektedir.

Metin tabanlı programlama eğitimi, öğrencilerin algoritmik düşünme ve problem çözme becerilerini geliştirmek için kritik bir araç olarak görülmektedir. Yapılan bir araştırmada, metin tabanlı programlamanın, özellikle temel programlama dillerini öğrenme sürecinde öğrencilerin analitik düşünme yeteneklerini geliştirmelerine olanak tanıdığı vurgulanmaktadır (Erümit vd., 2019). Metin tabanlı programlama, öğrencilerin sadece bir dilin sözdizimini öğrenmesini değil, aynı zamanda yazılım geliştirme süreçlerini anlamalarını ve uygulamalarını da teşvik etmektedir. Bu bağlamda, Python programlama dili, basit ve okunabilir sözdizimi ile yeni başlayanlar için ideal bir dil olarak öne çıkmaktadır. Python, özellikle web geliştirme, veri analizi, yapay zeka ve otomasyon gibi çeşitli alanlarda geniş bir kullanım yelpazesine sahiptir (Van Rossum & Drake, 2009). Ayrıca, Python'un geniş kütüphane desteği ve platform bağımsız

yapısı, öğrencilerin farklı projelerde deneyim kazanmalarını kolaylaştırmaktadır. Bu özellikleri sayesinde Python, eğitim kurumlarında en yaygın kullanılan programlama dillerinden biri haline gelmiştir (Lutz, 2013). Bunlara karşın, metin tabanlı programlama eğitimin uygulanmasında karşılaşılan zorluklara dikkat çekmektedir. Öğrencilerin soyut kavramları somut uygulamalara dönüştürmekte zorlanmaları, karmaşık sözdizimi yapılarının anlaşılması ve hata ayıklama süreçlerinin başlangıç düzeyindeki öğrenciler için zorlayıcı olması, bu zorluklar arasında yer almaktadır (Erümit vd., 2019). Bununla birlikte, metin tabanlı programlama eğitiminin, bireylerin bilişsel ve mesleki gelişimlerine olan katkısı göz önüne alındığında, eğitim süreçlerinde etkin bir şekilde entegre edilmesinin önemi vurgulanmıştır. Bu zorluklar göz önüne alındığında, metin tabanlı programlama öğretim sürecinin daha etkili hale getirilmesi ve öğrenciler için kolaylaştırılması gerekliliği ön plana çıkmaktadır. Özellikle, öğrencilerin programlama kavramlarını anlamalarını zorlaştıran soyut yapılar ve hata ayıklama süreçlerindeki güçlükler, öğrenme sürecine yönelik daha destekleyici yaklaşımların benimsenmesini zorunlu kılmaktadır. Bu bağlamda, yapay zeka teknolojilerinin programlama eğitimi sürecine entegre edilmesi, hem öğretim süreçlerini kolaylaştırmak hem de öğrencilerin öğrenme deneyimlerini zenginleştirmek açısından önemli bir fırsat sunmaktadır. Son yıllarda, yapay zekâ destekli araçların programlama öğretiminde kullanımı giderek artmış ve bu araçların, öğrencilerin öğrenme süreçlerini bireyselleştirme, geri bildirim sağlama ve problem çözme becerilerini geliştirme konularında etkili olduğu kanıtlanmıştır (OpenAI, 2021). Bu gelişmeler, programlama eğitiminde yenilikçi yaklaşımların benimsenmesinin ve yapay zekâ teknolojilerinin etkin bir şekilde kullanılmasının önemini bir kez daha ortaya koymaktadır. Bu bağlamda, eğitimde yapay zeka destekli araçların kullanımı gibi yenilikçi yaklaşımlar, bu zorlukların üstesinden gelmek ve öğrencilerin programlama becerilerini daha etkili bir şekilde geliştirmek için potansiyel çözümler sunmaktadır.

Yapay zekâ, insan zekâsını taklit ederek algılama, öğrenme, karar verme ve problem çözme gibi bilişsel süreçleri gerçekleştirebilen sistemlerin tasarımı ve geliştirilmesi ile ilgilenen bir bilim dalıdır. İlk olarak 1950’lerde Alan Turing’in “Makineler düşünebilir mi?” sorusuyla teorik temelleri atılan yapay zekâ, günümüzde eğitimden sağlığa, mühendislikten sanata kadar geniş bir uygulama yelpazesine sahiptir (Gökçe, 2023). Yapay zekâ teknolojilerinin gelişimi, insan yaşamını daha verimli hale getirmektedir.

Yapay zekâ, insan beyninin düşünme, karar verme ve öğrenme süreçlerini modelleyerek bu işlevleri bilgisayarlar veya robotlar aracılığıyla gerçekleştirme amacını taşır. Bu kapsamda yapay sinir ağları, bulanık mantık, genetik algoritmalar ve bilgisayarlı görme gibi yöntemler sıklıkla kullanılmaktadır (Gökçe, 2023). Yapay zekâ uygulamalarının gelişimi, özellikle

otonom araçlar, sağlık teşhis sistemleri ve eğitim teknolojilerinde çığır açıcı yeniliklere olanak tanımıştır. Bu gelişmelerin bir uzantısı olarak, üretken yapay zekâ (ÜYZ) teknolojileri, yapay zekâ alanında yeni bir paradigma sunmaktadır. Üretken yapay zekâ, mevcut verileri analiz ederek yeni ve yaratıcı içerikler üretme yeteneğiyle, özellikle eğitim süreçlerinde kişiselleştirilmiş öğrenme deneyimleri sağlamada önemli bir rol oynamaktadır. Bu teknoloji, öğrencilerin bireysel ihtiyaçlarına yönelik öğrenme materyalleri oluşturma, karmaşık sorunları çözmede rehberlik sağlama ve öğrenme süreçlerini daha interaktif hale getirme potansiyeliyle ön plana çıkmaktadır.

Üretken yapay zekâ, karmaşık veri örneklerinden öğrenerek insan yapımı eserlerden ayırt edilemeyecek içerikler oluşturma yeteneğine sahip sistemlerin geliştirilmesiyle ilgilenir. Bu alandaki ilerlemeler, özellikle Üretken Çekişmeli Ağlar (Generative Adversarial Networks - GANs) gibi teknolojiler aracılığıyla dikkat çekici bir seviyeye ulaşmıştır. GAN'lar, gerçek veri kümelerindeki dağılımı öğrenerek, bu dağılıma benzer sentetik örnekler oluşturmaya hedefler ve görüntü üretiminden metin ve müzik üretimine kadar geniş bir yelpazede uygulanmaktadır (Gözet vd., 2023).

Üretken yapay zekânın en dikkat çekici özelliklerinden biri, öğrenme sürecinde denetimsiz yöntemlere dayanmasıdır. Bu durum, etiketlenmiş veri gereksinimini azaltarak veri işleme süreçlerini hızlandırır. GAN'lar, bir yandan veri üretirken diğer yandan bu verilerin gerçekliğini değerlendiren iki derin ağı rekabetçi bir şekilde çalışmasına dayanır. Bu yapı, hem yüksek kaliteli hem de çeşitli veri üretimini mümkün kılmaktadır (Gözet vd., 2023).

ÜYZ'nin görüntü, metin ve ses üretiminden sağlık ve eğitime kadar geniş bir yelpazede uygulanabilirliği bulunmaktadır. Eğitim bağlamında, özellikle programlama öğretiminde, ÜYZ araçlarının kullanımı öğrencilerin öğrenme süreçlerini desteklemek için güçlü bir potansiyel sunmaktadır. Bu teknolojiler, öğrencilerin kod yazma becerilerini geliştirme, algoritmik düşüncüyü teşvik etme ve hata ayıklama süreçlerini kolaylaştırma gibi işlevlerle öğrenme sürecini daha verimli hale getirebilmektedir (Utrecht University, 2023).

Programlama eğitiminde ÜYZ'nin en önemli katkılarından biri, bilgi işlemsel düşünme becerilerinin geliştirilmesine olanak tanımasıdır. Bilgi işlemsel düşünme, algoritmaların tasarlanması ve karmaşık problemlerin çözülmesi gibi bilişsel süreçleri içermektedir. Özellikle hata ayıklama süreçlerinde, ÜYZ'nin öğrenciler için anında geri bildirim sağlaması ve kodlama hatalarını anlamalarına rehberlik etmesi, bu becerilerin kazanılmasını desteklemektedir. Hata ayıklama, öğrencilerin kodlama becerilerini derinlemesine öğrenmesi ve kendi hatalarını düzeltme alışkanlığı kazanması açısından kritik bir önem taşımaktadır (ACM, 2023).

ÜYZ'nin eğitimde kullanımıyla ilgili çalışmalar, teknolojinin programlama öğreniminde güçlü bir destek sunduğunu ortaya koysa da, bu alanda özellikle ortaokul düzeyinde araştırmalar sınırlıdır. Bu düzeyde uygulamalı çalışmalara ve yenilikçi öğretim yöntemlerine duyulan ihtiyaç büyüktür. Özellikle, ÜYZ destekli araçların ortaokul öğrencilerinin öğrenme deneyimlerini nasıl şekillendirdiği, bilgi işlemsel düşünme becerilerini nasıl etkilediği ve programlama tutumlarını nasıl geliştirdiği üzerine daha fazla araştırma yapılması gerekmektedir (SSRN, 2023).

ÜYZ teknolojileri, yazılım geliştirme süreçlerinde giderek daha yaygın kullanılmakta; ChatGPT, GitHub Copilot ve Amazon CodeWhisperer gibi araçlar, kod tamamlama, hata düzeltme ve açıklama gibi işlevlerde önemli kolaylıklar sunmaktadır (Kiesler & Schiffner, 2023; Tian vd., 2023). Ancak bu araçların büyük çoğunluğu, yetişkin kullanıcılar ya da ileri düzey programlama bilgisine sahip bireyler için tasarlanmış olup, pedagojik yönlendirme ya da öğrenen odaklı açıklama sunma açısından sınırlı kalmaktadır (Becker, 2023; Chen vd., 2024). Özellikle ortaokul düzeyindeki öğrencilerin ihtiyaç duyduğu basitleştirilmiş kavramsal açıklamalar, hata nedenlerine yönelik anlaşılır geri bildirimler ve yapılandırılmış öğrenme ortamları, bu genel amaçlı kod asistanlarında yeterince karşılık bulmamaktadır.

Bu bağlamda, bu tez kapsamında geliştirilen üretken yapay zekâ destekli programlama öğrenme modülü, ortaokul öğrencilerinin bilişsel düzeyine uygun, açıklayıcı ve kavramsal bağlamdan kopmayan geri bildirimler sunmak üzere tasarlanmıştır. Modül, sadece doğru-yanlış değerlendirmesi yapmakla kalmamakta, aynı zamanda öğrencinin hatasının muhtemel nedenini yorumlayarak anlamlı dönütler sağlamaktadır. Böylece öğrencilerin programlama öğrenme süreçlerini destekleyen daha kişiselleştirilmiş, rehberlik temelli bir öğrenme ortamı oluşturulmaktadır. Literatürde sıklıkla eleştirilen “üretken araçların öğrenmeyi yüzeysel hale getirme” riski (Anagnostopoulos, 2023) bu modülde eğitsel yönlendirme ile azaltılmış ve öğrenmeyi derinleştirme hedeflenmiştir.

Araştırmanın Amacı

Bu araştırmanın amacı, ortaokul düzeyinde programlama eğitimi için üretken yapay zekâ destekli bir öğrenme modülünün ortaokul öğrencilerinin bilgi işlemsel düşünme becerileri, programlama başarıları ve tutumlarının değerlendirilmesidir. Günümüzün dijital çağında, bireylerin algoritmik düşünme, problem çözme ve yaratıcı düşünme becerileri geliştirmesi, eğitim sistemlerinin en kritik hedeflerinden biri haline gelmiştir. Ancak, geleneksel programlama öğretim yöntemleri, bireysel öğrenme hızlarına ve ihtiyaçlara yeterince cevap verememekte, bu da öğrenme sürecinde verimliliği düşürmektedir. Bu bağlamda, üretken yapay

zekâ teknolojilerinin sunduğu yenilikçi çözümler, programlama eğitime yönelik daha etkili ve kişiselleştirilmiş yaklaşımlar geliştirilmesi için büyük bir potansiyel taşımaktadır.

Araştırma kapsamında, üretken yapay zekâ teknolojilerinin, öğrencilerin programlama eğitime yönelik tutumları, öğrenme motivasyonları ve bilişsel becerileri üzerindeki etkileri derinlemesine incelenmiştir. Modülün uygulama sürecinde, öğrencilerin programlama öğrenme deneyimleri ortaya çıkarılarak, bu süreçte karşılaşılan güçlükler ve olumlu sonuçlar ortaya konulmuştur. Bu çalışmanın hem teorik hem de pratik düzeyde literatüre önemli katkılar sağlaması beklenmektedir. Teorik olarak, üretken yapay zekâ teknolojilerinin programlama eğitim süreçlerindeki rolüne ilişkin literatürdeki boşlukları doldurmayı hedeflemektedir. Pratik düzeyde ise, öğretmenlere ve eğitim politikacılarına, programlama eğitiminde daha etkili öğretim stratejileri geliştirmek için rehberlik edecek bulgular sunmayı amaçlamaktadır.

Araştırma Soruları

1. Üretken yapay zeka destekli programlama öğreniminin ortaokul öğrencilerinin bilgi işlemsel düşünme becerileri üzerindeki etkisi nedir?
2. Üretken yapay zeka destekli programlama öğreniminin ortaokul öğrencilerinin programlamaya yönelik tutumları üzerindeki etkisi nedir?
3. Üretken yapay zeka destekli programlama öğreniminin ortaokul öğrencilerinin programlama başarısı üzerindeki etkisi nedir?
4. Ortaokul öğrencilerinin üretken yapay zeka destekli programlama öğreniminde davranışları ve deneyimleri nasıldır?

Araştırmanın Önemi ve Gerekçesi

Dijitalleşmenin hızla ilerlediği 21. yüzyılda, bireylerin teknolojiyi yalnızca tüketen değil, aynı zamanda üreten bireyler olarak yetiştirilmesi, eğitim sistemlerinin öncelikli hedeflerinden biri haline gelmiştir. Programlama eğitimi, bireylerin problem çözme, analitik düşünme ve algoritmik düşünme becerilerini geliştirerek, onları geleceğin bilgi ekonomisine hazırlayan kritik bir eğitim alanıdır. Ancak, mevcut programlama eğitim yöntemlerinin, öğrencilerin bireysel öğrenme ihtiyaçlarına yeterince cevap veremediği ve bu nedenle öğrencilerde öğrenme motivasyonu ve tutumları üzerinde olumsuz etkiler yarattığı gözlenmektedir. Bu durum, eğitimde yenilikçi ve kişiselleştirilmiş öğretim yöntemlerine olan gereksinimi artırmaktadır. Üretken yapay zekâ (ÜYZ) teknolojileri, eğitim alanında öğrenme süreçlerini dönüştürme potansiyeline sahiptir. Özellikle ÜYZ tabanlı araçların öğrencilerin bireysel öğrenme hızlarına uyum sağlama, öğrenme süreçlerini kişiselleştirme ve öğrencilerin karşılaştığı öğrenme engellerini tespit etme konularında güçlü bir etki yaratabileceği

öngörülmektedir. Bunun yanı sıra, ÜYZ'nin görsel, işitsel ve yazılı içerik oluşturma yetenekleri, programlama eğitimi daha erişilebilir ve motive edici hale getirme potansiyeline sahiptir. Bu bağlamda, yapay zekâ destekli eğitim modüllerinin, öğrencilerin öğrenme süreçlerine yenilikçi bir boyut kazandıracağı düşünülmektedir.

Türkiye bağlamında, programlama eğitimi son yıllarda müfredatta daha fazla yer bulmaya başlamış olsa da, öğretim yöntemlerinin etkililiği, araç ve materyal çeşitliliği gibi konularda hâlâ sınırlılıklar bulunmaktadır. Özellikle ortaokul düzeyinde programlama eğitiminin yaygınlaşması ve etkili bir şekilde uygulanması, bu yaş grubundaki öğrencilerin hem bilişsel hem de duyuşsal becerilerinin gelişimi açısından büyük bir önem taşımaktadır. Ancak, mevcut literatür incelendiğinde, yapay zekâ destekli araçların ortaokul düzeyinde programlama eğitimi süreçlerine entegrasyonuna dair sınırlı sayıda çalışma olduğu görülmektedir (Benzer vd., 2023).

Bu araştırmanın önemi, programlama eğitiminin bireysel öğrenme farklılıklarını dikkate alarak kişiselleştirilmiş bir öğrenme deneyimi sunması ve eğitim süreçlerinde yapay zekâ teknolojilerinin etkin kullanımını desteklemesinde yatmaktadır. Araştırma, yalnızca eğitim teknolojileri literatürüne katkı sağlamakla kalmayıp, aynı zamanda öğretmenlere, eğitim tasarımcılarına ve politika yapıcılara rehberlik edecek yenilikçi bir çerçeve sunmayı hedeflemektedir. Bu doğrultuda, çalışma, yapay zekâ teknolojilerinin eğitim alanındaki uygulanabilirliğini göstermek ve gelecekteki eğitim reformlarına temel oluşturmak açısından kritik bir role sahiptir.

Bu araştırmanın önemi, yalnızca programlama eğitiminin bireysel öğrenme farklılıklarını dikkate alarak kişiselleştirilmiş bir deneyim sunmasında değil, aynı zamanda eğitimde fırsat eşitliğini teşvik etme potansiyelinde de yatmaktadır. Özellikle dezavantajlı bölgelerdeki devlet okullarında bu tür yeni teknolojilerin kullanılması, öğrencilerin öğrenme süreçlerine erişimini artırarak, eğitimin daha kapsayıcı ve adil hale gelmesine katkı sağlamaktadır (Eynon & Malmberg, 2021). Araştırmanın odaklandığı bir diğer önemli nokta, üretken yapay zekâ destekli bir öğrenme modülünün geliştirilmesidir. Bu durum, çalışmayı yalnızca bir teorik incelemeden öteye taşıyarak, doğrudan uygulamaya dönük bir katkı sunmaktadır. Ayrıca, bu çalışmada ele alınan değişkenlerin özgünlüğü ve deneysel yöntemlerin sistematik bir şekilde kullanılması, araştırmayı metodolojik açıdan değerli kılmaktadır. Öğrencilerin bilgi işlemsel düşünme becerileri, programlama başarıları ve tutumlarının birlikte incelenmesi, eğitim teknolojilerinin etkinliğini çok boyutlu olarak değerlendirme fırsatı sunmaktadır. Deneysel tasarım sürecinde kullanılan karma yöntem yaklaşımı, hem nitel hem de nicel verilerin birlikte değerlendirilmesine olanak tanıyarak, araştırmanın bulgularını daha

güvenilir hale getirmektedir.

Bu bağlamda, çalışma yalnızca eğitim teknolojileri literatürüne katkı sağlamakla kalmayıp, aynı zamanda öğretmenlere ve eğitim tasarımcılarına rehberlik edecek yenilikçi bir çerçeve sunmakta ve gelecekteki eğitim reformlarına temel oluşturma potansiyeli taşımaktadır.

Araştırmanın Sınırlılıkları

Araştırmanın sınırlılıkları aşağıda belirtilmiştir:

- **Süre Kısıtı:** Araştırma, belirli bir akademik dönem içerisinde gerçekleştirilmiş olup, modülün uzun vadeli etkileri veya öğrenme kalıcılığı değerlendirilmemiştir.
- **Katılımcı Sayısı:** Çalışma, sınırlı sayıda ortaokul öğrencisiyle gerçekleştirilmiştir. Bu durum, bulguların genelleştirilebilirliğini kısıtlamaktadır.
- **Konu İçeriği:** Modül, Python programlama dili ile sınırlı bir içerik sunmuş olup, farklı programlama dillerinin öğretimine yönelik etkileri değerlendirilememiştir.
- **Teknik Altyapı:** Modülün uygulanabilirliği, okulların mevcut teknolojik altyapısına bağlıdır. Düşük internet hızı, yetersiz bilgisayar sayısı ve teknik destek eksikliği gibi faktörler uygulamanın verimliliğini sınırlamıştır.
- **Sistemin Teknik Kapasitesi:** Kullanılan üretken yapay zekâ modülünün teknik kapasitesi, tüm öğrenme ihtiyaçlarını karşılamakta yetersiz kalmış olabilir. Özellikle karmaşık kodlama örnekleri ve ileri düzey programlama konularında sınırlılıklar gözlemlenmiştir.

Varsayımlar

Araştırmadaki varsayımlar aşağıdaki şekilde belirlenmiştir:

- Araştırmaya katılan öğrencilerin veri toplama sürecinde kullanılan ölçeklerde yer alan maddelere dikkatli ve dürüst bir şekilde yanıt verdikleri varsayılmıştır.

Terim ve Tanımlar

Programlama Eğitimi: Programlama eğitimi, bireylerin algoritmik düşünme, problem çözme ve yazılım geliştirme becerilerini geliştirmek amacıyla yapılandırılmış öğrenme süreçlerini ifade eder. Bu süreçte, öğrencilerin bilgi işlemsel düşünme becerileri kazanmaları hedeflenir (Wing, 2006).

Yapay Zekâ (YZ): Yapay zekâ, insan zekâsını taklit eden, öğrenme, akıl yürütme ve problem çözme gibi bilişsel süreçleri gerçekleştiren sistemler olarak tanımlanır (Russell & Norvig, 2016).

Üretken Yapay Zekâ (ÜYZ): Üretken yapay zekâ, öğrenme sürecinde gerçek veri dağılımını modelleyerek yeni, gerçek veriye benzeyen içerik üretebilen yapay zekâ sistemlerini ifade eder. ÜYZ'nin en yaygın uygulamaları arasında görseller, metinler ve müzik üretimi bulunmaktadır (Goodfellow vd., 2014).

Bilgi İşlemsel Düşünme: Bilgi işlemsel düşünme, bireylerin bir problemin çözümünü soyutlama, algoritma oluşturma ve bilgisayar desteğiyle uygulama becerisini ifade eder. Bu beceri, günümüz eğitiminde dijital okuryazarlığın temel bir bileşeni olarak kabul edilmektedir (Wing, 2006).

Python: Python, açık kaynaklı, yüksek seviyeli, kullanımı kolay bir programlama dilidir. Python, basit sözdizimi ve geniş kütüphane desteğiyle hem yeni başlayanlar hem de profesyonel geliştiriciler için tercih edilmektedir (Van Rossum, 1995).

İKİNCİ BÖLÜM

Kuramsal Çerçeve ve İlgili Araştırmalar

Günümüzün hızla dijitalleşen dünyasında, programlama eğitimi ve yapay zekâ teknolojilerinin eğitim süreçlerine entegrasyonu, eğitim araştırmalarının öncelikli alanlarından biri haline gelmiştir. Bu bölümde, programlama eğitimi ve yapay zekâ arasındaki ilişkiyi açıklayan teorik temeller, üretken yapay zeka ve bilgi işlemsel düşünme özelinde yapılan önceki çalışmalar detaylandırılmıştır. Aynı zamanda, üretken yapay zekâ teknolojilerinin eğitim alanındaki rolü ve etkileri incelenmiştir.

Bilgi İşlemsel Düşünme

Bilgi işlemsel düşünme, modern çağın karmaşık problem çözme süreçlerini anlamak ve yönetmek için geliştirilmiş bir yetkinlik alanı olarak tanımlanmaktadır. İlk kez Seymour Papert tarafından 1980'lerde ortaya atılan bu kavram, problem çözme, sistem tasarımı ve insan davranışlarını bilgisayar temelli kavramlar üzerinden analiz etmeye dayalı bir yaklaşımı ifade etmektedir (Papert, 1980; Wing, 2006). Özellikle 21. yüzyıl becerileri arasında yer alan bu yetkinlik, bireylerin dijital dünyada karşılaştıkları karmaşık durumlara adapte olmalarını ve çözüm üretmelerini sağlama amacı taşımaktadır. Bilgi işlemsel düşünme; soyutlama, algoritmik düşünme, problem çözme, parçalara ayırma ve hata ayıklama gibi farklı boyutlardan oluşmaktadır. Bu çok yönlü yapısı, disiplinler arası bir anlayışla bilgi işlemsel düşünmenin eğitim, mühendislik ve bilim gibi alanlarda yaygınlaşmasını sağlamıştır (Üzümcü & Bay, 2018).

Bilgi işlemsel düşünme (BİD), disiplinler arası bir yaklaşımla ele alındığında gerçek yaşam problemlerine çözüm üretme becerisini geliştiren bir kavramdır. Top ve Arabacıoğlu (2021) tarafından yapılan sistematik bir alanyazın taramasında, 2015-2020 yılları arasında Türkiye’de BİD üzerine yapılan çalışmaların önemli bir artış gösterdiği belirtilmiştir. Özellikle kodlama ve robotik kodlama konuları bu çalışmalarda en çok incelenen alanlar olarak öne çıkmıştır. Çalışmanın sonuçlarına göre, BİD becerisi kazandırmaya yönelik etkinlikler, bireylerin problem çözme, algoritmik düşünme ve yaratıcılık gibi becerilerini artırmaktadır. Ayrıca, bu becerilerin geliştirilmesinde gerçek yaşam problemlerinin disiplinler arası bir yaklaşımla ele alınması önerilmektedir (Top & Arabacıoğlu, 2021). Eğitimde bu yaklaşımın

yaygınlaşması, sadece bilişim teknolojileri değil, matematik, fen bilimleri ve hatta sosyal bilimler gibi farklı disiplinlerde de olumlu sonuçlar ortaya koyabilir.

Bilgi işlemsel düşünme etkinlikleri, öğrencilerin problem çözme becerileri ve problem çözmeye yönelik tutumları üzerinde önemli etkiler yaratmaktadır. Temel ve Yavuz Mumcu'nun (2024) gerçekleştirdiği deneysel bir araştırmada, bu etkinliklerin özellikle 7. sınıf öğrencilerinde problem çözme süreçlerine nasıl katkı sağladığı incelenmiştir. Çalışma, hem nicel hem de nitel yöntemlerle yürütülmüş ve öğrencilerin problem çözme becerilerinde anlamlı bir artış gözlenmiştir. Ayrıca, etkinliklerin öğrencilere sadece teknik bilgi kazandırmakla kalmayıp, aynı zamanda günlük yaşam problemlerine farklı açılardan yaklaşma becerisini de kazandırdığı belirtilmiştir. Araştırmada kullanılan "parçalara ayırma," "soyutlama" ve "örüntü-model oluşturma" gibi farklı bilgi işlemsel düşünme boyutları, öğrencilerin karmaşık problemleri daha kolay analiz etmelerine ve çözüm önerileri geliştirmelerine olanak tanımıştır (Temel vd., 2024).

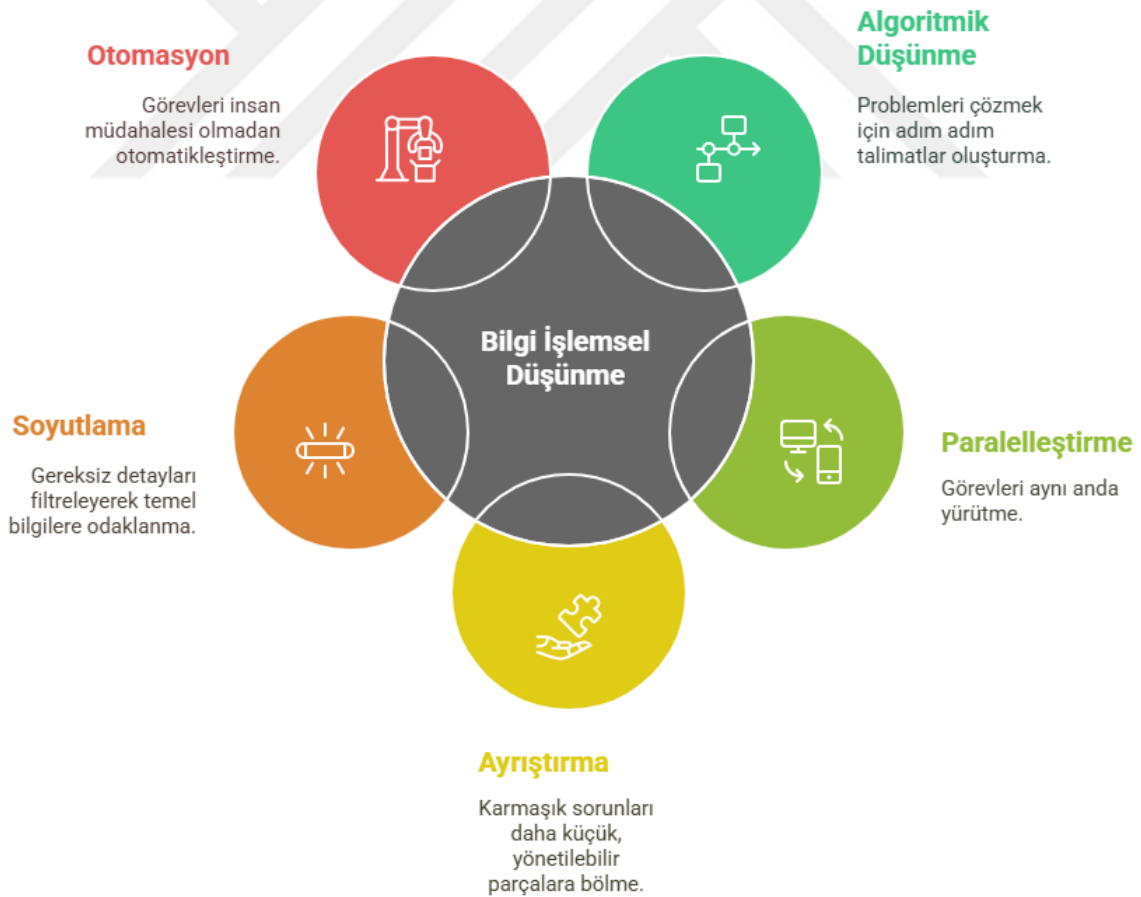
Bilgi işlemsel düşünme becerileri, bireylerin problem çözme yeteneklerini geliştiren önemli bir kavram olarak öne çıkmaktadır. Bilici ve Güler'in (2021) yaptığı çalışmada, ortaokul öğrencilerinin bilgi işlemsel düşünme becerileri ile akademik öz yeterlikleri arasında orta düzeyde, pozitif bir ilişki olduğu belirlenmiştir. Çalışma, öğrencilerin bilgi işlemsel düşünme becerilerinin özellikle yaratıcılık, algoritmik düşünme ve işbirlikli çalışma gibi alt boyutlarda gelişim gösterdiğini ortaya koymaktadır. Bununla birlikte, bu becerilerin cinsiyet, bilgisayara sahip olma durumu, anne ve baba eğitim düzeyi gibi çeşitli sosyo-demografik değişkenlere bağlı olarak anlamlı farklılıklar gösterdiği bulunmuştur. Örneğin, bilgisayara sahip olan öğrencilerin bilgi işlemsel düşünme becerilerinin daha yüksek olduğu tespit edilmiştir. Ayrıca, bu becerilerin geliştirilmesinde disiplinler arası etkinliklerin ve uygulamaların büyük bir rol oynadığı vurgulanmıştır (Bilici & Güler, 2021).

Bilgi işlemsel düşünme, bireylerin problem çözme ve eleştirel düşünme becerilerini geliştirmek için önemli bir eğitim alanıdır. Ekinci vd. (2023) çalışmasında, ortaokul öğrencilerinin kodlamaya yönelik tutumları (KYT) ile bilgi işlemsel düşünme becerileri (BİDB) arasındaki ilişki incelenmiştir. Araştırma, öğrencilerin KYT ve BİDB düzeylerinin orta seviyede olduğunu, bu iki kavram arasında pozitif yönde orta düzeyde anlamlı bir ilişki bulunduğunu ortaya koymuştur. Bunun yanı sıra, günlük bilgisayar kullanım süresinin problem çözme becerileri üzerinde etkili olduğu, ancak diğer BİDB alt boyutları üzerinde anlamlı bir fark yaratmadığı belirlenmiştir. Bu bulgular, bilgi işlemsel düşünme becerilerinin geliştirilmesi için kodlama eğitiminin teşvik edilmesi gerektiğini vurgulamaktadır. Özellikle blok tabanlı kodlama araçlarının küçük yaş gruplarına yönelik etkin bir yöntem olarak kullanılması,

öğrencilerin motivasyonlarını artırarak kodlama ve problem çözme süreçlerinde daha başarılı olmalarını sağlamaktadır (Ekinci vd., 2023).

Bilgi işlemsel düşünme, yalnızca temel bir problem çözme becerisi olarak değil, aynı zamanda bireylerin dijital çağda etkili, analitik ve sistematik düşünebilmesini sağlayan bir zihinsel çerçeve olarak değerlendirilmektedir. Bu becerinin eğitimde kazandırılması, öğrencilerin hem akademik başarılarını artırmakta hem de gerçek yaşam problemlerine daha sistemli yaklaşımlar geliştirmelerine olanak tanımaktadır (Yadav vd., 2014). Literatürde bilgi işlemsel düşünmenin birçok boyuttan oluştuğu ve bu boyutların bir bütün olarak bireylerin bilişsel gelişimini desteklediği vurgulanmaktadır (Shute vd., 2017). Bu araştırmada kullanılan “Bilgi İşlemsel Düşünme Öz Değerlendirme Ölçeği” de bu çok boyutlu yapıyı esas alarak; algoritmik düşünme, paralelleştirme, ayrıştırma, soyutlama ve otomasyon olmak üzere beş temel alt boyutu kapsamaktadır. Aşağıda bu alt boyutların her biri, alan yazında yer alan tanımları ve örnek uygulamalarıyla birlikte detaylı biçimde açıklanmıştır.

Şekil 1. Bilgi İşlemsel Düşünmenin Alt Boyutları



Algoritmik düşünme, bilgi işlemsel düşünmenin temel yapı taşlarından biri olarak kabul edilmektedir. Bu düşünme biçimi, bir problemin çözümünü adım adım tanımlayarak bir algoritma haline getirme sürecini içerir (Czerkowski & Lyman, 2015). Öğrenciler algoritmik düşünme becerisi sayesinde, karmaşık problemleri daha küçük ve yönetilebilir parçalara ayırarak sistematik çözüm yolları geliştirebilirler. Bu beceri sadece programlamada değil, matematik, fen bilimleri ve günlük yaşam problemlerinde de etkili çözüm üretimi açısından önemlidir (Grover & Pea, 2013). Wing (2006), algoritmik düşünmeyi bilgi işlemsel düşünmenin merkezine yerleştirerek, bireylerin bilgisayar bilimleri sadece teknik bir beceri değil, düşünsel bir yaklaşım olarak benimsemeleri gerektiğini vurgular. Ayrıca algoritmik düşünmenin, öğrencilerin kodlama sırasında mantıksal sıra takibi, koşullu ifadeleri anlama ve döngüleri kullanma gibi kritik programlama becerilerini kazanmalarında belirleyici bir rol oynadığı görülmektedir (Kalelioğlu, 2015).

Paralleleştirme, aynı anda birden fazla işlemi tasarlayabilme ve bu işlemleri eş zamanlı olarak yönetebilme becerisini ifade etmektedir. Bu alt boyut, öğrencilerin bir problemi çözmek için birbirinden bağımsız görevleri aynı anda planlayabilmelerine ve bu görevleri uyum içinde yürütebilmelerine olanak sağlamaktadır (Grover & Pea, 2013). Özellikle karmaşık sistemlerde farklı bileşenlerin eş zamanlı çalışmasının anlaşılmasında paralelleştirme önemli bir rol üstlenmektedir. Bu beceri, günlük yaşamda çoklu görevleri planlama gibi durumlarda da işlevsel olabilmektedir.

Programlama eğitiminde paralelleştirme; döngüler, olay tabanlı komutlar ve eşzamanlı çalışan nesne ya da karakterlerin kontrolü gibi uygulamalar aracılığıyla kazandırılabilir. Örneğin, bir oyun tasarımında bir karakterin hareket ederken aynı anda puan toplaması gibi durumlar, bu becerinin uygulamalı karşılıkları arasında yer almaktadır. Shute ve arkadaşları (2017), paralelleştirmenin öğrencilerin sistemsal düşünme becerilerini geliştirdiğini ve programlamaya karşı duydukları öz yeterliği artırdığını ifade etmektedir. Bu beceri sayesinde öğrenciler, aynı anda birden fazla değişkenin etkileşimini gözeterek daha verimli ve etkili algoritmalar geliştirebilmektedir.

Ayrıştırma, karmaşık bir problemin daha küçük, anlaşılabilir ve yönetilebilir alt problemlere bölünmesi süreci olarak tanımlanmaktadır. Bu beceri, öğrencilerin doğrudan tüm problemle baş etmek yerine, problemi bileşenlerine ayırarak çözüm sürecini daha sistematik ve etkili bir şekilde yürütmelerine olanak tanımaktadır (Wing, 2006). Ayrıştırma sayesinde öğrenciler, yalnızca yazılım geliştirme süreçlerinde değil; matematiksel problemlerden günlük yaşam senaryolarına kadar pek çok alanda stratejik düşünme becerisi kazanmaktadır. Literatürde, ayrıştırmanın öğrencilerin özellikle problem çözme basamaklarını netleştirmesinde

ve kodlamaya yönelik bilişsel yüklerini azaltmasında etkili olduğu belirtilmektedir (Grover vd., 2020). Örneğin, bir oyun tasarımı sürecinde; karakter hareketleri, skor takibi ve görsel geri bildirim gibi alt sistemlerin ayrı ayrı tasarlanması, ayrıştırma becerisinin doğrudan bir yansımasıdır. Ayrıca Papadakis (2022), bu becerinin öğrencilere yazılım modüllerini daha kolay kavrama, hata ayıklama sürecinde belirli bölgelere odaklanma ve tekrar kullanılabilir kod yapıları oluşturma gibi önemli kazanımlar sağladığını ifade etmektedir. Ayrıştırma becerisi, öğrencilerin yalnızca bir problemi çözmelerini değil; aynı zamanda problem çözme sürecini planlama, analiz etme ve uygulama yetkinliklerini artırmaktadır.

Soyutlama, bir problemin çözümünde yalnızca gerekli olan bilgilere odaklanarak, gereksiz ayrıntıların dışarıda bırakılması süreci olarak tanımlanmaktadır. Bu beceri, öğrencilerin karmaşık yapılar içinde genel örüntüleri ayırt edebilmelerine ve problemi temsil eden temel özellikleri tanımlayarak daha sade, etkili çözümler geliştirebilmelerine olanak tanımaktadır (Wing, 2006). Soyutlama sayesinde öğrenciler, bilgi yoğunluğunu azaltarak bilişsel yüklerini hafifletmekte ve farklı problemler arasında genellemeler yapabilme yetkinliği kazanmaktadır.

Programlama eğitiminde soyutlama; fonksiyon tanımlamaları, değişken kullanımı, sınıflar ve veri yapıları gibi kavramların öğretiminde kendini göstermektedir. Öğrenciler, benzer işlemleri bir araya getirerek işlevsel kod blokları oluşturmakta, bu sayede hem modüller hem de tekrar kullanılabilir yazılımlar geliştirebilmektedirler. Papadakis (2022), özellikle ilköğretim ve ortaokul düzeyinde yapılan kodlama etkinliklerinde soyutlama becerisinin, öğrencilerin algoritmik yapı kurma yeteneklerini önemli ölçüde artırdığını ortaya koymaktadır. Ayrıca Zhang ve Nouri (2022), öğrencilerin soyutlama becerilerinin gelişmesinin, farklı problemleri çözmede daha hızlı stratejiler üretmelerine ve alternatif çözüm yolları oluşturmalarına katkı sağladığını vurgulamaktadır. Soyutlama, aynı zamanda yapay zeka destekli programlama ortamlarında öğrencilerin yalnızca sonuca değil, sürece de odaklanmalarını sağlayarak, kodun arkasındaki mantığı daha net kavramalarına yardımcı olmaktadır.

Otomasyon, bireyin daha önce çözümünü bildiği bir problemi benzer durumlarda yeniden çözmek yerine, bu çözümü otomatikleştirmesi ve tekrar kullanabilir hale getirmesi sürecini ifade etmektedir (Wing, 2006). Bu beceri, öğrencilerin yazılım geliştirme sürecinde tekrarlayan görevleri belirleyerek bu görevleri otomatik hale getirmelerine, böylece zaman kazanarak daha karmaşık problemlere odaklanmalarına olanak tanımaktadır. Otomasyon yalnızca teknik becerilerin gelişmesini değil, aynı zamanda sistemsel düşünme ve süreç yönetimi gibi üst düzey düşünme becerilerinin de gelişmesini desteklemektedir. Programlama

öğretiminde otomasyon; döngüler, fonksiyonlar ve koşullu ifadeler gibi yapılarla ilişkilidir. Öğrenciler bir görevi yalnızca bir kez tanımlayarak farklı bağlamlarda kullanabildiklerinde, bu görev otomatikleşmiş bir yapı haline gelmektedir. Bu durum, özellikle algoritma verimliliği, kodun modülerliği ve yeniden kullanılabilirliği açısından önem arz etmektedir (Shute vd., 2017). Lee vd. (2021), otomasyon becerisinin erken yaşlarda kazandırılmasının öğrencilerin sistemsel düşünme kapasitelerini artırdığını ve kodlama sürecinde daha stratejik yaklaşımlar geliştirmelerine katkı sağladığını ifade etmektedir. Ayrıca Papadakis (2023), otomasyon becerisi gelişmiş öğrencilerin problem çözme süreçlerinde daha yüksek başarı gösterdiğini ve karmaşık sistemlerde daha az hata yaptığını belirtmektedir. Bu bağlamda otomasyon, bilgi işlemsel düşünmenin üst düzeydeki bir yansıması olarak, öğrencilerin kodlama süreçlerini daha verimli ve etkili hale getirmektedir.

Eğitimde, özellikle programlama öğretiminde, paralelleştirme; döngüler, olay tabanlı programlama ve çoklu karakter/nesne kontrolü gibi konularda öğrencilerin bilişsel esneklik kazanmasına yardımcı olur. Örneğin, bir oyun programında bir karakterin zıplarken aynı anda puan toplaması gibi çoklu işlemlerin senkronize edilmesi, bu becerinin pratik karşılığıdır. Shute vd. (2017) göre paralelleştirme, öğrencilerin sistemsel düşünme becerilerini de desteklemekte ve programlamaya karşı duydukları güveni artırmaktadır. Paralelleştirme sayesinde öğrenciler, aynı anda birden fazla değişkenin etkileşimini gözeterek daha verimli algoritmalar tasarlayabilirler.

Ayrıştırma, karmaşık bir problemi daha küçük, anlaşılabilir ve yönetilebilir alt problemlere bölme sürecidir. Bu alt boyut, öğrencilerin büyük bir problemi doğrudan çözmeye çalışmak yerine, problem bileşenlerini analiz ederek çözüm sürecini kolaylaştırmalarını sağlar (Wing, 2006). Ayrıştırma, bilgi işlemsel düşünmenin en temel stratejilerinden biri olarak kabul edilir çünkü hem bilişsel yükü azaltır hem de problem çözme sürecini daha sistematik hale getirir.

Öğrenciler programlama yaparken bir algoritmanın adımlarını planlamak, bir kod bloğunu alt fonksiyonlara bölmek ya da kullanıcı girişleriyle işlem çıktıları arasındaki ilişkiyi analiz etmek gibi durumlarda bu beceriyi aktif biçimde kullanırlar. Grover ve Pea (2013), ayrıştırmanın özellikle başlangıç düzeyinde kodlama öğrenen öğrenciler için soyut düşünmeyi kolaylaştıran bir yapı sunduğunu ve algoritmik yapıların anlaşılmasını hızlandığını belirtmektedir. Ayrıca ayrıştırma, öğrencilerin problem çözme sürecinde daha özgüvenli hareket etmelerini sağlar ve hata ayıklama sürecini kolaylaştırır çünkü sorunları daha küçük birimler halinde ele almayı mümkün kılar (Shute vd., 2017).

Programlama öğretimi, öğrencilerin algoritmik düşünme, ayrıştırma, genelleme, soyutlama ve otomasyon gibi bilgi işlemsel düşünme becerilerini geliştirmede kritik bir rol oynamaktadır (Wing, 2006). Bu beceriler, yalnızca programlama faaliyetlerinde değil; günlük yaşam problemlerinin çözümünde ve disiplinler arası uygulamalarda da etkili olmaktadır. Bilgi işlemsel düşünme becerilerinin değerlendirilmesinde uluslararası kabul gören ölçme araçlarından biri olan Bilge Kunduz (Bebras) Testi, dünya genelinde 50'den fazla ülkede uygulanmaktadır. Bebras Testi, öğrencilerin algoritma temelli düşünme süreçlerini ölçmeyi hedefleyen kısa, mantıksal ve çok adımlı problem senaryoları içermektedir (Dagienė & Futschek, 2008). Türkiye'de de Bilim ve Toplum projeleri kapsamında uygulanan Bilge Kunduz etkinlikleri, ortaokul ve lise düzeyinde öğrencilerin bilgi işlemsel düşünme becerilerinin teşhisinde ve programlama öğretiminin etkililiğinin değerlendirilmesinde kullanılmaktadır (Aydın, 2022). Bu bağlamda, programlama eğitimi, yalnızca teknik beceri kazandırmakla kalmamakta; aynı zamanda bilgi işlemsel düşünmenin temel yapı taşlarını öğrencilere kazandırarak onları dijital çağın gerekliliklerine hazırlamaktadır. Üzüm, Elçiçek ve Pesen'in (2024) meta-analiz çalışmasında, programlama öğretiminin bilgi işlemsel düşünme üzerindeki genel etkisinin pozitif ve geniş olduğu bulunmuştur. Araştırma, metin tabanlı programlamanın, diğer programlama türlerine göre bilgi işlemsel düşünme becerileri üzerinde daha etkili olduğunu ortaya koymuştur. Ayrıca, bu becerinin etkisinin uygulama süresine, örneklem büyüklüğüne ve eğitim düzeyine göre farklılık gösterdiği saptanmıştır. Örneğin, okul öncesi düzeyde programlama öğretiminin etkisi oldukça yüksek bulunurken, bu etkinin yükseköğretim seviyesinde azaldığı gözlemlenmiştir. Bu bulgular, bilgi işlemsel düşünme becerilerinin kazandırılmasında yaş ve eğitim düzeyine uygun yöntemlerin kullanılmasının önemini vurgulamaktadır (Üzüm vd., 2024).

Programlama Eğitimi

Programlama eğitimi, bireylerin algoritmik düşünme, problem çözme ve yaratıcı düşünme becerilerini geliştirmek için yapılandırılmış bir öğrenme sürecidir. 21. yüzyılda, dijital teknolojilerin hayatın her alanına entegre olmasıyla, programlama becerileri sadece yazılım geliştiriciler için değil, aynı zamanda her birey için kritik bir yetkinlik haline gelmiştir (Wing, 2006). Programlama eğitiminin erken yaşlarda başlaması, öğrencilerin bilgi işlemsel düşünme becerilerini kazanmalarına olanak tanır ve onları gelecekteki dijital ekonomiye hazırlamak için güçlü bir temel oluşturur (Grover & Pea, 2013). Bununla birlikte, programlama eğitimi yalnızca teknik beceriler kazandırmayı değil, aynı zamanda bireylerin eleştirel düşünme, takım çalışması ve iletişim becerilerini de geliştirmeyi amaçlar (Papert, 1980). Bu bağlamda, görsel programlama araçları, metin tabanlı programlama dillerine geçiş için etkili bir öğrenme ortamı

sunarak, öğrenme süreçlerini kolaylaştırır ve öğrencilerin motivasyonunu artırır (Resnick vd., 2009). Programlama eğitiminin etkili bir şekilde uygulanması, sadece yazılım geliştirme yetkinlikleri kazandırmanın ötesinde, bireylerin bilişsel süreçlerini güçlendirmeyi ve 21. yüzyıl becerilerini kazandırmayı hedeflemektedir (Lye & Koh, 2014). Araştırmalar, programlama öğretiminin yaratıcı düşünme (Navarrete, 2013) ve yansıtıcı düşünme becerilerini (Kalelioğlu, 2015) geliştirdiğini göstermiştir. Ayrıca, programlama kodlamanın ötesinde soyutlama ve ayırtırma becerilerini içeren daha kapsamlı bir öğrenme süreci sunmaktadır (Hromkovič vd., 2016).

Erken yaşta programlama eğitiminin önemi, bireylerin 21. yüzyıl becerilerini kazanmasındaki rolüyle açıklanmaktadır. Programlama eğitimi, yaratıcılık, eleştirel düşünme, problem çözme ve işbirliği gibi becerilerin gelişimini desteklemektedir (Durak vd., 2017). Araştırmalar, bu becerilerin çocukların hem bireysel hem de toplumsal gelişimlerinde kritik bir yer tuttuğunu göstermektedir. Ayrıca, erken yaşta programlama eğitimi, çocukların algoritmik düşünme ve teknolojik okuryazarlık gibi bilişsel yetkinliklerini güçlendiren bir süreç olarak değerlendirilmiştir. Programlama süreçlerinin öğrencilerin üst düzey düşünme becerilerine katkıda bulunduğu ve öğrenme deneyimlerini zenginleştirdiği ifade edilmektedir (Durak vd., 2017).

Programlama eğitimi üzerine yapılan çalışmalar, öğrencilerin bilişsel, duyuşsal ve sosyal becerilerinin geliştirilmesinde önemli katkılar sağladığını göstermektedir. Türkiye’de yapılan bir içerik analizi çalışması, kodlama ve robotik eğitiminde kullanılan teknolojilerin çeşitliliğini ve öğrenciler üzerindeki etkilerini vurgulamaktadır. Özellikle blok tabanlı programlama araçlarının, ortaokul düzeyinde kodlama öğretimi için en sık tercih edilen yöntemlerden biri olduğu görülmüştür. Scratch, bu alanda en yaygın kullanılan araçlardan biri olarak öne çıkmaktadır ve öğrencilerin yaratıcı düşünme, algoritmik problem çözme ve işbirlikçi öğrenme becerilerini desteklemektedir (Alper & Şanlı, 2024). Araştırma bulguları, kodlama ve robotik eğitiminin yalnızca akademik başarıyı artırmakla kalmadığını, aynı zamanda öğrencilerin motivasyon, öz-yeterlik ve iletişim becerileri üzerinde de olumlu etkiler yarattığını ortaya koymaktadır. Araştırmalar, ortaokul öğrencilerinin programlamaya yönelik tutumlarının genellikle olumlu olduğunu ve bu tutumların cinsiyet değişkenine göre anlamlı bir farklılık göstermediğini ortaya koymaktadır.

Programlama eğitimi, bireylerin algoritmik düşünme, problem çözme ve yaratıcı düşünme becerilerini geliştirmelerinde önemli bir araç olarak öne çıkmaktadır. Eğitimde kullanılan yöntemler ve teknolojiler, öğrencilerin programlama becerilerini kazanmasında ve geliştirmesinde kritik bir rol oynamaktadır. Örneğin, Scratch gibi blok tabanlı programlama

araçları, öğrencilerin algoritmik düşünme ve yaratıcı problem çözme becerilerini desteklemekle kalmayıp, aynı zamanda programlamaya yönelik olumlu tutumlar geliştirmelerine olanak tanımaktadır (Gürsoy & Çekmez, 2019). Bu tür araçların, özellikle programlama eğitimi alanında eşit fırsatlar sunma potansiyeli taşıdığı, erkek ve kız öğrenciler arasında tutum farklılıklarını minimize ettiği görülmektedir. Blok tabanlı programlama araçları, kod yazmayı soyut bir kavram olmaktan çıkararak öğrenciler için somut bir öğrenme deneyimine dönüştürmektedir.

Diğer yandan, metin tabanlı programlama gibi daha geleneksel yaklaşımlar, özellikle ileri düzey öğrenciler için önemli fırsatlar sunmaktadır. Geçitli ve Bümen'in (2020) araştırması, lise öğrencilerine yönelik kademelendirilmiş etkinliklerle desteklenen metin tabanlı programlama öğretiminin, öğrencilerin programlama becerilerini anlamlı derecede geliştirdiğini ortaya koymuştur. Bu tür etkinlikler, bireysel farklılıkları göz önünde bulundurarak, her seviyedeki öğrenciye uygun öğrenme içerikleri sunmayı mümkün kılmaktadır. Ayrıca, bu yöntemler sınıf içi iş birliğini teşvik ederek öğrencilerin derse olan motivasyonlarını artırmaktadır. Metin tabanlı programlama, öğrencilerin problem çözme süreçlerinde daha analitik bir yaklaşım geliştirmelerini sağlaması açısından da önemlidir.

Sonuç olarak, programlama eğitiminde kullanılan yöntem ve teknolojiler, hem bireysel öğrenme süreçlerini desteklemek hem de öğrencilerin genel beceri gelişimini sağlamak açısından büyük bir potansiyele sahiptir. Blok tabanlı araçlar, başlangıç düzeyindeki öğrenciler için erişilebilir ve eğlenceli bir giriş noktası sunarken, metin tabanlı programlama, daha ileri düzeydeki öğrenciler için derinlemesine bir öğrenme süreci sağlamaktadır. Bu çeşitlilik, programlama eğitiminde bireysel farklılıkların dikkate alınmasını ve öğrenme süreçlerinin daha etkili bir şekilde yapılandırılmasını mümkün kılmaktadır.

Son yıllarda programlama eğitimine yönelik artan ilgi, bu alanın hem bireysel hem de toplumsal düzeyde önemini ortaya koymaktadır. Saygıner ve Tüzün (2023) tarafından yapılan bir araştırma, programlama eğitiminde yaşanan başlıca zorlukların algoritma ve programlama mantığının öğretimindeki eksikliklerden kaynaklandığını vurgulamaktadır. Bu zorluklar, öğrencilerin programlama dillerinin karmaşık sözdizimlerini anlamakta zorlanmaları, yabancı dilde kod yazımına adapte olamamaları ve problem çözme süreçlerinde yetersiz kalmalarıyla ilişkilendirilmiştir. Çalışma, aynı zamanda geleneksel öğretim yöntemlerinin bu sorunları derinleştirdiğini belirtmektedir. Özellikle, öğretmenlerin sadece bir programlama dilini öğretmeye odaklanmaları ve dinamik kavramları statik materyallerle aktarmaları, öğrencilerin öğrenme sürecini olumsuz etkilemektedir.

Dünya’da Programlama Eğitimi

Programlama eğitimi, dünya genelinde eğitim politikalarının stratejik bir unsuru haline gelmiştir. Dijital dönüşümün hızla ilerlediği bir çağda, programlama becerileri yalnızca bilişim sektörü çalışanları için değil, aynı zamanda gelecekteki işgücü için de vazgeçilmez bir yetkinlik olarak görülmektedir (Balanskat & Engelhardt, 2015). Birçok ülke, programlama eğitiminin yaygınlaştırılması ve daha erken yaş gruplarına kazandırılması amacıyla eğitim müfredatlarında köklü değişikliklere gitmiştir. Bu bağlamda, programlama eğitimi, bireylerin algoritmik düşünme, problem çözme ve yaratıcı düşünme becerilerini geliştirmeyi hedeflerken, aynı zamanda dijital okuryazarlığın temel bir bileşeni olarak ele alınmaktadır (Grover & Pea, 2013).

Avrupa’da Programlama Eğitimi

Avrupa ülkeleri, programlama eğitimi müfredatlarına entegre etme konusunda öncülük eden bölgelerden biridir. İngiltere, 2014 yılında programlama eğitimi ilköğretim düzeyinden başlayarak zorunlu hale getirmiştir. Bu çerçevede, öğrencilerin basit algoritma yazma ve kodlama becerilerinden başlayarak, kendi yazılım projelerini geliştirebilecek seviyeye ulaşmaları hedeflenmiştir (Hubwieser vd., 2015). Benzer şekilde, Finlandiya 2016 yılında, ilköğretim müfredatına algoritmik düşünme ve kodlama becerilerini kazandırmaya yönelik bir eğitim reformu gerçekleştirmiştir (Heintz vd., 2016). Bu reformlar, öğrencilerin sadece teknik beceriler kazanmasını değil, aynı zamanda disiplinlerarası düşünme ve problem çözme süreçlerinde daha etkin olmasını sağlamayı amaçlamıştır.

Amerika Birleşik Devletleri’nde Programlama Eğitimi

Amerika Birleşik Devletleri’nde programlama eğitimi, son yıllarda ulusal müfredatlarda daha fazla yer bulmaya başlamıştır. Özellikle Code.org gibi kuruluşların başlattığı "Hour of Code" girişimi, dünya genelinde milyonlarca öğrenciye ulaşarak programlama eğitiminin yaygınlaşmasına katkı sağlamıştır. Bu tür girişimler, öğrencilerin teknik beceriler edinmesinin yanı sıra, programlama eğitiminin sosyal ve ekonomik eşitliği destekleyen bir araç olarak kullanılmasını da hedeflemektedir. Örneğin, Grover ve Pea (2013), programlama eğitiminin öğrencilerin problem çözme ve eleştirel düşünme becerilerini geliştirdiğini vurgulamaktadır. Ayrıca, Resnick ve arkadaşları (2009), Scratch gibi araçların kullanımıyla öğrencilerin yaratıcı düşünme ve iş birliği yeteneklerinin artırılabilirliğini belirtmektedir.

Asya'da Programlama Eğitimi

Japonya ve Güney Kore gibi Asya ülkeleri, programlama eğitiminde ileri düzey uygulamalar geliştirmiştir. Japonya, 2020 yılında ilköğretim müfredatına programlama eğitimini zorunlu bir bileşen olarak eklemiştir. Bu reform, öğrencilerin algoritmik düşünme ve problem çözme becerilerini kazanmalarını hedeflerken, aynı zamanda yapay zekâ ve otomasyon teknolojilerine adapte olma kapasitelerini artırmayı amaçlamaktadır (Nakajima, 2020). Güney Kore ise, "Smart Education" girişimi kapsamında dijital öğrenme materyallerini ve programlama araçlarını tüm eğitim düzeylerinde yaygınlaştırmayı hedeflemiştir.

Programlama eğitiminde kullanılan araçlar, öğrencilerin bu becerileri daha etkili bir şekilde öğrenmesini sağlamaktadır. Scratch, Tynker ve Blockly gibi görsel programlama araçları, dünya genelinde ilköğretim ve ortaöğretim düzeyinde programlama eğitiminde yaygın olarak kullanılmaktadır. Bu araçlar, özellikle küçük yaş gruplarındaki öğrenciler için algoritmik düşünmeyi somutlaştırarak öğrenme sürecini kolaylaştırmaktadır. Görsel programlama araçlarının yanı sıra, Python ve Java gibi metin tabanlı programlama dilleri de ileri düzey öğrenciler için yaygın olarak tercih edilmektedir (Fessakis vd., 2013).

Programlama eğitimi, öğrencilerin algoritmik düşünme ve problem çözme becerilerini geliştirmede kritik bir rol oynamaktadır. Bu süreçte kullanılan araçlar, öğrencilerin yaş ve deneyim seviyelerine göre çeşitlilik göstermektedir. Özellikle Scratch gibi blok tabanlı programlama araçları, kullanıcı dostu arayüzleri ve görsel özellikleri sayesinde, ilköğretim seviyesindeki öğrencilerin programlamayı kavramsal olarak anlamalarını kolaylaştırmaktadır (Resnick vd., 2009). Scratch, öğrencilerin problem çözme, yaratıcı düşünme ve mantıksal düşünme becerilerini geliştirmede etkili bir araç olarak kabul edilmektedir (Fessakis vd., 2013).

Öte yandan, Python gibi metin tabanlı programlama dilleri, ileri düzey öğrenciler için tercih edilmektedir. Python, açık kaynak kodlu yapısı ve geniş kullanım alanı sayesinde, öğrencilerin daha karmaşık programlama becerileri edinmelerine olanak tanımaktadır. Bu araçların eğitimde kullanımı, öğrencilerin programlamaya yönelik tutumlarını olumlu yönde etkileyerek, onların akademik başarılarını artırmaktadır (Fessakis vd., 2013; Grover & Pea, 2013).

Sonuç olarak, programlama eğitiminde kullanılan araçların, öğrencilerin yaş ve deneyim seviyelerine uygun olarak seçilmesi, onların programlama becerilerini geliştirmede önemli bir etkiye sahiptir. Görsel programlama araçları, küçük yaş gruplarındaki öğrenciler için öğrenmeyi kolaylaştırırken, metin tabanlı programlama dilleri, ileri düzey öğrencilerin daha derinlemesine bilgi edinmelerine olanak tanımaktadır.

Türkiye’de Programlama Eğitimi

Türkiye’de programlama eğitimi, dijitalleşen dünyanın ihtiyaçlarına cevap verebilmek ve bireyleri bilgi-işlemsel düşünme becerileriyle donatmak amacıyla eğitim politikalarında önemli bir yer edinmiştir. Milli Eğitim Bakanlığı (MEB), bu bağlamda 2013 yılında Bilişim Teknolojileri ve Yazılım dersini ortaokul müfredatına eklemiş ve 2018 yılından itibaren bu dersi tüm ilköğretim kurumlarında uygulanabilir hale getirmiştir (MEB, 2018). Programlama eğitiminin erken yaşlarda verilmesi, öğrencilerin algoritmik düşünme, problem çözme ve yaratıcı düşünme becerilerini geliştirme açısından kritik bir adım olarak görülmektedir.

Türkiye’deki programlama eğitimi uygulamaları, çoğunlukla Scratch gibi blok tabanlı görsel programlama araçlarıyla yürütülmektedir. Scratch, öğrencilerin soyut algoritmik kavramları somutlaştırmasına ve kodlama süreçlerini daha erişilebilir hale getirmesine olanak tanır. Yapılan araştırmalar, Scratch temelli programlama eğitiminin öğrencilerin derslere yönelik tutumlarını olumlu yönde etkilediğini ve motivasyonlarını artırdığını göstermektedir (Kalelioğlu & Gülbahar, 2014). Bununla birlikte, Python gibi metin tabanlı programlama dilleri, daha ileri seviyede teknik beceriler kazandırmak amacıyla lise ve üniversite düzeyinde sıklıkla kullanılmaktadır.

Türkiye’de programlama eğitimi üzerine yapılan akademik çalışmaların betimsel analizleri, bu alandaki eğilimleri ve eksiklikleri ortaya koymaktadır. Eryılmaz ve Deniz (2019) tarafından yapılan bir araştırma, 2008-2018 yılları arasında Türkiye’de programlama eğitimi üzerine yapılan akademik çalışmaların büyük bir kısmının ortaokul düzeyine odaklandığını ve Scratch gibi araçların en sık kullanılan programlama ortamları arasında yer aldığını belirtmiştir. Bununla birlikte, erken yaş gruplarına yönelik uygulamaların sınırlı olduğu ve öğretmenlerin programlama öğretimi konusundaki yeterliliklerini artırmaya yönelik destekleyici programların eksik olduğu ifade edilmiştir. Bu bulgular, programlama eğitiminde daha kapsamlı ve kapsayıcı yaklaşımlar geliştirilmesi gerektiğini ortaya koymaktadır. Bunun yanı sıra, programlama eğitiminin Türkiye’deki yaygınlaştırılma çabaları arasında Kodlama Haftası etkinlikleri ve TÜBİTAK projeleri önemli bir yer tutmaktadır. Kodlama Haftası etkinlikleri, öğrencilerin programlamaya olan ilgisini artırmak ve onları yaratıcılığa teşvik etmek amacıyla düzenlenmektedir. TÜBİTAK tarafından desteklenen projeler ise, programlama eğitiminin bilimsel temellerini güçlendirme ve yenilikçi öğretim materyalleri geliştirme amacını taşımaktadır.

Türkiye’de programlama eğitimi ile ilgili en dikkat çekici sorunlardan biri, teknolojik altyapı eksiklikleridir. Okullarda bilgisayar laboratuvarlarının yetersizliği, internet erişiminin sınırlı olması ve öğretim materyallerinin güncel olmaması, programlama eğitiminin etkin bir

şekilde uygulanmasını zorlaştırmaktadır. Ayrıca, öğretmenlerin bu alandaki bilgi ve becerilerinin artırılmasına yönelik hizmet içi eğitim programlarının yeterli düzeyde olmaması, programlama eğitiminin yaygınlaştırılmasında bir diğer engel olarak görülmektedir (Eryılmaz & Deniz, 2019).

Sonuç olarak, Türkiye’de programlama eğitimi, uluslararası standartlara ulaşma yolunda önemli adımlar atmış olmakla birlikte, öğretim yöntemleri, teknolojik altyapı ve öğretmen eğitimi konularında hâlâ geliştirilmeye ihtiyaç duymaktadır. Bu bağlamda, yapay zekâ destekli programlama araçlarının entegrasyonu, bireysel öğrenme farklılıklarını dikkate alan kişiselleştirilmiş öğretim yaklaşımlarının uygulanması ve teknoloji odaklı eğitim politikalarının geliştirilmesi, programlama eğitiminin etkinliğini artırmak için potansiyel çözümler sunmaktadır.

Yapay Zeka

Yapay zekâ, modern teknolojinin ve bilimsel araştırmaların öncü alanlarından biri olarak insan zekâsını taklit eden makinelerin geliştirilmesini hedeflemektedir. Bu teknoloji, insan beyninin düşünme, öğrenme, karar verme ve problem çözme gibi bilişsel işlevlerini modelleyen bilgisayar sistemleri ve robotlar üretmeyi amaçlar. Gökçe (2023) tarafından yapılan bir tanımlamada, yapay zekâ, zeki makineler ve bilgisayar programları tasarlama bilimi ve mühendisliği olarak ifade edilmiştir. Bu bağlamda, yapay zekâ, bireysel ve toplumsal hayatı dönüştürme potansiyeline sahip geniş bir uygulama yelpazesi sunmaktadır. YZ’nin gelişimi, makine öğrenimi, derin öğrenme, doğal dil işleme ve bilgisayarlı görme gibi alt alanlarla desteklenmiş ve sağlık, eğitim, mühendislik, lojistik ve sanat gibi birçok sektörde uygulanabilir hale gelmiştir.

YZ sistemleri, insan benzeri bilişsel işlevler sergilemek için büyük veri analitiği, algoritmalar ve yüksek işlem gücü kullanan makine öğrenimi modellerine dayanır. Bu sistemler, gözetimli, gözetimsiz ve pekiştirmeli öğrenme yöntemlerini kullanarak karmaşık veri yapılarından anlam çıkarabilir ve bu bilgiyi yeni durumlara uygulayabilir (Goodfellow vd., 2016). Örneğin, derin öğrenme algoritmaları, yapay sinir ağları kullanarak görüntü tanıma, konuşma analizi ve metin oluşturma gibi görevlerde olağanüstü başarılar elde etmiştir (LeCun vd., 2015).

Eğitim alanında YZ, bireyselleştirilmiş öğrenme ortamlarının tasarımı, öğrenci performansının analizi ve öğretim materyallerinin otomatik oluşturulması gibi birçok alanda yenilikçi çözümler sunmaktadır. Özellikle adaptif öğrenme sistemleri, öğrencilerin bireysel ihtiyaçlarına göre dinamik olarak uyarlanabilmekte ve böylece öğrenme sürecini optimize

edebilmektedir (Holmes vd., 2019). Ayrıca, YZ destekli chatbotlar ve sanal öğretmenler, öğrencilere ders dışı destek sunarak öğrenme deneyimini zenginleştirmektedir.

YZ'nin potansiyel faydalarının yanı sıra, etik ve toplumsal zorluklar da bu teknolojinin yaygınlaşmasıyla daha fazla gündeme gelmiştir. Örneğin, YZ sistemlerinin karar alma süreçlerindeki şeffaflık eksikliği, veri gizliliği ve önyargılı algoritmalar, bu teknolojinin güvenilirliği ve adil kullanımı konusunda endişeler yaratmaktadır (Binns, 2018). Ayrıca, YZ'nin hızlı gelişimi, iş gücü piyasasında değişikliklere yol açarak bazı mesleklerin otomasyon tehdidiyle karşı karşıya kalmasına neden olmuştur. Bu bağlamda, YZ'nin toplumsal etkilerinin dikkatle ele alınması ve bu teknolojinin etik kullanımını destekleyen politikaların geliştirilmesi gerekmektedir.

Tarihi perspektiften bakıldığında, yapay zekâ kavramı ilk kez 1956'da Dartmouth Konferansı'nda ortaya atılmıştır. Bu dönemde Alan Turing'in "Makineler düşünebilir mi?" sorusu, yapay zekâ araştırmalarının felsefi temelini oluşturmuştur. Daha sonra yapay sinir ağları, bulanık mantık, genetik algoritmalar ve uzman sistemler gibi çeşitli teknikler geliştirilerek, yapay zekâ teknolojileri farklı alanlarda uygulanabilir hale gelmiştir. Gökçe (2023), yapay zekânın uzman sistemler, bilgisayarlı görme, konuşma tanıma ve robotik gibi alt disiplinlerle sürekli evrim geçirdiğini vurgulamaktadır.

Eğitimde, yapay zekâ destekli kişiselleştirilmiş öğrenme ortamları, öğrencilerin bireysel öğrenme hızlarına ve ihtiyaçlarına uyarlanmış ders materyalleri sunarak öğrenme süreçlerini optimize etmektedir. Örneğin, doğal dil işleme teknolojileri, öğrencilere gerçek zamanlı geri bildirim sunmak için kullanılmakta, sanal öğretmenler ise öğretim materyallerini dinamik bir şekilde uyarlamaktadır.

Eğitimde Yapay Zeka Uygulamaları

Eğitimde yapay zekâ (YZ) uygulamaları, öğrenme ve öğretme süreçlerini kişiselleştirme, bireysel ihtiyaçlara göre optimize etme ve eğitimdeki verimliliği artırma potansiyeli taşıyan teknolojiler olarak hızla yaygınlaşmaktadır. Meço ve Coştu'nun (2022) araştırması, Türkiye'deki yapay zekâ temelli eğitim çalışmalarını inceleyerek, bu alanda yapılan tez ve makalelerin içerik analizi yöntemiyle sistematik bir şekilde değerlendirildiğini ortaya koymaktadır. Çalışmaya göre, yapay zekâ teknolojilerinin eğitimde uygulanması, özellikle kişiselleştirilmiş öğrenme, bireysel performans takibi ve ders içeriklerinin otomatik hazırlanması gibi alanlarda etkili sonuçlar sağlamaktadır. Araştırmada, yapay zekânın eğitimde kullanımı, çeşitli yöntemlerle ele alınmıştır. Örneğin, öğretmenlerin iş yükünü hafifletmek amacıyla not verme gibi temel etkinliklerin otomatikleştirilmesi, veri analizi ve karar destek

sistemleri oluşturulması gibi uygulamalar, eğitimde YZ'nin doğrudan faydalarına örnek teşkil etmektedir. Bunun yanı sıra, sanal öğrenme ortamlarının oluşturulması, öğrenci etkileşimlerinin artırılması ve bireysel öğrenme yollarının desteklenmesi gibi avantajlar, YZ'nin eğitimdeki rolünü güçlendirmektedir (Meço & Coştu, 2022).

Eğitimde YZ'nin etkisi yalnızca sınıf ortamıyla sınırlı kalmamış, okul dışı öğrenme ortamlarına da yayılmıştır. Örneğin, Chatbot gibi yapay zekâ destekli araçlar, öğrencilerin bireysel öğrenme ihtiyaçlarına yanıt vermekte ve öğretmenlerin rehberlik süreçlerini desteklemektedir. Bunun yanı sıra, akıllı öğretici sistemler ve erken uyarı sistemleri, öğrencilerin öğrenme güçlüklerini erken aşamada tespit ederek, daha etkili müdahaleler yapılmasını sağlamaktadır. Ayrıca, bu teknolojiler, öğrencilerin performanslarını analiz ederek, öğrenme deneyimlerini daha etkili ve eğlenceli hale getirebilecek şekilde uyarlamaktadır (Meço & Coştu, 2022). Bununla birlikte, eğitimde yapay zekâ kullanımına ilişkin bazı sınırlılıklar da dikkat çekmektedir. Örneğin, verilerin gizliliği ve güvenliği, YZ tabanlı sistemlerin etik kullanımına yönelik kaygıları artırmaktadır. Ayrıca, bu sistemlerin karar verme süreçlerindeki şeffaflık eksikliği ve önyargılı algoritmalar, eğitimde eşitlik ve adalet ilkelerini tehdit edebilmektedir. Bu bağlamda, YZ'nin eğitimde daha yaygın ve etkili bir şekilde kullanılabilmesi için etik yönergelerin geliştirilmesi ve teknolojinin insan odaklı bir perspektifle ele alınması gerektiği vurgulanmaktadır (Meço & Coştu, 2022).

Eğitimde yapay zekâ (YZ) uygulamaları, eğitim süreçlerini bireyselleştirme, değerlendirme süreçlerini iyileştirme ve öğrenmeyi daha erişilebilir kılma konularında devrim yaratmıştır. Łodzikowski ve diğerleri (2023), yapay zekânın özellikle generatif modeller aracılığıyla eğitim sistemlerini dönüştürdüğünü belirtmektedir. Generatif yapay zekâ, metin, görsel ve çoklu ortam içeriklerinin oluşturulmasında etkili bir şekilde kullanılmakta ve öğrenme materyallerini öğrencilerin bireysel ihtiyaçlarına göre uyarlamaktadır.

YZ'nin eğitimde en dikkat çekici uygulama alanlarından biri, bireyselleştirilmiş öğrenme ortamlarıdır. Akıllı öğretici sistemler, öğrencilerin öğrenme hızlarını ve ihtiyaçlarını analiz ederek dinamik bir şekilde uyum sağlayan içerikler sunar. Örneğin, ChatGPT gibi metin tabanlı modeller, öğrencilerin açık uçlu sorulara verdikleri cevapları analiz ederek anında geri bildirim sağlamaktadır. Bunun yanı sıra, otomatik değerlendirme araçları, yazılı cevapların içeriğini analiz ederek hem öğretmenlerin iş yükünü azaltmakta hem de öğrencilere anında geri bildirim sunarak öğrenme sürecini hızlandırmaktadır (Łodzikowski vd., 2023).

YZ'nin sunduğu bir diğer yenilik, eğitimde çoklu modalite kullanımını destekleyen generatif modellerdir. Metinden görsele dönüştürücü modeller, karmaşık teorik kavramların görsel olarak daha anlaşılır hale getirilmesine olanak tanır. Benzer şekilde, metinden sese

dönüştürücü modeller, dil öğrenimi ve konuşma pratiği gibi alanlarda etkili araçlar sunmaktadır. Bu tür araçlar, öğrenci etkileşimlerini artırmakta ve öğrenmeyi daha etkili bir hale getirmektedir. Ancak YZ'nin eğitimdeki kullanımına ilişkin bazı etik ve teknik sınırlamalar da bulunmaktadır. Verilerin gizliliği, algoritmik önyargılar ve yapay zekânın karar süreçlerindeki şeffaflık eksikliği, bu teknolojilerin geniş ölçekli uygulanabilirliğini sınırlamaktadır. Ayrıca, generatif yapay zekâ tarafından üretilen içeriklerin doğruluğu konusunda güven sorunları ortaya çıkmaktadır. Bu bağlamda, Łodzikowski vd. (2023), yapay zekâ sistemlerinin güvenilirliğini artırmak için veri kalitesinin iyileştirilmesi ve etik yönergelerin oluşturulması gerektiğini vurgulamaktadır.

Eğitimde yapay zekâ uygulamaları, Türkiye bağlamında da dikkat çekici bir şekilde ilerlemektedir. Tekin (2023), Türkiye'deki yapay zekâ uygulamalarının eğitsel çıktılarının son 20 yılda önemli bir artış gösterdiğini ve özellikle son beş yıl içinde yayımlanan çalışmaların bu alandaki araştırma hacmini yüzde 64 oranında oluşturduğunu ifade etmektedir. Yapay zekânın eğitim süreçlerindeki rolleri, öğretim ortamlarını geliştirme, eğitim süreçlerini iyileştirme ve bireysel öğrenme deneyimlerini zenginleştirme üzerine odaklanmaktadır. Bu uygulamalar, öğrencilerin akademik başarısını artırmak ve öğretmenlerin öğretim süreçlerini daha verimli hale getirmek için kritik bir araç haline gelmiştir.

Tekin (2023) tarafından yapılan içerik analizi, yapay zekâ uygulamalarının Türkiye'deki eğitim araştırmalarında, özellikle lisans düzeyinde öğrenim gören öğrencilerle gerçekleştirilen çalışmalarla öne çıktığını ortaya koymaktadır. Bunun yanı sıra, genel eğitim uygulamalarında yapay zekâ teknolojilerinin yaygın olarak kullanıldığı, ancak özel öğrenme alanlarına yönelik çalışmaların sınırlı olduğu vurgulanmıştır. Özellikle bilgisayar bilimleri ve fen eğitimi gibi disiplinlerde yapay zekâ destekli araçların kullanımı dikkat çekmektedir. Bununla birlikte, eğitim ortamlarının dijitalleştirilmesi ve kişiselleştirilmiş öğrenme materyallerinin oluşturulması, Türkiye'deki araştırmaların önemli bir yönünü oluşturmaktadır (Tekin, 2023).

Yapay zekânın eğitimdeki uygulamalarına ilişkin tartışmalar, bu teknolojilerin yalnızca eğitim sürecini desteklemekle kalmayıp aynı zamanda etik sorunları da beraberinde getirdiğini göstermektedir. Veri gizliliği, algoritmik önyargılar ve karar verme süreçlerindeki şeffaflık eksikliği, eğitimde yapay zekâ uygulamalarının sürdürülebilirliği açısından önemli konular olarak öne çıkmaktadır. Bu bağlamda, Tekin (2023), Türkiye'deki yapay zekâ temelli çalışmaların bu tür etik kaygılara yönelik daha fazla odaklanması gerektiğini önermektedir.

Eğitimde yapay zekâ (YZ) uygulamaları, öğrenme süreçlerini bireyselleştirme, öğretim materyallerini optimize etme ve eğitimcilere yenilikçi araçlar sunma konularında önemli bir

dönüşüm yaratmaktadır. Ogunleye vd. (2024), YZ'nin özellikle üretken yapay zekâ araçlarıyla eğitim sektöründe derin etkiler yarattığını ve bu teknolojilerin öğretim uygulamaları ile değerlendirme süreçlerini dönüştürdüğünü vurgulamaktadır. Araştırma, ÜYZ'nin öğretim tasarımı, kişiselleştirilmiş öğrenme materyalleri oluşturma ve öğrenme süreçlerine yönelik geri bildirim mekanizmalarını güçlendirmede büyük bir potansiyele sahip olduğunu ortaya koymaktadır. Generatif yapay zekâ, özellikle ChatGPT gibi büyük dil modellerine dayalı araçlarla, yüksek öğretimde eğitimciler ve öğrenciler için kapsamlı avantajlar sunmaktadır. Ogunleye vd. (2024) sistematik incelemesinde, ÜYZ'nin yazılım mühendisliği eğitimi gibi alanlarda ders içeriklerini geliştirme, kodlama pratiklerini destekleme ve öğrencilerin performans tahminlerini yapma konularında etkili olduğu belirtilmiştir. Bu teknolojilerin, öğrenci başarılarını artırmak ve ders bırakma oranlarını azaltmak için erken müdahaleler sağlamada önemli bir rol oynayabileceği ifade edilmiştir. Araştırmada ayrıca, ÜYZ'nin akademik yazım, sınav değerlendirme ve kişiselleştirilmiş öğretim gibi alanlarda mevcut ve potansiyel uygulamalarına dikkat çekilmektedir. Bunun yanı sıra, öğrenci algıları üzerinde yapılan çalışmalarda, ÜYZ araçlarının daha etkili öğrenme ortamları oluşturduğu ve öğrencilerin problem çözme ile yaratıcı düşünme becerilerini geliştirdiği görülmüştür. Ancak, Ogunleye vd. (2024), bu teknolojilerin akademik dürüstlük, etik ve veri gizliliği gibi konularda kaygılara yol açabileceğini de belirtmektedir. Bu bağlamda, ÜYZ'nin etkili bir şekilde kullanılabilmesi için disiplinler arası iş birlikleri ve yeni politika çerçevelerine ihtiyaç duyulduğu vurgulanmaktadır.

Eğitimde yapay zekâ uygulamaları, öğretim ve öğrenme süreçlerini kökten dönüştürme potansiyeline sahiptir. Kutlucan ve Seferoğlu (2024), özellikle ChatGPT gibi generatif yapay zekâ araçlarının eğitimde sağladığı fırsatlar ve karşılaşılan zorluklar üzerinde durmaktadır. Çalışmalarında, ChatGPT'nin öğretim süreçlerini destekleyen önemli bir araç olarak, eğitimde fırsat eşitliği sağlama ve bireyselleştirilmiş öğrenme deneyimlerini artırma potansiyeline sahip olduğu ifade edilmektedir.

ChatGPT'nin sunduğu en önemli katkılar arasında, öğretmenlere destek sunarak ders içerikleri oluşturma, geri bildirim mekanizmalarını iyileştirme ve öğrencilerin bireysel ihtiyaçlarına yönelik çözümler sunma kapasitesi yer almaktadır. Özellikle KEFE (SWOT) analiziyle değerlendirilen bu teknolojinin güçlü yönleri arasında, eğitim süreçlerini optimize etme ve erişilebilirliği artırma özellikleri bulunmaktadır. Ancak, etik ve intihal gibi konuların, ChatGPT'nin eğitimdeki kullanımında dikkate alınması gereken zayıf yönler olarak öne çıktığı belirtilmiştir (Kutlucan & Seferoğlu, 2024). Makale ayrıca, ChatGPT'nin eğitimde politik, ekonomik, sosyal ve teknolojik etkilerini PEST analizi ile detaylandırmıştır. Bu analiz,

ChatGPT'nin eğitim politikaları üzerindeki etkisini, teknolojinin ekonomik uygulanabilirliğini, sosyolojik etkilerini ve teknolojik gelişimlere katkısını ortaya koymaktadır. Özellikle teknoloji ile uyumlu çalışabilme kapasitesi, ChatGPT'nin birçok eğitim kurumunda yaygın bir şekilde benimsenmesine olanak tanımaktadır. Bununla birlikte, ChatGPT'nin kullanımıyla ilgili bazı engeller de dikkat çekmektedir. Örneğin, yanlış bilgi üretme potansiyeli, mahremiyet sorunları ve sosyal eşitsizlikler, yapay zekâ uygulamalarının geniş çaplı kullanımını sınırlayan faktörler arasındadır. Kutlucan ve Seferoğlu (2024), bu sorunların üstesinden gelmek için bilinçli politikaların geliştirilmesi gerektiğini vurgulamaktadır.

Eğitimde yapay zekâ, bireyselleştirilmiş öğrenme ortamlarının oluşturulmasından, öğretmenlerin pedagojik süreçleri optimize etmesine kadar pek çok alanda dönüşüm yaratmaktadır. Körükçüoğlu ve Ata (2023) tarafından yapılan bir çalışmada, yapay zekâ teknolojilerinin öğrencilerin öğrenme süreçlerine olan katkısına ve öğretmenlerin üzerindeki idari yükleri azaltma potansiyeline dikkat çekilmiştir. Özellikle akıllı öğretici sistemler ve diyalog tabanlı uygulamalar, öğrencilere özel öğrenme modülleri sunarak öğrenme sürecini bireyselleştirmektedir.

Uzman sistemlerin eğitimdeki rolü, öğrencilere bireyselleştirilmiş geribildirim sağlama ve öğretmenlerin rehberlik görevlerini destekleme kapasitesini artırmaktadır. Holmes vd. (2023) çalışmalarında vurgulandığı üzere, bu tür sistemler öğrenci verilerini analiz ederek ilerleme süreçlerini değerlendirebilmekte ve gerekli müdahaleleri öneren algoritmalarla öğrenme deneyimini zenginleştirmektedir. Ek olarak, yapay zekâ temelli uygulamalar, öğrencilerin eğitime erişimindeki eşitsizlikleri azaltma potansiyeline sahiptir. Murphy (2019), kırsal bölgelerdeki öğrencilerin bu teknolojiler sayesinde eğitime daha kolay ulaşabildiğini ve eğitimde fırsat eşitliğine katkı sağladığını belirtmektedir. Ancak, teknolojik altyapı eksiklikleri ve öğretmenlerin yeterlilik seviyelerindeki farklılıklar, bu sürecin verimliliğini etkileyebilmektedir.

Sonuç olarak, eğitimde yapay zekâ uygulamaları, öğrenme ve öğretme süreçlerini dönüştürme potansiyeline sahiptir. Ancak, bu dönüşümün sürdürülebilir ve etik bir çerçevede gerçekleşmesi, teknolojiye ilişkin bilinçli politikaların oluşturulması ve eğitimcilerin bu alanda yeterliliklerinin artırılmasıyla mümkün olacaktır.

Eğitimde Üretken Yapay Zeka

Üretken yapay zekâ (ÜYZ), öğrenme süreçlerinde karmaşık veri yapılarından örüntüler çıkararak yeni, orijinal ve yaratıcı içerikler oluşturma yeteneğine sahip sistemlerin geliştirilmesiyle ilgilenen bir yapay zekâ alt alanıdır. ÜYZ, özellikle Generative Adversarial

Networks (GANs) ve Transformer modelleri gibi yenilikçi teknolojiler aracılığıyla görsel, metinsel ve işitsel içerik oluşturma kapasitesini olağanüstü bir şekilde genişletmiştir (Goodfellow vd., 2014). Bu teknoloji, sanat, medya, eğitim, sağlık ve eğlence gibi çeşitli sektörlerde önemli dönüşümlere yol açmıştır.

Üretken yapay zekâ, temelde iki modelin etkileşimi üzerine kuruludur: bir "üretici" (generator) ve bir "ayırt edici" (discriminator). Üretici model, rastgele bir girdi alarak yeni bir veri örneği oluştururken, ayırt edici model bu örneğin gerçek mi yoksa sahte mi olduğunu değerlendirmeye çalışır. Bu süreç, her iki modelin de performansını artıracak şekilde tekrarlanır ve sonuçta gerçek veri dağılımına benzer yüksek kaliteli içerikler üretilir (Radford vd., 2015). Transformer tabanlı modeller ise doğal dil işleme ve metin oluşturma görevlerinde etkileyici sonuçlar elde etmiştir. Özellikle GPT (Generative Pre-trained Transformer) mimarisi, metin oluşturma, çeviri ve özetleme gibi görevlerde insan benzeri performans sergilemektedir (Brown vd., 2020).

ÜYZ'nin eğitimdeki potansiyeli, bireyselleştirilmiş öğrenme ortamlarının geliştirilmesi, öğretim materyallerinin otomatik oluşturulması ve öğrencilerin yaratıcı düşünme becerilerinin desteklenmesi gibi yenilikçi uygulamalarla kendini göstermektedir. Örneğin, ÜYZ tabanlı sistemler, öğrenci düzeyine göre uyarlanmış test soruları oluşturabilir, öğrenci başarısını analiz edebilir ve öğretim yöntemlerini optimize edebilir. Ayrıca, ÜYZ'nin sanal öğretmenler ve interaktif öğrenme materyalleri tasarımı gibi uygulamaları, öğrenme süreçlerini daha etkili ve eğlenceli hale getirmektedir (Holmes vd., 2019). Bununla birlikte, ÜYZ'nin yarattığı etik ve toplumsal zorluklar da göz ardı edilemez. Derin sahtecilik (deepfake) teknolojileri, veri güvenliği ve telif hakları gibi konular, ÜYZ'nin kontrolsüz kullanımında ortaya çıkabilecek risklere işaret etmektedir. Ayrıca, bu teknolojilerin karar alma süreçlerindeki önyargıları artırma potansiyeli, dikkatle ele alınması gereken bir sorundur (Binns, 2018). Bu bağlamda, ÜYZ'nin geliştirilmesi ve uygulanmasında etik standartların ve yasal düzenlemelerin hayata geçirilmesi büyük önem taşımaktadır.

Üretken yapay zekâ (ÜYZ), veri setlerinden öğrenerek bu veri setlerine benzer yeni ve özgün içerikler üretebilen teknolojiler geliştirmeyi amaçlayan bir yapay zekâ alt dalıdır. Bu teknolojinin en yaygın kullanılan uygulamalarından biri olan Üretken Çekişmeli Ağlar (GANs), hem akademik hem de endüstriyel alanda önemli yenilikler sunmaktadır. Gözet vd. (2023), GAN'ların karmaşık veri yapılarından örüntüler çıkararak yüksek kaliteli sentetik veri üretiminde önemli bir rol oynadığını vurgulamaktadır. GAN'ların temel çalışma prensibi, "üretici" ve "ayırt edici" olmak üzere iki modelin karşılıklı etkileşimi üzerine kuruludur. Bu iki

modelin ekifsmeli bir fekilde eęitilmesi, gerek veri setine benzer nitelikte sentetik veri rnekleri retilmesini saęlar.

GAN tabanlı modeller, yaratıcı ierik retiminde ıęır aıcı yenilikler sunmaktadır. rneęin, grsel sentezleme, grnt restorasyonu ve stil transferi gibi uygulamalarda olaęanst sonular elde edilmiřtir. Transformer tabanlı modeller ise doęal dil iřleme alanında etkili bir fekilde kullanılmaktadır. ChatGPT gibi modeller, insana benzer metin oluřturma, dil evirisi, zetleme ve yazım denetimi gibi birok grevde yksek bařarı gstermektedir. Bu modeller, aynı zamanda kullanıcıların yaratıcılıęını destekleyerek, problem zme ve ierik geliřtirme srelerine aktif bir fekilde katkıda bulunmaktadır (nal, 2023).

Eęitim alanında YZ, ęrencilerin ęrenme deneyimlerini zenginleřtiren ve ęretim srelerini kolaylařtıran bir ara olarak ne ıkmaktadır. Kiřiselleřtirilmiř ęrenme materyalleri oluřturma, ęrencilerin bireysel ęrenme hızına uyum saęlama ve gerek zamanlı geri bildirim sunma gibi avantajları, YZ'yi eęitimde vazgeilmez bir teknoloji haline getirmiřtir. Bununla birlikte, YZ'nin yaratıcı ierik retimindeki yetenekleri, ęretmenlerin roln ęrenme deneyimlerini tasarlayıcı ve ynlendirici bir pozisyona dnřtrmektedir (Bozkurt, 2023).

Bozkurt (2023), YZ'nin eęitimde faydalı bir fekilde kullanılabilmesi iin yapay zek okuryazarlıęı ve komut mhendislięi gibi becerilerin geliřtirilmesi gerektięini belirtmektedir. Bu baęlamda, eęitimciler ve ęrenciler, YZ'nin potansiyelini tam anlamıyla kullanabilmek iin teknolojinin sınırlılıklarını ve olanaklarını anlamalıdır. retken yapay zek (YZ), insan yaratıcılıęına benzer fekilde yeni ve anlamlı ierikler oluřturabilme kapasitesine sahip algoritmalarından oluřur. Waseem ve dięerleri (2024), YZ'nin yazılım geliřtirme srelerinde nemli bir rol oynadıęını ve zellikle kod retimi, hata tespiti, ve dokmantasyon gibi alanlarda verimlilięi artırdıęını vurgulamaktadır. alıřmada, ChatGPT gibi YZ aralarının, yazılım mhendislięi eęitiminde nemli beceri aıklarını kapatma ve ęrencilerin temel yazılım geliřtirme kavramlarını daha hızlı anlamalarına olanak saęlama potansiyeline sahip olduęu belirtilmiřtir.

retken yapay zek (YZ) teknolojileri, programlama eęitiminde ęrencilere eřitli aılardan destek saęlayarak ęrenme srelerini zenginleřtirmektedir. zellikle ChatGPT gibi geliřmiř dil modelleri, yazılım geliřtirme yařam dngsnn farklı ařamalarında etkin bir fekilde kullanılmaktadır.

Gereksinim analizi ve tasarım ařamalarında, ChatGPT, doęru gereksinimlerin belirlenmesi ve tasarım srelerinin daha aık ve leklenebilir hale getirilmesinde yardımcı

olmaktadır. Doğal dil işleme yetenekleri sayesinde, kullanıcıların ihtiyaçlarını analiz ederek uygun yazılım çözümleri önerme kapasitesine sahiptir (BlueMark Academy, 2023).

Kodlama aşamasında ise, ChatGPT'nin kod üretimi sırasında hata oranını azalttığı ve kod kalitesini iyileştirdiği belirtilmektedir. Öğrencilerin karmaşık kodlama sorunlarını çözmelerine yardımcı olarak problem çözme becerilerini geliştirmekte ve iş birliği süreçlerini kolaylaştırmaktadır (Bubeck vd., 2023). Ayrıca, ChatGPT, kodlama sırasında karşılaşılan hataların tespit edilmesi ve düzeltilmesinde de etkilidir. Hata ayıklama süreçlerinde öğrencilere rehberlik ederek, kodların doğru çalışmasını sağlamaktadır (Bilgi.net, 2024).

Geri bildirim ve değerlendirme süreçlerinde de ChatGPT, öğrencilerin yazdığı kodları analiz ederek anında geri bildirim sağlayabilmektedir. Bu sayede, öğrenciler hatalarını hızlıca fark edip düzeltebilmekte ve öğrenme süreçlerini daha verimli hale getirebilmektedirler (Derin & Akpınar, 2023).

Sonuç olarak, ÜYZ'nin programlama eğitiminde kullanımı, öğrencilerin öğrenme deneyimlerini zenginleştirerek, yazılım geliştirme süreçlerindeki becerilerini artırmaktadır. ChatGPT gibi araçlar, gereksinim analizinden kodlamaya, hata ayıklamadan geri bildirim sağlamaya kadar pek çok aşamada öğrencilere destek sunarak, programlama eğitiminde önemli bir rol oynamaktadır. Eğitimden sanata, sağlıktan iletişime kadar geniş bir alanda etkili olan ÜYZ, toplumsal fayda sağlamak için sorumlu bir şekilde kullanılmalıdır. Bu teknoloji, bireylerin ve toplumların yaratıcı potansiyelini ortaya çıkarmanın yanı sıra, dijital çağın gerekliliklerine uyum sağlamalarında güçlü bir araç olarak değerlendirilmektedir.

Üretken Yapay Zeka ile Programlama Eğitimi

Programlama eğitimi, öğrencilerin algoritmik düşünme, problem çözme ve kodlama becerilerini geliştirme amacını taşır. Ancak bu süreç, öğrencilerin genellikle karmaşık kodlama yapıları ve sözdizimleriyle karşılaşmasından dolayı zorluklar içermektedir. ÜYZ, öğrencilerin bu zorlukları aşmalarına yardımcı olan bir araç olarak devreye girmektedir. Örneğin, ChatGPT ve benzeri modeller, öğrencilerin kodlama görevlerinde anında geri bildirim almalarına olanak tanırken, aynı zamanda algoritmik mantıklarını geliştirmek için rehberlik sağlamaktadır (Brown vd., 2020). Bunun yanı sıra, ÜYZ tabanlı araçlar, örnek kodlar ve çözümler sunarak öğrencilerin programlama becerilerini pekiştirmelerine yardımcı olmaktadır. ChatGPT, OpenAI tarafından geliştirilen bir dil modeli olup, doğal dil işleme teknolojisi ile çalışmaktadır. Kullanıcıların yazılı girdilerini analiz ederek anlamlı çıktılar üretmekte ve programlama bağlamında, kod açıklamaları, hata düzeltmeleri ve algoritma önerileri gibi işlevler sunmaktadır. Özellikle eğitim ortamlarında, öğrencilerin kavramları anlamalarına ve karmaşık

kodlama sorunlarını çözmelerine yardımcı olmaktadır (OpenAI, 2021). GitHub Copilot, OpenAI Codex üzerine inşa edilmiş bir yapay zekâ aracıdır ve kullanıcıların kod yazma süreçlerini desteklemektedir. Bu araç, programlama dillerindeki kod örüntülerini analiz ederek, kullanıcıya tamamlayıcı kod önerileri sunar. Ayrıca, belirli görevler için optimize edilmiş kod blokları oluşturmakta ve yazılım geliştirme süreçlerini hızlandırmaktadır (GitHub, 2021). Özellikle öğrencilerin kodlama pratiklerinde sık yapılan hataları tespit ederek, düzeltmeler ve iyileştirme önerileri sağlamaktadır. Gemini, Google DeepMind tarafından geliştirilen Gemini, dil modelleri ile öğrenme süreçlerini birleştirerek kodlama ve problem çözme becerilerini geliştiren bir araçtır. Kullanıcıların yazılım geliştirme döngüsünde daha yaratıcı çözümler bulmasına olanak tanımakta ve özellikle karmaşık algoritmaların geliştirilmesinde rehberlik etmektedir (DeepMind, 2023). Replit tarafından sunulan Grok ise, özellikle kod öğrenimi ve hata ayıklama süreçlerini destekleyen bir üretken yapay zekâ platformudur. Kod yazmayı öğrenen öğrenciler için anında geri bildirim sunmakta, hataları belirlemekte ve kod yapısını daha verimli bir şekilde kavramalarını sağlamaktadır (Replit, 2023).

ÜYZ'nin programlama eğitimindeki bir diğer önemli katkısı, kişiselleştirilmiş öğrenme deneyimleri sunmasıdır. Bu teknolojiler, her öğrencinin bireysel öğrenme hızına ve ihtiyaçlarına uygun materyaller sağlayarak öğrenme sürecini optimize etmektedir. Ayrıca, ÜYZ'nin otomatik değerlendirme kapasitesi, öğrencilerin hatalarını daha hızlı fark etmelerini ve öğrenme süreçlerini iyileştirmelerini sağlamaktadır. Örneğin, Python veya Java gibi metin tabanlı programlama dillerinde hata ayıklama süreçleri, ÜYZ tabanlı araçlar sayesinde daha hızlı ve etkili bir şekilde gerçekleştirilmektedir (Radford vd., 2021). Bununla birlikte, ÜYZ'nin programlama eğitime entegrasyonunda bazı zorluklar da bulunmaktadır. Özellikle etik ve teknik sınırlamalar, bu teknolojilerin sorumlu bir şekilde kullanılmasını gerektirmektedir. Örneğin, öğrencilerin yalnızca ÜYZ araçlarına bağımlı hale gelmesi, kendi problem çözme becerilerini geliştirme süreçlerini olumsuz etkileyebilir. Ayrıca, ÜYZ araçlarının karar süreçlerindeki önyargılar ve doğruluk sorunları, öğrenme deneyimini olumsuz yönde etkileyebilir (Binns, 2018). Bu bağlamda, ÜYZ'nin eğitimde etkili bir şekilde kullanılabilmesi için bu teknolojilerin pedagojik yaklaşımlarla entegre edilmesi ve etik standartların belirlenmesi büyük önem taşımaktadır.(Binns, 2018).

Üretken yapay zekâ (ÜYZ), programlama eğitiminde sadece öğretim süreçlerini desteklemekle kalmaz, aynı zamanda öğrencilerin yazılım geliştirme becerilerini hızlandıran, derinleştiren ve kişiselleştiren bir yardımcı olarak önemli bir rol oynar. Tian vd. (2023) tarafından yapılan bir çalışma, özellikle ChatGPT'nin, programlama eğitimi için kod üretimi, program onarımı ve kod açıklamaları gibi görevlerde nasıl etkili bir şekilde kullanılabileceğini

incelemiştir. Araştırma, ChatGPT gibi büyük dil modellerinin (LLM'ler), yazılım mühendisliği eğitiminde öğretim süreçlerini otomatikleştirme ve öğrencilerin günlük yazılım geliştirme ihtiyaçlarına yönelik kapsamlı destek sağlama potansiyeline dikkat çekmektedir.

Yapılan bir diğer çalışmada, ChatGPT'nin kod üretimindeki başarısı, özellikle LeetCode benchmark testlerinde belirgin bir üstünlük göstermektedir. Kod üretiminde doğru ve verimli çözümler sağlama kapasitesi, öğrencilerin problem çözme süreçlerini desteklemekte ve öğretmenlerin değerlendirme yükünü hafifletmektedir. Ancak, uzun ve karmaşık açıklamalarda modelin dikkat dağılımı nedeniyle performansının düştüğü de vurgulanmıştır. Bu durum, prompt mühendisliğinin eğitimde ÜYZ teknolojilerinin etkili kullanımı için hayati önem taşıdığını göstermektedir (Tian vd., 2023). Ek olarak, ÜYZ'nin program onarımında öğrencilere sağladığı destek, manuel hata ayıklama sürecini hızlandırırken aynı zamanda yazılım güvenilirliğini artırmaktadır. Örneğin, ChatGPT'nin, QuixBugs gibi hata ayıklama veri setlerinde yüzde 84 başarı oranına ulaşarak mevcut yöntemlere kıyasla rekabetçi bir performans sergilediği bulunmuştur. Bununla birlikte, karmaşık sorunlarda, ÜYZ'nin sunduğu onarımların doğruluğu ve çeşitliliği artırılması gereken alanlar olarak belirtilmiştir.

Üretken yapay zekâ (ÜYZ) araçları, programlama eğitiminde öğrencilerin bilgiye erişimini kolaylaştırarak ve kodlama becerilerini geliştirmelerine yardımcı olarak devrim niteliğinde bir etki yaratmaktadır. Anagnostopoulos (2023), ChatGPT gibi büyük dil modellerinin, programlama eğitimi ve bilgisayar bilimi öğreniminde önemli avantajlar sunduğunu, ancak bazı etik ve teknik sınırlamaların da bu araçların kullanımını etkileyebileceğini vurgulamaktadır. ChatGPT'nin eğitimdeki rolü, kod üretimi, teknik dökümantasyon oluşturma ve doğal dil işleme gibi çok yönlü uygulamaları ile dikkat çekmektedir. Çalışmaya göre, ChatGPT'nin kodlama süreçlerindeki etkisi, öğrencilere temel programlama kavramlarını anlamaları için kişiselleştirilmiş rehberlik sağlamaktan, öğretmenlerin değerlendirme süreçlerini optimize etmeye kadar geniş bir yelpazede incelenmiştir. Özellikle, bu araçların yazılım geliştirme eğitiminde, kodlama pratiklerini desteklemek ve hataları analiz ederek geri bildirim sunmak gibi işlevleri, öğrenme süreçlerini hızlandırmaktadır. Bununla birlikte, ChatGPT'nin doğruluk ve bağlam anlamında sınırlılıkları, özellikle karmaşık programlama projelerinde dikkatli bir kullanım gerektirmektedir (Anagnostopoulos, 2023). Anagnostopoulos (2023), aynı zamanda ÜYZ'nin programlama eğitimindeki etik boyutlarına da dikkat çekmektedir. Özellikle, ChatGPT gibi araçların aşırı kullanımı, öğrencilerin problem çözme ve eleştirel düşünme becerilerinde azalmaya neden olabilir. Ayrıca, bu araçlar tarafından üretilen içeriğin doğruluğunun ve özgünlüğünün sürekli olarak denetlenmesi gerektiği ifade edilmektedir. Eğitimecilere düşen görev, bu teknolojilerin

destekleyici bir araç olarak kullanımını teşvik etmek ve öğrencilerin bağımsız düşünme becerilerini korumalarını sağlamaktır.

Üretken yapay zekâ (ÜYZ) araçları, programlama eğitiminde yalnızca öğretim süreçlerini desteklemekle kalmayıp aynı zamanda öğrencilerin programlama kavramlarını daha derinlemesine anlamalarına olanak tanıyan kişiselleştirilmiş öğrenme deneyimleri sunmaktadır. Ma vd. (2024), ChatGPT'nin Python programlama kurslarında nasıl kullanıldığını inceleyerek, bu tür araçların öğrencilerin öğrenme süreçlerini nasıl zenginleştirdiğini ortaya koymuştur. Sekiz haftalık bir program kapsamında yapılan çalışmada, öğrencilerin ChatGPT'yi hata ayıklama, kod üretimi ve kod açıklamaları gibi çeşitli alanlarda aktif olarak kullandıkları tespit edilmiştir. Araştırmada, ChatGPT'nin öğrencilerin programlama bilgilerini derinleştirdiği ve öğrenme sürecindeki zorlukları azaltarak motivasyonlarını artırdığı belirtilmiştir. Özellikle, ChatGPT'nin gerçek zamanlı geri bildirim sağlayarak, öğrencilerin kavram karmaşıklığını çözmesine yardımcı olduğu ve böylece daha etkin öğrenme sağladığı vurgulanmıştır. Ayrıca, öğrenciler, ChatGPT'nin sunduğu kod örneklerini ve hataları analiz ederek önerdiği çözümleri oldukça faydalı bulmuşlardır. Bu durum, öğrencilerin teorik bilgilerini pratik uygulamalara dönüştürme süreçlerinde önemli bir kolaylık sağlamaktadır (Ma vd., 2024). Bununla birlikte, çalışma, ChatGPT'nin programlama eğitimi üzerindeki bazı sınırlılıklarını da ortaya koymaktadır. Öğrencilerin bu tür araçlara aşırı bağımlı hale gelebileceği ve bu durumun problem çözme becerilerinin gelişimini olumsuz etkileyebileceği belirtilmiştir. Özellikle, ChatGPT'nin bazı durumlarda aşırı karmaşık veya ders kapsamı dışındaki çözümler sunması, öğrenme sürecinde kafa karışıklığına yol açabilmektedir. Bu nedenle, öğrencilerin ChatGPT'yi bilinçli ve destekleyici bir araç olarak kullanmaları önemlidir.

Üretken yapay zeka (ÜYZ) araçları, yazılım geliştirme eğitimi ve uygulamalarında hem bireysel öğrenme deneyimlerini destekleyen hem de takım çalışmalarını optimize eden yenilikçi çözümler sunmaktadır. Waseem vd. (2024), ChatGPT'nin yazılım geliştirme döngüsündeki farklı aşamalarda nasıl etkili bir şekilde kullanılabileceğini incelemiş ve bu araçların yazılım mühendisliği öğrencileri üzerindeki etkisini değerlendirmiştir. Çalışmada, ChatGPT'nin gereksinim analizi, yazılım tasarımı, kodlama, test etme ve dağıtım gibi süreçlerde öğrencilerin yetkinliklerini artırdığı ve proje çıktılarını geliştirdiği bulunmuştur. Araştırmaya göre, ChatGPT, gereksinim analizi aşamasında öğrencilerin zaman kazanmasına ve doğru ihtiyaçları belirlemelerine olanak sağlamaktadır. Bu, öğrencilerin karmaşık problemleri daha iyi anlamalarını ve çözüm önerilerini optimize etmelerini desteklemiştir. Tasarım aşamasında ise ChatGPT, sistem mimarilerini daha açık ve ölçeklenebilir bir şekilde tasarlamak için yönlendirici bir araç olarak kullanılmıştır. Ayrıca, kodlama süreçlerinde

öğrenciler, ChatGPT'nin sağladığı kod üretimi, hataların tespiti ve çözümü gibi özelliklerden yararlanarak hatasız ve kaliteli yazılım geliştirme süreçleri deneyimlemişlerdir (Waseem vd., 2024).

Waseem vd. (2024), ChatGPT'nin öğrenme eğrisi üzerindeki olumlu etkilerini de vurgulamıştır. Araç, öğrencilerin yazılım geliştirme konseptlerini daha hızlı anlamalarına ve yeni teknolojilere adapte olmalarına yardımcı olmuştur. Bununla birlikte, çalışmada, ChatGPT'nin kullanımı sırasında öğrencilerin karşılaştığı bazı zorluklar da ele alınmıştır. Özellikle, sistemin karmaşık gereksinimleri tam anlamıyla karşılayamaması ve bazı tasarım önerilerinin pratik olmaması gibi sınırlamalar, bu araçların etkili bir şekilde entegre edilmesi gerektiğini göstermektedir. Son olarak, ÜYZ'nin kod açıklama yetenekleri, programlama öğrenen bireyler için güçlü bir rehberlik aracı olarak hizmet etmektedir. ChatGPT, yanlış kod parçalarını dahi doğru bir şekilde özetleyebilmekte ve bu özelliğiyle eğitimde test tasarımı gibi alanlarda da kullanılma potansiyeli taşımaktadır. Bu bulgu, ÜYZ'nin eğitimdeki rolünün yalnızca teknik değil, aynı zamanda pedagojik olarak da değerli olduğunu göstermektedir.

Sonuç olarak, ChatGPT gibi ÜYZ araçları, yazılım mühendisliği eğitiminin farklı aşamalarında önemli avantajlar sunmaktadır. Ancak, bu araçların eğitimde daha geniş kapsamlı bir şekilde kullanılabilmesi için rehber ilkeler geliştirilmesi ve öğrencilere bu teknolojileri etkili bir şekilde kullanma becerilerinin kazandırılması gerekmektedir. Tablo 1'de üretken yapay zeka ve programlama eğitimi ile ilgili yapılan literatür taraması detaylarıyla sunulmuştur.

İlgili Çalışmalar

Tablo 1. *İlgili Araştırmalar*

Yazar(lar)	Yıl	Araştırma Adı	Araştırma Amacı	Örneklem	Ana Bulgular
Anagnostopoulos	2023	ChatGPT Impacts in Programming Education: Literature Overview	ChatGPT'nin programlama eğitimine etkilerini ele alan çalışmaları sistematik biçimde inceleyerek genel eğilimleri ortaya koymak.	40+ çalışma	ChatGPT'nin programlama öğretimine etkileri 5 başlıkta özetlenmiş; pedagojik potansiyeli güçlü şekilde vurgulanmıştır.
Atun	2024	Üretken Yapay Zeka Destekli Programlama Eğitiminde Sorgu Mühendisliğinin Değerlendirilmesi	Üretken yapay zekâ destekli programlama eğitiminde sorgu mühendisliği düzeyi ile öğrenme çıktıları arasındaki ilişkiyi incelemek.	Üniversite düzeyinde öğrenciler	ChatGPT destekli programlama öğretiminde sorgu kalitesi, bilişsel yük ve öz düzenleyici öğrenme ilişkili olarak ele alınmıştır.
Becker	2023	Uses of Generative AI in Secondary School CS	Ortaöğretim düzeyinde bilgisayar bilimi öğretiminde üretken yapay zekâ araçlarının kullanım örneklerini ortaya koymak.	Ortaokul öğrencileri	ÜYZ araçlarının ortaokul düzeyinde bilgisayar bilimleri öğretiminde nasıl kullanıldığına dair gözlem verileri sunulmuştur.
Çetin & Aktaş	2021	Yapay Zekâ ve Eğitimde Gelecek Senaryoları	Eğitimde yapay zekânın geleceğine yönelik olası senaryoları uzman görüşleri doğrultusunda değerlendirmek.	-	YZ'nin eğitimdeki geleceği senaryolarla tartışılmıştır; ÜYZ özelinde doğrudan katkı sınırlı ama bağlam zenginliği sağlamaktadır.
Dickey vd.	2023	Innovating Computer Programming Pedagogy: The AI-Lab Framework for Generative AI Adoption	Üretken yapay zekânın programlama öğretimine entegrasyonu için AI-Lab pedagojik çerçevesini geliştirmek ve uygulanabilirliğini değerlendirmek.	Lisans düzeyi programlama öğrencileri	AI-Lab çerçevesi ile dengeli ÜYZ kullanımı; temel beceriler korunurken öğrenci ilgisi artmıştır.

Tablo 1 (Devamı)

Gözet & Yılmaz	2023	Üretken Yapay Zekâ	Üretken yapay zekâ kavramını teknik ve kuramsal boyutlarıyla ele almak	-	ÜYZ kavramı teknik düzeyde açıklanmış, eğitimde doğrudan uygulamalara sınırlı yer verilmiştir.
İncemen & Öztürk	2024	Farklı Eğitim Alanlarında Yapay Zekâ: Uygulama Örnekleri	Farklı eğitim alanlarında yapay zekâ teknolojilerinin kullanım örneklerini tanıtmak ve bu örneklerin öğretimsel katkılarını tartışmak.	-	Chatbotlar ve akıllı öğretici sistemler gibi ÜYZ sınıfına giren uygulamalara yer verilmiştir; öğretimsel bağlamda değerlidir.
Keuning vd.	2024	Students' Perceptions and Use of Generative AI Tools for Programming Across Different Computing Courses	Farklı bilgisayar bilimi derslerinde öğrencilerin üretken yapay zekâ araçlarını nasıl kullandıklarını ve bu araçlara yönelik algılarını belirlemek.	264 öğrenci, 59 farklı ders	Öğrencilerin ÜYZ algıları pozitif yönde; kullanımları ders türlerine göre değişmektedir.
Kutlucan & Seferoğlu	2024	Eğitimde Yapay Zekâ Kullanımı: ChatGPT'nin KEFE ve PEST Analizi	ChatGPT'nin eğitimde kullanımını kuramsal, etik, finansal ve teknik boyutlarda değerlendirmek.	150 Akademik çalışma	ChatGPT'nin eğitimde kullanımına ilişkin kavramsal, etik, finansal ve teknik boyutlar detaylı şekilde değerlendirilmiştir
Lee vd.	2024	Generative AI's Impact on Programming Students: Frustration and Productivity	Üretken yapay zekâ araçlarının programlama öğrencileri üzerindeki etkilerini, özellikle üretkenlik ve hayal kırıklığı boyutlarında incelemek.	Üniversite öğrencileri	ÜYZ araçları üretkenliği artırsa da bazı öğrencilerde hayal kırıklığı yaratmaktadır.
Lodzowski vd.	2023	Generative AI and Its Educational Implications	Üretken yapay zekânın eğitim ortamlarına etkilerini pedagojik ve teknolojik yönleriyle değerlendirmek.	-	ÜYZ'nin eğitimdeki genel etkileri ele alınmış; programlama eğitimi özelinde sınırlı ama bağlam açısından anlamlıdır.

Tablo 1 (Devamı)

Ma vd.	2024	Enhancing Education with ChatGPT	Programming Python programlama dersi bağlamında öğrencilerin ChatGPT ile olan etkileşimlerini ve öğrenme deneyimlerini analiz etmek.	Üniversite öğrencileri (Python eğitimi)	Öğrenciler ChatGPT'yi öğrenme sürecinde faydalı bulmakta, öz düzenleyici öğrenmeye katkı sağladığı düşünülmektedir.
Mello vd.	2024	Education in the Age of Generative AI	Üretken yapay zekâ çağında eğitimin nasıl şekillendiğini ve öğretmen-öğrenci rollerinin nasıl dönüştüğünü incelemek.	-	ÜYZ'nin eğitimdeki etkileri pedagojik, etik ve teknolojik boyutlarda incelenmiş; öğretmenlerin rehberlik rolü vurgulanmıştır.
Ogunleye vd.	2024	A Systematic Review of Generative AI for Teaching and Learning Practice	Öğretim ve öğrenme süreçlerinde üretken yapay zekâ kullanımına yönelik yapılmış araştırmaları sistematik biçimde derlemek.	625 Makale	ÜYZ'nin öğretim uygulamalarındaki yaygın kullanım alanları, fırsatları ve sınırlılıkları detaylıca incelenmiştir.
Papadakis	2023	Generative Artificial Intelligence in Computer Programming Education	Üretken yapay zekâ teknolojilerinin bilgisayar programlama eğitiminde kullanımını incelemek.	-	ÜYZ'nin programlama eğitiminde nasıl konumlandırıldığını ve gelecekteki araştırma boşluklarını ortaya koymaktadır.
Prather vd.	2024	Beyond the Hype: A Comprehensive Review of Current Trends in Generative AI Research, Teaching Practices, and Tools	Üretken yapay zekâ ile ilgili güncel araştırma eğilimlerini, öğretim uygulamalarını ve kullanılan araçları kapsamlı biçimde incelemek.	59 farklı programlama dersi ve öğretmen	ÜYZ araçları öğretimde kişiselleştirme, hızlı geri bildirim ve öğrenci desteği sağlamakta etkilidir.

Tablo 1 (Devamı)

Tekin	2023	Eğitimde Yapay Zekâ: Türkiye Kaynaklı Araştırmaların Eğilimleri Üzerine Bir İçerik Analizi	Türkiye merkezli yapay zekâ ve eğitim ilişkili akademik yayınların eğilimlerini içerik analizi yöntemiyle belirlemek.	39 Çalışma	Türkiye'deki yapay zekâ odaklı eğitim araştırmalarında artış gözlemlenmiştir; ÜYZ sınırlı da olsa yer bulmuştur.
Tian vd.	2023	Is ChatGPT the Ultimate Programming Assistant?	ChatGPT'nin yazılım geliştirme süreçlerinde kod üretimi, hata ayıklama ve açıklama görevlerindeki performansını değerlendirmek.	Lisans düzeyinde öğrenciler	ChatGPT'nin kod üretimi, hata düzeltme ve açıklama görevlerinde yüksek performans gösterdiği bulunmuştur.
UNESCO	2024	Eğitim ve Araştırmada Üretken Yapay Zekâ Kılavuzu	Eğitim ve araştırma ortamlarında üretken yapay zekâ araçlarının güvenli, etik ve etkili kullanımını yönlendirmek amacıyla bir kılavuz sunmak.	-	ÜYZ'nin eğitimde kullanımı için küresel ilkeler ve etik çerçeve sunulmuştur; öğretmen ve öğrenciler için yönergeler yer almaktadır.
Waseem vd.	2024	ChatGPT as a Software Development Bot	Lisans düzeyinde yazılım geliştirme süreçlerinde ChatGPT'nin öğrenme ve üretkenliğe etkilerini incelemek.	Yazılım mühendisliği öğrencileri	Proje tabanlı öğrenmede ChatGPT'nin öğrenci başarısı ve etkileşimini olumlu etkilediği gözlemlenmiştir.
Zhang & Nouri	2022	The Effect of Generative AI-Based Tool Use on Programming Education	Üretken yapay zekâ tabanlı araçların programlama eğitimi üzerindeki etkilerini değerlendirmek.	Bilgisayar bilimi öğrencileri	ÜYZ araçları öğrencilerin programlama öğrenme sürecini hızlandırmakta ve başarıyı artırmaktadır.

İlgili arařtırmalar incelendiğinde, ÜYZ araçlarının özellikle yükseköğretim düzeyinde programlama öğretiminde ele alındığı, ancak ortaokul kademesine yönelik doğrudan deneysel uygulamaların sınırlı olduğu görölmektedir. Özellikle bilgi işlemsel düşünme becerileri ve programlamaya yönelik tutumlar gibi bilişsel ve duyuşsal değişkenlerin birlikte ele alındığı kapsamlı arařtırmalar oldukça azdır.

Bu tez, üretken yapay zekâ destekli dijital öğrenme ortamlarının ortaokul düzeyinde programlama öğretiminde nasıl etkili kullanılabileceğini inceleyen nadir çalışmalardan biridir. Öğrencilerin hem bilişsel kazanımlarını (bilgi işlemsel düşünme) hem de duyuşsal yönelimlerini (programlamaya yönelik tutum) birlikte ele alarak bu teknolojilerin öğretime entegrasyonundaki potansiyelini bütüncül bir bakışla değerlendirmektedir. Bu yönüyle tez, literatürdeki pedagojik boşluklardan birini doldurmakta ve gelecek çalışmalara güçlü bir zemin sunmaktadır.

ÜÇÜNCÜ BÖLÜM

Yöntem

Araştırma Yöntemi

Bu araştırmada, üretken yapay zekâ destekli öğrenme modülünün ortaokul programlama eğitimindeki etkilerini değerlendirmek amacıyla karma araştırma yöntemi kullanılmıştır. Karma araştırma yöntemi, nitel ve nicel veri toplama ve analiz süreçlerinin bir arada kullanılmasıyla, araştırma problemini derinlemesine inceleme olanağı sunar (Creswell, 2008). Araştırma, karma araştırma yöntemi kapsamında sıralı açıklayıcı desen kullanılarak gerçekleştirilmiştir. Bu desende öncelikle nicel veriler toplanmış ve analiz edilmiş, ardından bu sonuçları derinlemesine anlamlandırmak amacıyla nitel veriler toplanarak yorumlayıcı bir analiz yapılmıştır.

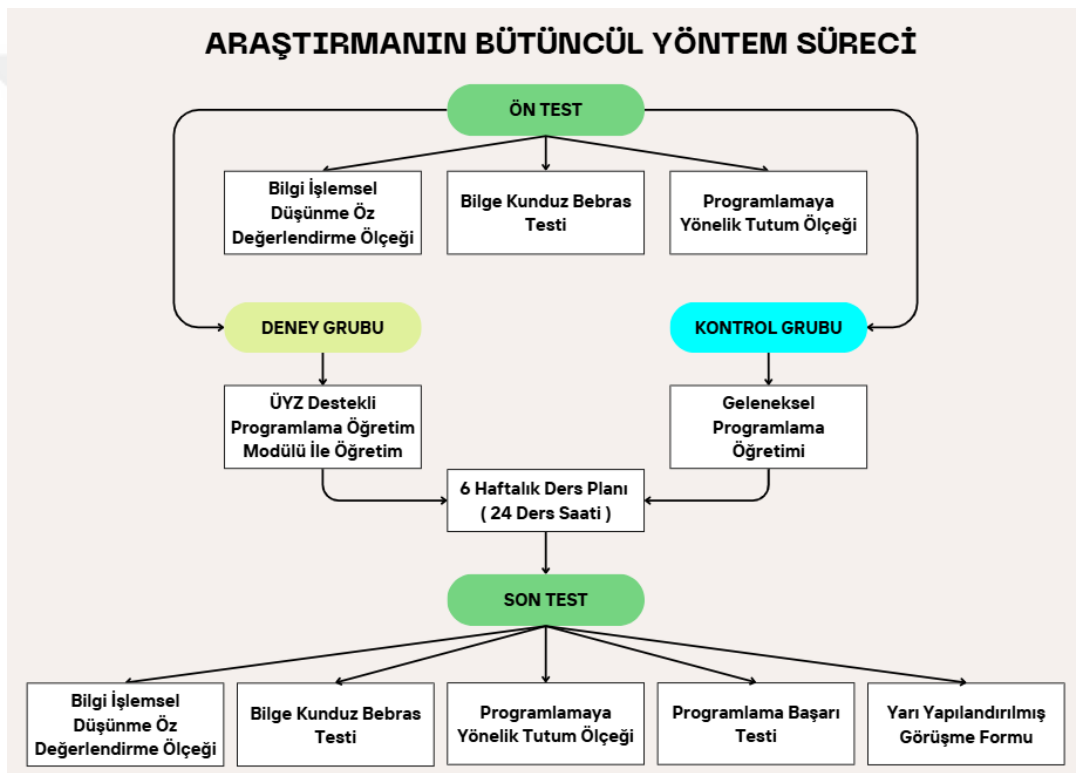
Araştırma Deseni

Araştırmada tercih edilen sıralı açıklayıcı desen, ilk olarak nicel verilerin toplanması ve analiz edilmesini, ardından bu veriler doğrultusunda nitel verilerin toplanarak sonuçların derinlemesine açıklanmasını içeren bir karma yöntem yaklaşımıdır. Bu desenin seçilmesinin temel nedeni, üretken yapay zekâ destekli öğrenme modülünün öğrencilerin programlama başarısı, bilgi işlemsel düşünme becerileri ve programlamaya yönelik tutumları üzerindeki etkilerini daha kapsamlı biçimde değerlendirebilmektir (Baki & Gökçek, 2012; Creswell & Plano-Clark, 2017).

Sıralı açıklayıcı desen, özellikle nicel bulguların pedagojik bağlamda açıklanması ve desteklenmesi için nitel verilerin yapılandırılmış bir biçimde kullanılmasını sağlar. Bu yaklaşımda, araştırmanın ilk aşamasında nicel veriler başarı testleri, ölçek puanları ve ölçme araçlarıyla toplanmış; bu aşamadan elde edilen sonuçlar, ikinci aşama olan nitel veri toplama sürecinin temelini oluşturmuştur. Nitel veriler ise öğrencilerle yapılan yarı yapılandırılmış görüşmeler yoluyla elde edilmiştir. Bu iki veri türünün zaman sıralı bir biçimde toplanması, hem araştırma bulgularının neden-sonuç ilişkisiyle açıklanmasına hem de elde edilen sonuçların bütüncül biçimde yorumlanmasına olanak tanımıştır.

Creswell ve Plano-Clark (2017), sıralı açıklayıcı deseni, araştırmacının öncelikle genel bir nicel sonuç elde etmek ve ardından bu sonuçları açıklamak amacıyla nitel veriler topladığı bir strateji olarak tanımlamaktadır. Bu yapı, özellikle eğitim araştırmalarında, ölçülen etkilerin arka planını anlamaya yönelik güçlü bir yöntem sunmaktadır. Bu çalışmada da, öğrencilerin bilgi işlemsel düşünme becerilerinde ve programlama başarılarındaki değişimler önce nicel verilerle ortaya konmuş, ardından bu değişimlerin altında yatan nedenler nitel analizlerle açıklanmıştır. Böylece, araştırmada kullanılan öğrenme modülünün etkileri hem istatistiksel hem de pedagojik düzlemde çok boyutlu olarak değerlendirilmiştir. Çalışmanın karma yöntem kapsamındaki bütüncül araştırma süreci Şekil 2’de sunulmuştur.

Şekil 2. *Araştırmanın Bütüncül Yöntem Süreci*



Nicel Bölüm: Yarı Deneysel Desen

Araştırmanın nicel bölümünde yarı deneysel desen kullanılmıştır. Yarı deneysel desen, deney ve kontrol grupları arasında gözlenen farklılıkların karşılaştırılmasını sağlar (Büyüköztürk, 2007). Çalışmada, ortaokul 7. sınıf öğrencilerinden rastgele seçilen iki şube, deney ve kontrol grubu olarak belirlenmiştir. Deney grubundaki öğrenciler, üretken yapay zekâ destekli öğrenme modülü kullanarak Python programlama dilini öğrenirken, kontrol grubundaki öğrenciler geleneksel yöntemlerle öğretilmiştir.

Öğrencilerin programlama başarısı, bilgi işlemsel düşünme becerileri ve programlama tutumları nicel veri toplama araçları kullanılarak ölçülmüştür.

Nitel Bölüm: Durum Çalışması

Araştırmanın nitel bölümü, belirli bir durumun derinlemesine incelenmesini sağlayan durum çalışması modeli ile yürütülmüştür (Yıldırım & Şimşek, 2006). Bu bölümde, üretken yapay zekâ destekli öğrenme modülünün uygulama süreci ve özellikleri öğrenci görüşleri aracılığıyla değerlendirilmiştir.

Nicel analizler sonucunda elde edilen bulguların daha derinlemesine anlamlandırılması için öğrencilerle yarı yapılandırılmış görüşmeler yapılmıştır. Bu yöntemle, öğrencilerin üzerinde üzerinde çalışılan modül hakkındaki deneyimlerini, güzellikleri ve zorlukları kendi sözleriyle ortaya koymaları hedeflenmiştir. Nitel veriler, betimsel ve içerik analizi yöntemleriyle analiz edilmiş, nicel bulgularla ilişkilendirilerek yorumlanmıştır.

Katılımcılar

Bu araştırmanın katılımcıları Bitlis ili Merkez ilçesinde bulunan ortaokul 7. sınıf öğrencilerinden oluşmaktadır. Araştırmada örneklem seçiminde araştırmanın amaçlarına uygun bir yaklaşım benimsenmiştir. Aşağıda, evren ve örneklem detayları sunulmuştur.

Evren

Araştırmanın evreni, Bitlis Merkez ilçesindeki tüm ortaokullarda eğitim gören 7. sınıf öğrencilerini kapsamaktadır. Bu evren, Türkiye'nin doğu bölgelerindeki eğitimsel dinamikleri, demografik çeşitliliği ve sosyo-kültürel farklılıkları yansıtması açısından önemlidir. Araştırmanın evreni, üretken yapay zekâ destekli eğitim araçlarının farklı sosyo-ekonomik ve kültürel özelliklere sahip öğrenci grupları üzerindeki etkilerini incelemek için uygun bir çerçeve sunmaktadır.

Örneklem

Bu araştırma için seçilen örneklem, Bitlis Merkez ilçesindeki Muştakbaba Ortaokulu'nda eğitim gören 7. sınıf öğrencilerinden oluşmaktadır. Örneklem, 34 öğrenciden meydana gelmekte olup bu öğrenciler A ve B şubelerinde eğitim gördükleri için örneklem seçimi A şubesi deney, B şubesi kontrol grubu olarak belirlenmiştir. Örneklem seçiminde, araştırmanın genelleştirilebilirliğini ve güvenilirliğini artırmak hedeflenmiştir. Örneklemdeki

öğrencilerin 16'sı erkek, 18'i kızdır ve A ile B şubelerinde rastgele atama yöntemi ile dağılmıştır. Her şubede 17 öğrenci bulunmaktadır.

Deney grubu, A şubesinde yer alan öğrencilerden oluşmakta ve bu grup, üretken yapay zekâ destekli öğrenme modülü kullanarak Python programlama dilini öğrenmişlerdir. Kontrol grubu ise B şubesindeki öğrencilerden oluşmakta ve bu grup, geleneksel programlama eğitimi yöntemleri ile ders almışlardır. Gruplar, başlangıç düzeyinde benzer demografik ve akademik özelliklere sahip olacak şekilde belirlenmiş olup, bu durum araştırma bulgularının geçerliliğini artırmayı hedeflemektedir.

Araştırma kapsamında seçilen örneklem, 7. sınıf düzeyindeki 34 öğrenciden oluşmaktadır ve öğrenciler haftada 40'ar dakikalık 4 ders saati boyunca programlama eğitimi almaktadır. Bu öğrenciler, geçtiğimiz yıl programlama temelleri ve algoritma derslerini başarıyla tamamlamış olup, bu dönem başında Python programlama dilini öğrenmeye başlamışlardır. Öğrencilerin programlama bilgi düzeyi başlangıç ve temel seviye arasında değişmekte olup, bu durum çalışmanın modül tasarımı ve uygulanabilirliği açısından dikkate alınmıştır. Bu örneklem, küçük bir grup üzerinde yapılan çalışmanın detaylı analizine olanak tanırken, aynı zamanda Türkiye'nin doğu bölgesindeki öğrenciler için etkili bir eğitim modeli geliştirme potansiyelini değerlendirmek için uygun bir temel sunmaktadır.

Araştırmanın nitel bölümünde, deney grubundan 10 öğrenci ile yarı yapılandırılmış görüşmeler yapılmıştır. Bu öğrenciler, modülün kullanımına yönelik farklı deneyimlerin kapsamlı bir şekilde değerlendirilmesi amacıyla rastgele seçilmiştir. Görüşmeler sırasında öğrencilerin programlama becerileri, bilgi işlemsel düşünme süreçleri ve modül kullanımına yönelik geri bildirimleri toplanmıştır. Ayrıca, öğrencilerin bireysel farklılıklarının ve öğrenme süreçlerindeki etkilerinin anlaşılabilmesi için çeşitli sosyo-ekonomik özellikler de göz önünde bulundurulmuştur.

Çalışmada etik kurallara büyük bir özen gösterilmiştir. Araştırmaya katılan tüm öğrencilerin ebeveynlerinden ve kendilerinden yazılı onam formları alınmıştır. Araştırmanın amacı, kapsamı ve süreçleri hakkında katılımcılar ve ebeveynler bilgilendirilmiş, katılımın tamamen gönüllülük esasına dayandığı belirtilmiştir. Ayrıca, öğrencilerin kişisel bilgileri ve verileri, gizlilik ilkelerine uygun olarak korunmuş ve yalnızca araştırma amaçları doğrultusunda kullanılmıştır. Araştırma boyunca öğrencilerin fiziksel ve psikolojik rahatlıklarının sağlanması önceliklendirilmiş ve çalışmanın etik kurallara uygunluğu ilgili etik kurul tarafından onaylanmıştır.

Veri Toplama Araçları

Bu tez çalışmasında, üretken yapay zeka destekli öğrenme modülünün etkilerini değerlendirmek amacıyla çeşitli veri toplama araçları kullanılmıştır. Bu araçlar; bilgi işlemsel düşünme öz değerlendirme ölçeği, bilge kunduz bebras testi, programlamaya yönelik tutum ölçeği, programlama başarı testi ve yarı yapılandırılmış görüşme formundan oluşmaktadır.

Bilgi İşlemsel Düşünme Becerisi

Bilge Kunduz ve Bebras Testi

Bilge Kunduz, uluslararası düzeyde düzenlenen ve öğrencilerin bilgi işlemsel düşünme becerilerini geliştirmeyi hedefleyen bir etkinliktir. Bu etkinlik, öğrencilere enformatik kavramlarını öğretmek ve bilgisayar bilimine olan ilgilerini artırmak amacıyla farklı ülkelerde uygulanmaktadır. 2018 yılında, Bilge Kunduz etkinliği 45 ülkede gerçekleştirilmiş ve 2.614.000 öğrencinin katılımı ile büyük bir etki yaratmıştır (Gülbahar vd., 2020). Etkinlik, her yıl daha fazla öğrenciye ulaşarak, bilgi işlemsel düşünme becerilerinin gelişimine katkıda bulunmayı sürdürmektedir.

Bu araştırmada kullanılan Bebras Testi, Bilge Kunduz etkinliğinin bir parçası olarak tasarlanmış ve öğrencilerin bilgi işlemsel düşünme becerilerini ölçmek amacıyla kullanılan çevrimiçi bir değerlendirme aracı olarak uygulanmıştır. Bebras Testi, çeşitli problem çözme görevleri içermekte olup, öğrencilerin mantıksal düşünme, algoritma geliştirme ve veri analiz etme becerilerini değerlendirmeye yönelik bir yapıya sahiptir (Dagienė & Futschek, 2008). Ayrıca, Bebras görevleri, öğrencilerin bilgi teknolojilerini kavrayarak problemleri çözme yeteneklerini geliştirmeyi amaçlayan etkili bir yöntem sunmaktadır (Dagienė & Sentance, 2016).

2024 yılındaki Bebras Testi final sınavında, toplamda 15 soru sorulmuş ve bu sorular üç farklı puanlama kategorisine göre değerlendirilmiştir. İlk 5 soru 6 puan, sonraki 5 soru 9 puan ve son 5 soru ise 12 puan üzerinden değerlendirilmiştir. Yanlış yanıtlar için ise sırasıyla -2, -3 ve -4 puan düşülerek sonuçlar hesaplanmıştır. Tüm katılımcılar başlangıçta 45 puanla değerlendirmeye başlamış ve en fazla 180 puan alabilecek şekilde bir değerlendirme yapısı oluşturulmuştur. Bu puanlama sistemi, öğrencilerin problem çözme süreçlerindeki başarılarını detaylı bir şekilde değerlendirmeye olanak tanımaktadır.

Bebras Testi'nin bu araştırmada kullanılması, öğrencilerin bilgi işlemsel düşünme becerilerindeki değişimlerin nicel olarak analiz edilmesine olanak sağlamıştır. Ayrıca, testin

uygulanabilirliği ve değerlendirme süreçlerinin güvenilirliği, daha önceki uygulamalardan elde edilen verilerle desteklenmiştir. Bu bağlamda, Bebras Testi, üretken yapay zekâ destekli öğrenme modülünün bilgi işlemsel düşünme becerilerine etkisini değerlendirmek için uygun bir araç olarak kullanılmıştır (Ek 2.). Uluslararası alanda geçerliliği kanıtlanmış olan Bebras Testi, çeşitli yaş gruplarına uygun olarak yapılandırılmıştır. Testin klasik anlamda bir Cronbach Alpha katsayısı olmamakla birlikte, uluslararası uygulamalarda madde güçlüğü ve ayırt edicilik analizlerine dayalı olarak güvenilirliği sağlanmaktadır (Dagienė & Futschek, 2008).

Bilgi İşlemsel Düşünme Öz Değerlendirme Ölçeği

Bilgi İşlemsel Düşünme Öz-Değerlendirme Ölçeği, 11-13 yaş arası öğrencilerin bilgi işlemsel düşünme becerilerini değerlendirmek amacıyla geliştirilmiş bir ölçme aracıdır (Ceylan & Vural, 2023). Ölçeğin geliştirilme süreci, bilgi işlemsel düşünme kavramının kapsamlı bir şekilde tanımlanması ile başlamış ve 17 öğretmenle yapılan görüşmelerle desteklenmiştir. İlk madde havuzu 40 maddeden oluşmuş, uzman değerlendirmesi ve pilot uygulama süreçleri sonucunda bu sayı 36'ya indirilmiştir. Açıklayıcı ve doğrulayıcı faktör analizleri sonucunda, ölçek, algoritmik düşünme, paralelleştirme, ayırıştırma, otomasyon ve soyutlama olmak üzere beş faktör altında toplanmış ve 20 maddelik nihai form oluşturulmuştur (Ek 3). Ölçek, algoritmik düşünme, paralelleştirme, ayırıştırma, soyutlama ve otomasyon olmak üzere beş alt boyuttan oluşmaktadır. Ölçeğin özgün geliştirme çalışmasında Cronbach Alpha güvenirlik katsayısı genel ölçekte .88 olarak raporlanmıştır (Yıldız & Yılmaz, 2018). Bu çalışmada elde edilen veriler üzerinden hesaplanan Cronbach Alpha değeri ise .86'dır.

Algoritmik Düşünme, bir problemi çözmek için adım adım bir çözüm yolu belirleme becerisini ifade eder. Bu faktör, öğrencilerin problem çözme süreçlerinde sistematik bir yaklaşım sergileyip sergilemediklerini değerlendirir.

- Örnek Madde: "Bir problemi çözmek için önce yapmam gerekenleri sırayla planlarım."

Paralelleştirme, birden fazla sürecin eş zamanlı olarak yürütülmesini ifade eder. Öğrencilerin aynı anda birden fazla görev üzerinde çalışabilme yeteneklerini ölçer.

- Örnek Madde: "Birden fazla işlemi aynı anda yapabilmek için görevleri bölerek düzenlerim."

Ayırıştırma, bir problemi daha küçük ve yönetilebilir parçalara ayırma becerisini ifade eder. Bu faktör, öğrencilerin karmaşık problemleri analiz etme yeteneklerini değerlendirir.

- Örnek Madde: "Bir problemi çözmeden önce onu daha küçük parçalara bölerim."

Otomasyon, tekrarlayan görevlerin otomatik hale getirilmesini ifade eder. Öğrencilerin süreçleri optimize etme ve tekrarlı işlemleri düzenleme becerilerini ölçer.

- Örnek Madde: "Aynı işlemi birden fazla kez yapmamak için bir yol geliştiririm."

Soyutlama, bir problemin temel bileşenlerini ayıklayıp önemsiz detayları görmezden gelme becerisidir. Bu faktör, öğrencilerin genelleştirme ve model oluşturma kapasitelerini değerlendirir.

- Örnek Madde: "Bir problemin çözümüne ulaşırken, yalnızca önemli bilgilere odaklanırım."

Ölçek, 5'li Likert tipi bir değerlendirme ile uygulanmakta olup, her maddeye verilen puanlar 1'den 5'e kadar değişmektedir (1: Hiç katılmıyorum, 5: Tamamen katılıyorum). Toplam puan, öğrencinin bilgi işlemsel düşünme becerilerini genel olarak değerlendirmektedir. Faktörlere verilen ayrı puanlar ise becerilerin hangi alanlarda daha güçlü ya da zayıf olduğunu belirlemek için kullanılmaktadır.

Bu ölçek, öğrencilerin bilgi işlemsel düşünme becerilerini öz-değerlendirme yoluyla analiz etme imkânı sunmaktadır. Öğrencilerin kendi algılarına dayalı olarak bilgi işlemsel düşünme becerilerini değerlendirmeleri, becerilerindeki gelişimin izlenmesine ve eksikliklerin belirlenmesine olanak sağlamaktadır. Ölçeğin, eğitim araştırmalarında yaygın olarak kullanılan ve etkinliği kanıtlanmış bir araç olması, bu araştırmada kullanılmasını destekleyen önemli bir unsurdur.

Bu araştırma kapsamında Bilgi İşlemsel Düşünme Öz-Değerlendirme Ölçeği, üretken yapay zekâ destekli öğrenme modülünün öğrencilerin bilgi işlemsel düşünme becerileri üzerindeki etkilerini değerlendirmek amacıyla uygulanmıştır. Ölçek, araştırmanın başında ve sonunda uygulanarak öğrencilerin becerilerindeki değişim analiz edilmiş ve modülün etkinliği değerlendirilmiştir.

Programlamaya Yönelik Tutum Ölçeği

Altay ve Kışla (2018) tarafından geliştirilen Programlamaya Yönelik Tutum Ölçeği, öğrencilerin programlamaya ilişkin tutumlarını ölçmek amacıyla kullanılan bir değerlendirme aracıdır. Ölçek, 5'li Likert tipi bir yapıya sahip olup, 14 maddeden ve üç faktörden oluşmaktadır. Ölçeğin faktörleri; programlamaya yönelik istek, programlamanın yararına yönelik inanç ve programlamaya yönelik ilgi olarak belirlenmiştir. Tutum ölçeği üç alt boyuttan (İstek, İnanç, İlgi) oluşmaktadır. Ölçeğin geliştirme çalışmasında genel güvenilirlik katsayısı .89

olarak bildirilmiştir (Korkmaz & Altun, 2014). Bu araştırmada uygulama sonrası elde edilen Cronbach Alpha değeri .83'tür.

Programlamaya Yönelik İstek, öğrencilerin programlama öğrenmeye yönelik motivasyonlarını ve bu süreçten duydukları memnuniyeti ölçer. Öğrencilerin programlama ile ilgili heves ve kararlılık düzeylerini değerlendirmektedir.

- Örnek Madde: "Programlama öğrenmekten büyük bir keyif alıyorum."

Programlamanın Yararına Yönelik İnanç, öğrencilerin programlama becerilerinin hayatlarına ve kariyerlerine olan katkılarına ilişkin inançlarını değerlendirir. Programlamanın problem çözme ve düşünme becerilerini geliştirdiğine dair algılar bu faktör altında incelenir.

- Örnek Madde: "Programlama becerilerinin gelecekteki meslek hayatımda çok önemli olduğunu düşünüyorum."

Programlamaya Yönelik İlgi, öğrencilerin programlama konularına olan ilgilerini ve bu konularla meşgul olma isteklerini ölçer. Programlama konularına duyulan doğal merak ve ilgi bu başlık altında değerlendirilmektedir.

- Örnek Madde: "Programlama ile ilgili makaleleri ve haberleri okumaktan hoşlanırım."

Ölçeğin geçerlilik ve güvenilirlik çalışmaları kapsamlı bir şekilde gerçekleştirilmiştir. Yapılan analizlerde, ölçeğin Cronbach Alfa güvenilirlik katsayısı .89 olarak hesaplanmış ve bu sonuç, ölçeğin iç tutarlılık açısından yüksek düzeyde güvenilir olduğunu göstermiştir. Ayrıca, toplam varyans değeri %65.71 olarak bulunmuş ve ölçeğin, programlamaya yönelik tutumları değerlendirmede yeterli bir geçerliliğe sahip olduğu doğrulanmıştır. Uzman değerlendirmeleri ve pilot uygulamalar sonucunda, ölçeğin maddeleri öğrencilerin programlamaya yönelik tutumlarını ölçmek için uygun ve etkili bir yapı sunacak şekilde düzenlenmiştir.

Bu araştırmada, Programlamaya Yönelik Tutum Ölçeği, üretken yapay zekâ destekli öğrenme modülünün öğrencilerin programlamaya ilişkin tutumlarına etkisini değerlendirmek amacıyla uygulanmıştır. Ölçek, uygulama öncesinde ve sonrasında kullanılarak öğrencilerin tutumlarındaki değişim analiz edilmiştir. Elde edilen veriler, öğrencilerin programlamaya yönelik motivasyonları, inançları ve ilgilerindeki değişimleri kapsamlı bir şekilde ortaya koymuştur.

Programlamaya Yönelik Tutum Ölçeği, öğrencilerin programlama sürecine karşı duygusal, bilişsel ve davranışsal tepkilerini anlamak için güvenilir bir veri kaynağı sunmuş ve araştırma sonuçlarının doğruluğunu desteklemiştir. Ölçeğin uygulanması sırasında toplanan

veriler, üretken yapay zekâ destekli öğrenme modülünün öğrencilerin programlama tutumlarına olan etkisini değerlendirmek için kritik bir temel oluşturmuştur. (Ek 4.)

Programlama Başarı Testi (Rubrik)

Programlama Başarı Testi, öğrencilerin programlama becerilerini, bilgi düzeylerini ve problem çözme yetkinliklerini değerlendirmek amacıyla hazırlanmış bir ölçme aracıdır. Test, öğrencilerin Python programlama diline ilişkin temel kazanımlarını ve algoritmik düşünme yetkinliklerini ölçmek için yapılandırılmıştır. Testin içeriği, algoritma geliştirme, kodlama yetkinliği, hata ayıklama ve problem çözme boyutlarını kapsamaktadır. (Ek 1.)

Test, problem temelli öğrenme yaklaşımına dayanmakta ve öğrencilerden gerçek hayatta karşılaşılabilecek senaryoları temel alarak kodlama projeleri geliştirmeleri beklenmektedir. Sorular, öğrencilerin programlama kavramlarını anlama ve uygulama düzeylerini değerlendirecek şekilde üç ana boyutta ele alınmıştır:

- Algoritmik Düşünme: Öğrencilerin verilen bir problem için uygun algoritmayı tasarlama becerisini ölçen sorular.
- Kodlama Yetkinliği: Python programlama dilindeki temel kavramların (değişkenler, döngüler, koşullu ifadeler) doğru bir şekilde kullanımı ve uygulanmasını içeren sorular.
- Problem Çözme ve Hata Ayıklama: Öğrencilerin mevcut bir kodu analiz etme, hataları bulma ve düzeltme becerilerini değerlendiren görevler.

Her bir kazanım için öğrencilerin performansını değerlendirmek amacıyla ayrıntılı bir rubrik geliştirilmiştir. Rubrik, üç temel düzeyde (başlangıç, gelişmekte olan, ustalaşmış) öğrencilerin başarılarını nesnel ve sistematik bir şekilde değerlendirmek üzere tasarlanmıştır. Her düzey, öğrenci performansını açıklayan ölçütleri ve bu ölçütlere dayalı puanlama kriterlerini içermektedir. Örneğin, algoritmik düşünme boyutunda "başlangıç" düzeyi, basit algoritmaların anlaşılmasını içerirken, "ustalaşmış" düzeyde, karmaşık algoritmaların oluşturulması ve optimize edilmesi beklenmektedir.

Test, araştırma sürecinde yalnızca son test olarak uygulanmıştır. Test sonuçları, bu iki grubun programlama becerileri üzerindeki farklılıkların değerlendirilmesi ve üretken yapay zekâ destekli eğitimin etkisinin analiz edilmesi amacıyla kullanılmıştır. Tablo 2’de gösterildiği üzere programlama başarı testi MEB 7.sınıf Bilişim Teknolojileri ve Yazılım dersi müfredatı kazanımları ile ilişkilendirilerek hazırlanmıştır.

Tablo 2. *Programlama Başarı Testi Soru Kazanım Tablosu*

Soru No	Soru İçeriği	Kazanım
1	Not analiz algoritması: ortalama, geçme-kalma, maksimum-minimum bulma	Problem çözmede algoritmik düşünmeyi uygular.
2	Çift sayı bulma ve toplamını yazdırma (liste kullanım ve döngü)	Değişken, karar yapısı ve döngü ifadeleri kullanarak algoritma geliştirir.
3	Hatalı Python kodunu düzeltme ve nedenlerini açıklama	Yazdığı programlarda oluşabilecek hataları ayıklar ve çözümler üretir.
4	Alışveriş toplamı hesaplama ve 100 TL üstü müşteri sayısı bulma	Değişken, karar yapısı ve döngü ifadeleri kullanarak algoritma geliştirir.
5	Harfleri karıştırma ve tahmin oyunu algoritması (rastgelelik + kullanıcı etkileşimi)	Oyun, bulmaca gibi problem senaryolarına yönelik algoritmalar geliştirir.
6	Faktöriyel hesaplama, hatalı kodu düzeltme ve açıklama	Yazdığı programlarda oluşabilecek hataları ayıklar ve çözümler üretir.

Bu yapılandırılmış test, öğrencilerin programlama eğitimi sırasında kazandıkları becerilerin sistematik bir şekilde değerlendirilmesini sağlamakta ve araştırmanın bulgularının daha anlamlı hale getirilmesine katkıda bulunmaktadır.

Yarı Yapılandırılmış Görüşme Formu

Bu araştırmada, öğrenme modülünün özelliklerini belirlemek ve öğrencilerin uygulama sürecindeki deneyimlerini anlamak amacıyla yarı yapılandırılmış görüşme formu kullanılmıştır. Görüşme formu, ilgili literatür taraması ve uzman görüşleri doğrultusunda hazırlanmış olup, modülün pedagojik ve teknik özelliklerini değerlendirecek sorular içermektedir. Sorular öğrencilerin uygulama sonundaki geri bildirimlerini toplamak amacıyla tasarlanmıştır.

Öğrencilerle gerçekleştirilen yarı yapılandırılmış görüşmeler ise, modülün uygulama sürecindeki kullanım deneyimlerini ve öğrenci algılarını derinlemesine anlamayı hedeflemiştir. Bu görüşmelerde, öğrencilerin modüle ilişkin genel değerlendirmeleri, karşılaştıkları zorluklar, faydalı buldukları özellikler ve geliştirilmesi gerektiğini düşündükleri alanlara dair veriler toplanmıştır.

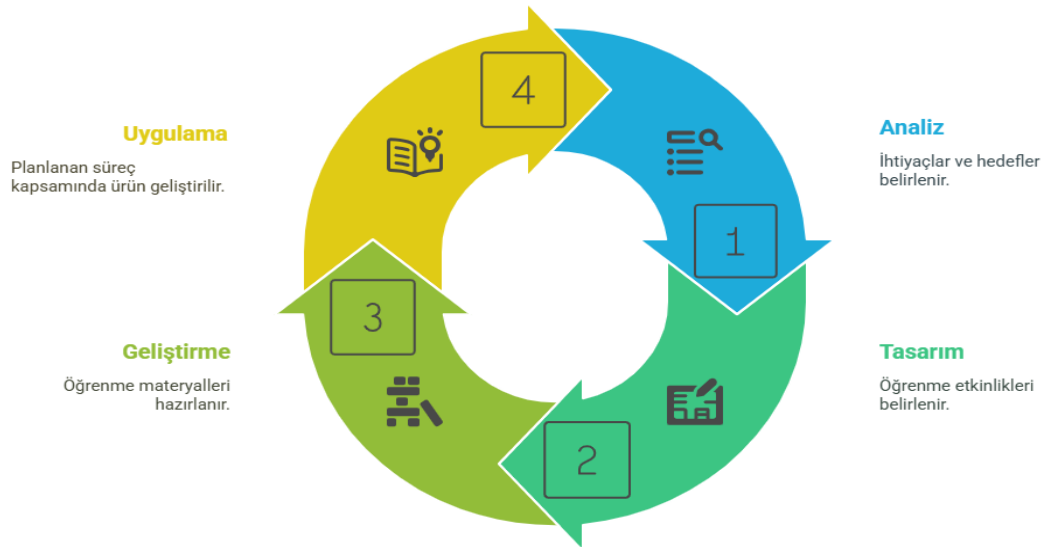
Görüşmeler, ses kayıt cihazları ile kaydedilmiş ve içerik analizi yöntemiyle tematik olarak analiz edilmiştir. İçerik analizi sürecinde, görüşme verilerinden elde edilen bulgular temalar ve kategoriler altında sınıflandırılmıştır.

Yarı yapılandırılmış görüşme, modülün tasarım ve uygulama süreçlerini değerlendirmede nitel bir veri kaynağı sağlamış ve araştırmanın kapsamlı sonuçlar sunmasına katkıda bulunmuştur. (Ek 5.)

Uygulama

Bu araştırmanın uygulama süreci, ADDIE modeli (Analysis, Design, Development, Implementation, Evaluation) temel alınarak sistematik bir şekilde planlanmıştır. ADDIE modeli, eğitim tasarım süreçlerinde etkili bir yaklaşım sağlayarak, her bir aşamanın amaçlarına uygun olarak düzenlenmesini ve değerlendirilmesini mümkün kılmaktadır. Araştırmanın uygulama süreci toplam 6 hafta sürmüş olup, her hafta belirli etkinlikler ve değerlendirme aşamalarını içermiştir. ADDIE modelinin aşamaları Şekil 3'te sunulmuştur.

Şekil 3. ADDIE Modeli Basamakları



Analiz Basamağı

Uygulamanın ilk haftasında, yapay zekâ destekli öğrenme modülünün tasarımına temel oluşturacak kapsamlı analiz çalışmaları gerçekleştirilmiştir. Bu analiz süreci, öğrenci ihtiyaçlarını belirlemek, mevcut öğrenme ortamlarını değerlendirmek ve modül tasarımı için kritik veriler elde etmek amacıyla yapılandırılmıştır.

Öğrencilerin mevcut programlama bilgileri, öğrenme düzeyleri ve demografik özelliklerini anlamak amacıyla bireysel bilgi formları kullanılmıştır (Ek 6.). Formlarda

öğrencilerin yaş, cinsiyet, önceki eğitim deneyimleri ve teknolojiye erişim düzeylerine ilişkin sorular yer almıştır. Bunun yanı sıra, öğrencilerin programlamaya ilişkin tutumlarını değerlendirmek için Programlamaya Yönelik Tutum Ölçeği uygulanmıştır. Bu ölçek, öğrencilerin programlamaya yönelik istek, ilgi ve programlamanın yararına yönelik inançlarını ölçmek amacıyla geliştirilmiştir ve çalışmanın başlangıç aşamasında deney ve kontrol gruplarının tutum düzeylerini karşılaştırmak için kullanılmıştır.

Uygulamanın gerçekleştirileceği laboratuvar ortamı detaylı bir şekilde analiz edilmiştir. Laboratuvarda mevcut bilgisayarların donanım ve yazılım özellikleri, internet bağlantı hızları ve diğer teknik altyapı unsurları değerlendirilmiştir. İnceleme sonuçları, laboratuvar şartlarının yapay zekâ destekli modülün uygulanması için yeterli olduğunu göstermiştir. Her öğrencinin bireysel olarak erişim sağlayabileceği bir bilgisayar bulunması, uygulama sürecinin verimli şekilde ilerlemesi için uygun bir ortam sunmuştur. Uygulamanın gerçekleştirileceği laboratuvar ortamında 17 adet Pardus işletim sistemine sahip 256 GB harddisk, 8 GB RAM kapasitesine sahip internet bağlantısı mevcut bilgisayarlar bulunmaktadır.

Bu aşamada elde edilen analiz sonuçlarında, öğrencilerin programlamaya yönelik tutumlarında başlangıç seviyesinde homojen bir dağılım olduğu belirlenmiştir. Öğrencilerin öğrenme materyallerinin çeşitlendirilmesi gerektiği görülmüştür. Laboratuvar ortamının uygulama süreci için yeterli olduğu tespit edilmiştir.

Tasarım Basamağı

Geliştirilen öğrenme modülü, ortaokul öğrencilerine programlama eğitimini daha etkin ve interaktif bir şekilde sunmayı amaçlamaktadır. Bu modül, öğrencilere Python programlama dilini öğretmeyi hedefleyen, çeşitli etkinlikler ve interaktif öğrenme materyalleri içeren bir eğitim platformu olarak tasarlanmıştır. Modülün en yenilikçi unsurlarından biri olan üretken yapay zekâ entegrasyonu, ChatGPT gelişmiş dil modeli aracılığıyla sağlanmıştır. Bu entegrasyon sayesinde, üretken yapay zekâ destekli modül, öğrencilerin yazdıkları kodların doğruluğunu kontrol edebilmelerini, hatalarını anında tespit ederek düzeltebilmelerini ve kodlarını optimize etmek için öneriler alabilmelerini sağlayacak şekilde tasarlanmıştır. Ayrıca, öğrencilerin modül içerisindeki materyallere ve etkinliklere yönelik sorularını üretken yapay zekâyâ yönelterek detaylı ve anında cevaplar alabilmelerini mümkün kılan bir yapı oluşturulmuştur. Bu özellik, öğrencilerin öğrenme süreçlerini desteklemenin ötesine geçerek, onları öğrenme süreçlerinin aktif bir parçası haline getirecek şekilde planlanmıştır. Üretken yapay zekâ, problem çözme süreçlerini hızlandırmak ve öğrencilerin algoritma geliştirme

becerilerini desteklemek amacıyla tasarlanmış bir araç olarak modüle entegre edilmiştir. Modül, web tabanlı bir platform olarak tasarlanmış ve erişilebilirlik ilkesi göz önünde bulundurularak, herhangi bir cihazdan ve herhangi bir yerden erişim sağlanabilecek şekilde geliştirilmiştir. Modülün teknik altyapısında Python tabanlı Django framework tercih edilmiş olup, bu güçlü ve esnek yapı modülün hızlı ve güvenilir bir şekilde çalışmasını sağlamıştır. Veritabanı yönetimi için PostgreSQL kullanılmıştır. Bu yüksek performanslı ve ölçeklenebilir sistem, kullanıcı verilerinin güvenli bir şekilde saklanmasını ve yönetilmesini mümkün kılmıştır. Modülün kullanıcı arayüzü, HTML, CSS ve JavaScript kullanılarak geliştirilmiş; interaktif ve dinamik içeriklerin sunulabilmesi için React.js kütüphanesi entegre edilmiştir. Üretken yapay zekâ modeli, OpenAI'den alınan API anahtarı ile modüle entegre edilmiştir. Bu API entegrasyonu, yapay zekâ özelliklerinin hızlı ve sorunsuz bir şekilde kullanılmasını sağlamıştır.

Modül tasarımında, öğrencilerin kişisel verilerinin güvenliği ve gizliliği öncelikli olarak ele alınmıştır. Veri güvenliğini sağlamak için SSL sertifikaları kullanılmış ve kullanıcı verileri şifrelenerek saklanmıştır. Ayrıca, yapay zekâ ile etkileşim sırasında toplanan verilerin gizliliği korunmuş ve yalnızca eğitim amaçlı kullanılmıştır. Bu süreç, uluslararası veri gizliliği standartlarına uygun olarak gerçekleştirilmiştir.

Üretken yapay zeka destekli programlama öğrenme modülünün geliştirme sürecinde, ortaokul düzeyindeki öğrencilerin bilişsel gelişim düzeyleri göz önünde bulundurularak yapay zekaya verilecek girdilerin (promptların) önceden tasarlanması önemli bir aşama olarak planlanmıştır. Modülün öğrencilere sade, anlaşılır ve yönlendirici cevaplar verebilmesi için üretken yapay zekâ sistemine belirli şablonlar ve komutlar entegre edilmiştir.

Bu süreçte kullanılan promptlar, eğitimsel prompt mühendisliği yaklaşımı temel alınarak yapılandırılmıştır. Promptlar; açıklama, örnek verme, ipucu sağlama, hata buldurma ve motivasyon artırma gibi farklı işlevlere göre kategorilendirilmiştir. Aşağıda, modüle önceden tanımlanmış örnek prompt kategorileri yer almaktadır:

1. Açıklama İsteyen Promptlar

Öğrencilerin temel kavramları anlaması için aşağıdaki türden yönlendirmeler tanımlanmıştır:

- “Kod satırlarını 7. sınıf düzeyinde açıkla.”
- “Bu kod ne işe yarar? Basit bir dille anlat.”

- “Python'da değişken nedir? Bir örnekle anlat.”

2. Örnek Üreten Promptlar

Öğrencilerin yapı benzerliği yoluyla öğrenmesini desteklemek amacıyla kullanılmıştır:

- “Bir if-else yapısını günlük hayattan bir örnekle göster.”
- “Kullanıcının yaşını soran ve ona göre mesaj yazan bir Python kodu üret.”

3. Hata Ayıklama Destekli Promptlar

Öğrencilerin kendi yazdığı kodları anlaması ve hataları tanınması için tanımlanmıştır:

- “Aşağıdaki kodda ne hata olabilir? Basitçe açıkla.”
- “Kod doğru çalışmıyorsa en olası neden nedir?”

4. Geri Bildirim ve Teşvik Promptları

Üretken yapay zekanın öğrenciyi sadece bilgilendirmekle kalmayıp, motive etmesi hedeflenmiştir:

- “Doğru yoldasın! Şimdi bir sonraki adımı düşün.”
- “Harika! Ama bir de input fonksiyonu kullanarak yapmayı dene.”

5. Yanlış Anlamayı Önleyici Promptlar

Kodun amacının yanlış yorumlanmasını önlemek için tanımlanmıştır:

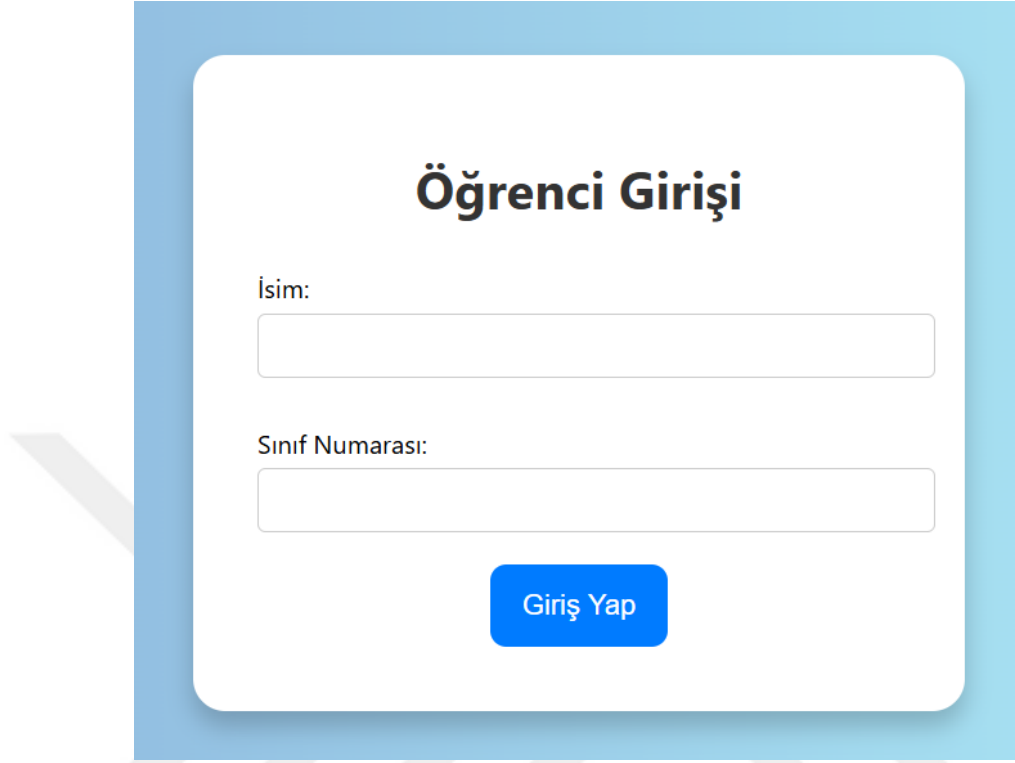
- “Bu kod sadece ekrana ‘Merhaba’ yazar, başka bir işlem yapmaz.”
- “Sonsuz döngü ne zaman oluşur, açıkla ve bir örnek ver.”

Bu promptlar, modüle gömülü olarak sabitlenmiş ve kullanıcıdan gelen girdilere göre otomatik olarak çalışacak şekilde yapılandırılmıştır. Böylece üretken yapay zeka, öğrencilerin düzeyine uygun, kavramsal bağlamdan kopmayan ve öğretici cevaplar üretmiştir. Bu yapı, öğrencilerin yalnızca çıktıya değil; sürece odaklanmalarını sağlamış, kavramsal öğrenmeyi desteklemiştir.

Şekil 4’de gösterilen Öğrenci Giriş Sayfası, kullanıcıların modüle erişim sağlamaları için tasarlanmış temel giriş ekranıdır. Bu sayfa, öğrencilerin kullanıcı adı ve şifre bilgilerini girerek güvenli bir şekilde sisteme giriş yapmalarını sağlar. Kullanıcı dostu bir arayüz sunan bu ekran, basit ve sade tasarımıyla öğrencilerin kolayca erişim sağlamasına olanak tanır. Giriş

işlemi tamamlandıktan sonra, öğrencilerin kod yazabilecekleri ve kodlarını sorabilecekleri sayfalarına yönlendirme işlemi gerçekleştirilir.

Şekil 4. Öğrenci Giriş Sayfası

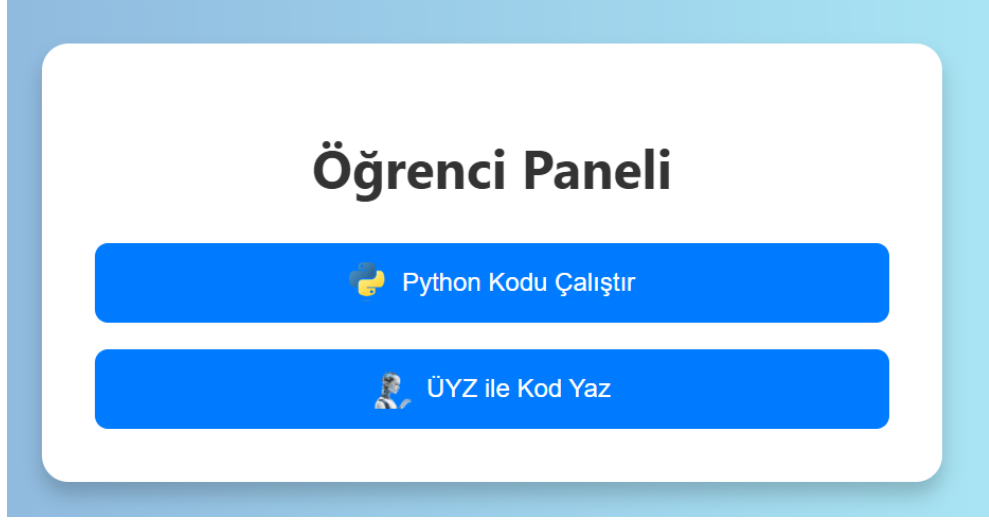
The image shows a login form titled "Öğrenci Girişi" (Student Login) centered on a light blue background. The form itself is a white rounded rectangle. It contains two input fields: the first is labeled "İsim:" (Name) and the second is labeled "Sınıf Numarası:" (Class Number). Below these fields is a blue button with the text "Giriş Yap" (Login). The entire form is set against a background with faint, stylized geometric shapes.

Şekil 5’de gösterilen Öğrenci Giriş Paneli, öğrencilerin öğrenme modülünde kullanacakları araçlara yönlendirilmesini sağlayan bir ana sayfa niteliğindedir. Panel, iki temel buton içerir: "Python Kodu Çalıştır" ve "ÜYZ ile Kod Yaz".

- "Python Kodu Çalıştır" butonu, öğrencilerin Python kodlarını çalıştırabilecekleri bir alana erişim sağlamak ve öğrendikleri kavramları uygulamalı olarak test etmelerine olanak tanımaktadır.
- "ÜYZ ile Kod Yaz" butonu, öğrencilerin üretken yapay zekâ desteğiyle kod yazmalarını ve geliştirmelerini destekleyen bir platforma yönlendirmektedir. Bu özellik, öğrencilerin yazılım süreçlerinde rehberlik almasını ve yaratıcı çözümler üretmesine teşvik etmektedir.

Panelin sade ve kullanıcı dostu tasarımı, öğrencilerin hızlı ve kolay bir şekilde ilgili bölümlere erişimini sağlamaktadır. Açılıştaki bir yönlendirme işlevi gören bu sayfa, öğrencilerin öğrenme süreçlerini etkin bir şekilde organize etmelerine yardımcı olmaktadır.

Şekil 5. Öğrenci Paneli Sayfası



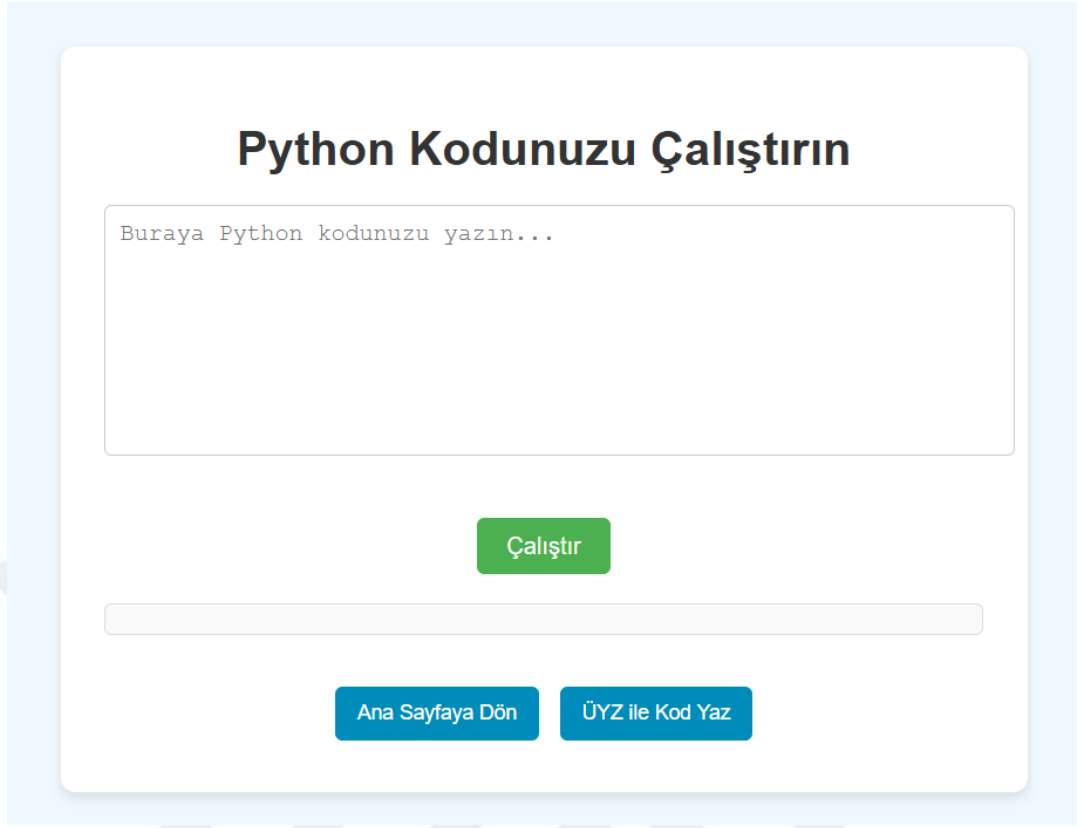
Şekil 6 ve Şekil 7’de gösterilen Python Kodu Çalıştırma Sayfası, öğrencilerin Python programlama dilinde kod yazmasını ve anında çalıştırmasını sağlayan bir ortam sunmaktadır. Bu sayfa, kullanıcıların yazılım geliştirme becerilerini pratik yaparak geliştirmelerine olanak tanımaktadır.

Sayfa, bir kod düzenleyici ve çıktı alanı olmak üzere iki ana bölümden oluşmaktadır:

- **Kod Düzenleyici Alanı:** Öğrencilerin Python kodlarını yazmaları için optimize edilmiş bir arayüz sağlamaktadır. Bu alan, kullanıcı dostu bir yapıya sahiptir.
- **Çıktı Alanı:** Yazılan kod çalıştırıldığında sonuçların anında görüntülenmesini sağlayan bir çıktı bölümü sunulmaktadır. Bu özellik, öğrencilerin hata ayıklama süreçlerini kolaylaştırmakta ve anında geri bildirim almalarını mümkün kılmaktadır.

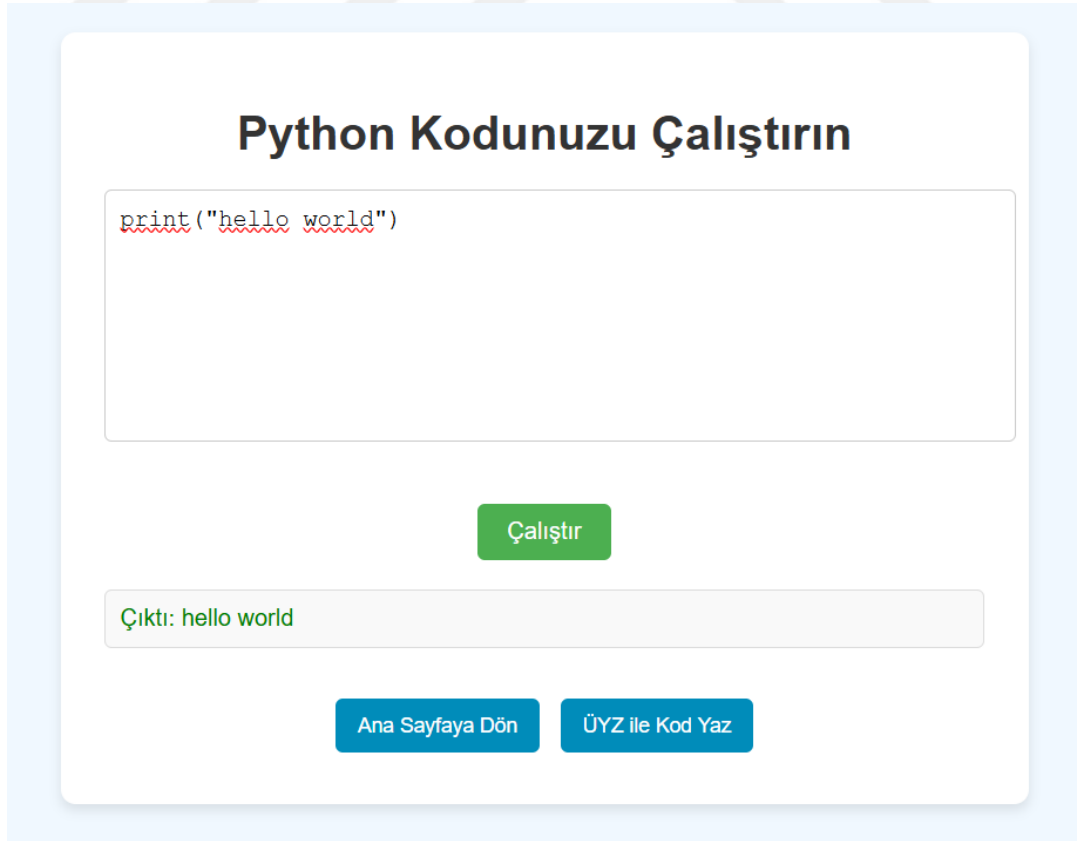
Python Kodu Çalıştırma Sayfası, öğrencilerin bağımsız olarak kodlama yapmasına olanak tanırken, pratik yaparak algoritmik düşünme becerilerini geliştirmelerine de katkı sağlamaktadır. Sayfa, öğrenme sürecini destekleyen sade ve işlevsel bir tasarıma sahiptir.

Şekil 6. *Python Kod Çalıştır Sayfası*



The screenshot shows a web interface for running Python code. At the top, there is a heading "Python Kodunuzu Çalıştırın". Below it is a large text area with the placeholder text "Buraya Python kodunuzu yazın...". In the center, there is a green button labeled "Çalıştır". Below the button is a horizontal progress bar. At the bottom, there are two blue buttons: "Ana Sayfaya Dön" and "ÜYZ ile Kod Yaz".

Şekil 7. *Python Kod Çalıştır Sayfası*



The screenshot shows the same web interface as Şekil 6, but with the code "print('hello world')" entered in the text area. The green "Çalıştır" button is still present. Below the button, the output "Çıktı: hello world" is displayed in a green font. The two blue buttons at the bottom, "Ana Sayfaya Dön" and "ÜYZ ile Kod Yaz", are also present.

Şekil 8’de gösterilen Üretken Yapay Zekâ (ÜYZ) ile Kod Yazma Arayüzü Sayfası, öğrencilerin üretken yapay zekâ araçlarını kullanarak kod yazma, hata ayıklama ve kod çalıştırma süreçlerini desteklemek amacıyla yönlendirme görevi görmektedir. Bu sayfa, üç temel buton aracılığıyla ÜYZ’nin farklı özelliklerine erişim sağlamaktadır:

- "ÜYZ’ye Kod Sor" Butonu: Öğrencilerin üretken yapay zekâya programlama ile ilgili sorular sormasını ve detaylı yanıtlar almasını mümkün kılmaktadır. Bu özellik, öğrencilerin öğrenme süreçlerine rehberlik etmektedir.
- "ÜYZ ile Kod Çalıştır" Butonu: Öğrencilerin yazdıkları kodları ÜYZ aracılığıyla çalıştırmasını ve geri bildirim almasını sağlamaktadır. Kod çalıştırma sürecinde, hatalar ve olası iyileştirme önerileri sunulmaktadır.
- "ÜYZ ile Hata Bul" Butonu: Öğrencilerin yazdıkları kodlardaki hataları ÜYZ desteği ile tespit etmelerine ve düzeltmelerine olanak tanımaktadır. Bu özellik, kodlama süreçlerinde karşılaşılan sorunların hızlı bir şekilde çözülmesini kolaylaştırmaktadır.

Bu arayüz, öğrencilere ÜYZ araçlarına kolayca erişim sağlayan sade ve kullanıcı dostu bir tasarım sunmaktadır. Yönlendirme işlevi gören bu sayfa, öğrencilerin öğrenme süreçlerini organize etmelerine ve ÜYZ desteği ile programlama becerilerini geliştirmelerine yardımcı olmaktadır.

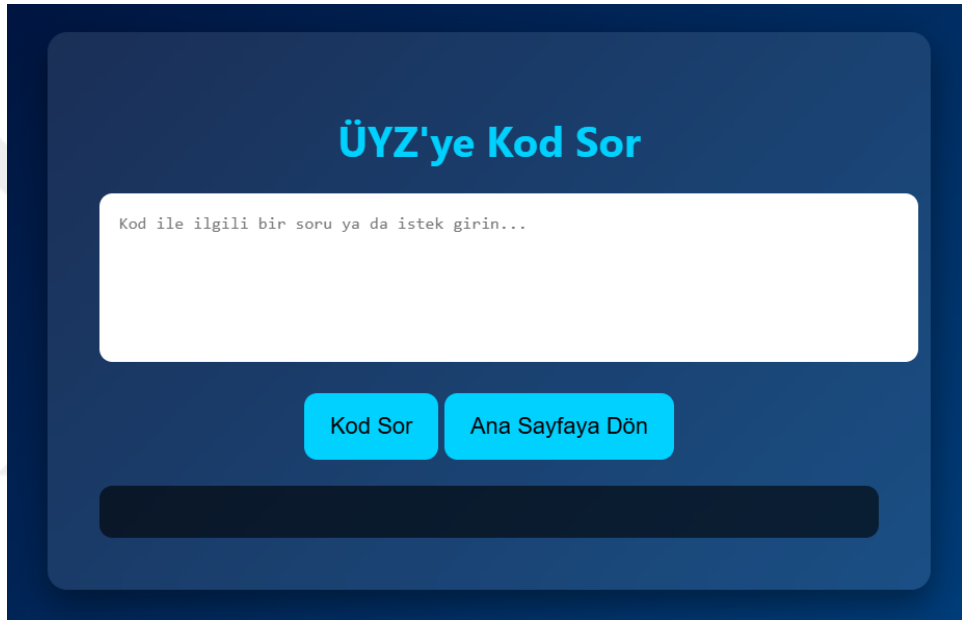
Şekil 8. ÜYZ İle Kod Yazma Arayüzü Sayfası



Şekil 9 ve Şekil 10’da gösterilen Üretken Yapay Zekâ (ÜYZ) ile Kod Sor Sayfası, öğrencilerin üretken yapay zekâya Python programlama ile ilgili sorularını yöneltebilecekleri bir arayüz sunmaktadır. Bu sayfa, öğrencilerin yazılı bir komut ya da prompt girerek ÜYZ’den programlama ile ilgili öneriler, açıklamalar veya çözümler almasını sağlamaktadır.

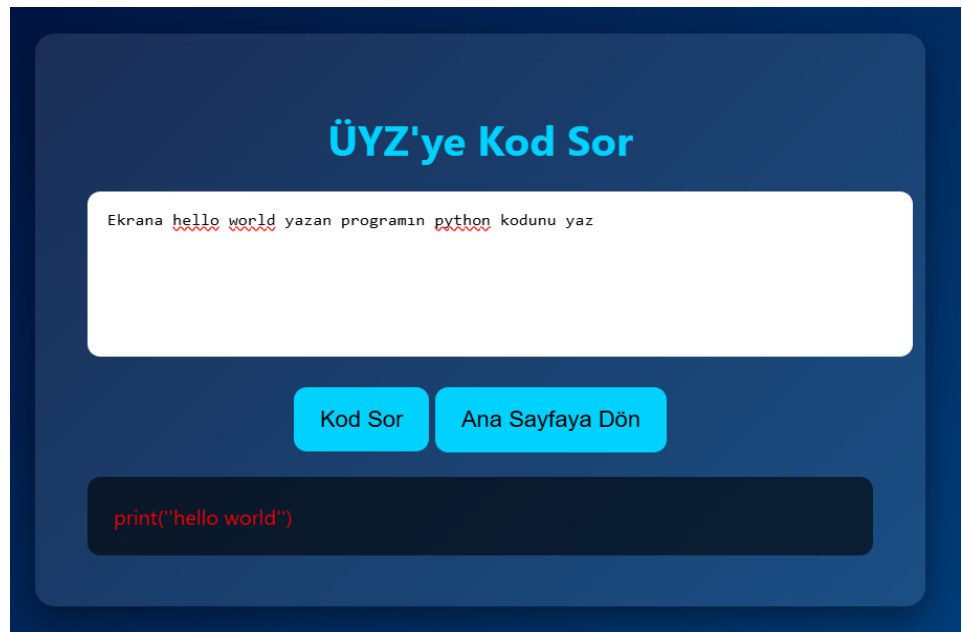
Sayfa, kullanıcı dostu bir giriş alanı ile öğrencilerin sorularını net bir şekilde iletmelerine olanak tanımaktadır. ÜYZ, öğrencilerin sorduğu sorulara yönelik detaylı yanıtlar üretmekte ve çözüm önerileri sunmaktadır. Bu özellik, öğrencilerin öğrenme sürecine rehberlik etmekte ve problem çözme becerilerini geliştirmektedir. ÜYZ'ye Kod Sor Sayfası, öğrencilerin programlama kavramlarını anlamalarına yardımcı olmakta, karmaşık problemleri çözmelerine destek sağlamakta ve öğrenme sürecini daha etkileşimli hale getirmektedir. Sayfa, öğrencilerin programlama becerilerini uygulamalı olarak geliştirmelerine katkıda bulunacak şekilde tasarlanmıştır.

Şekil 9. ÜYZ'ye Kod Sor Sayfası



The screenshot shows a dark blue background with a white text input field. Above the field is the title "ÜYZ'ye Kod Sor" in white. Below the field are two white buttons with black text: "Kod Sor" and "Ana Sayfaya Dön". Below the buttons is a dark blue horizontal bar.

Şekil 10. ÜYZ'ye Kod Sor Sayfası

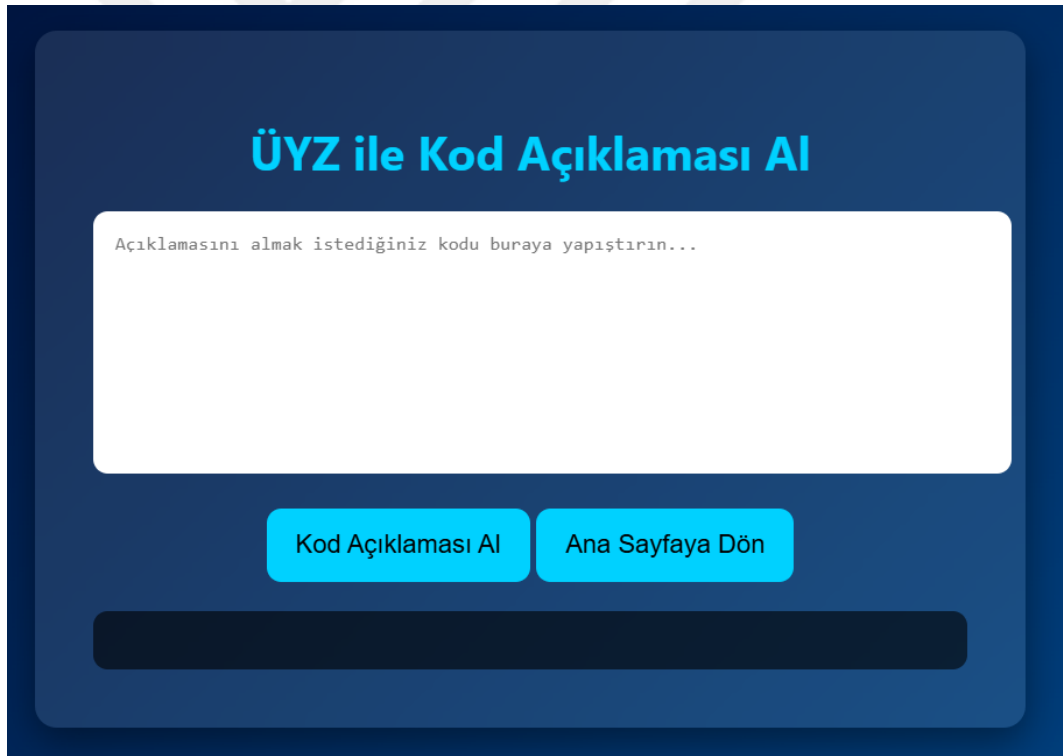


The screenshot shows a dark blue background with a white text input field. Above the field is the title "ÜYZ'ye Kod Sor" in white. Below the field are two white buttons with black text: "Kod Sor" and "Ana Sayfaya Dön". Below the buttons is a dark blue horizontal bar containing the code snippet: `print("hello world")`.

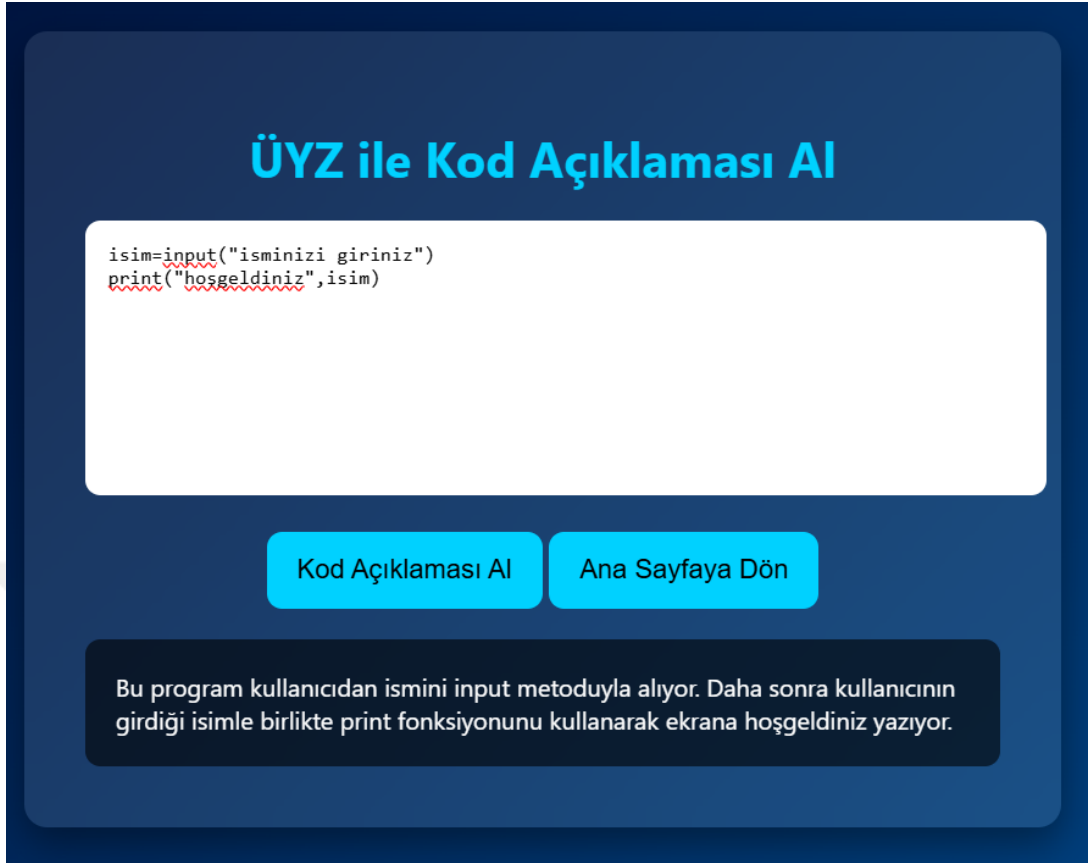
Şekil 11 ve Şekil 12’de gösterilen Üretken Yapay Zekâ (ÜYZ) ile Kod Açıklaması Al Sayfası, öğrencilerin yazdıkları Python kodlarının ne işe yaradığını öğrenmelerine olanak sağlayan bir arayüz sunmaktadır. Bu sayfada öğrenciler, belirli bir kod parçasını ÜYZ’ye girerek bu kodun işlevi, amacı ve çalışma mantığı hakkında detaylı açıklamalar alabilmektedir.

Sayfa, kod yazmak için optimize edilmiş bir giriş alanı sunmakta ve öğrencilerin yazılım geliştirme süreçlerinde rehberlik almasını kolaylaştırmaktadır. ÜYZ, öğrencilerin girdiği kodu analiz ederek, kodun nasıl çalıştığını ve hangi problemi çözmek için kullanılabileceğini açıklamaktadır. Ayrıca, kodun işleyişine dair ek bilgiler ve iyileştirme önerileri de sunulabilmektedir. Bu özellik, öğrencilerin kod yazma becerilerini geliştirirken aynı zamanda yazılım mantığını daha iyi kavramalarına yardımcı olmaktadır. ÜYZ ile Kod Açıklaması Al Sayfası, öğrencilerin programlama öğrenme süreçlerini destekleyen interaktif bir araç olarak işlev görmektedir.

Şekil 11. ÜYZ İle Kod Açıklaması Al Sayfası



Şekil 12. ÜYZ İle Kod Açıklaması Al Sayfası



Şekil 13 ve Şekil 14’de gösterilen Üretken Yapay Zekâ (ÜYZ) ile Hata Bul Sayfası, öğrencilerin hatalı Python kodlarını analiz ederek hatanın nedenlerini öğrenmelerini sağlayan bir arayüz sunmaktadır. Bu sayfada öğrenciler, hatalı kodlarını ÜYZ’ye girerek, bu kodlarda bulunan hataların sebeplerini ve olası çözüm önerilerini öğrenebilmektedir.

ÜYZ, öğrencilerin seviyesine uygun bir dil ve açıklama yöntemi kullanarak koddaki hataları analiz etmekte ve hatanın kaynağını açık bir şekilde ifade etmektedir. Ayrıca, hatanın düzeltilmesine yönelik rehberlik sunmakta ve öğrencilerin doğru çözüme ulaşmasını desteklemektedir. Bu özellik, öğrencilerin hata ayıklama becerilerini geliştirmelerine ve yazılım geliştirme süreçlerinde daha bilinçli kararlar almalarına yardımcı olmaktadır.

ÜYZ ile Hata Bul Sayfası, kullanıcı dostu tasarımıyla öğrencilerin kolayca kullanabileceği bir ortam sağlamaktadır. Bu sayfa, öğrencilere hatalarını öğrenme fırsatı sunarak, programlama bilgilerini pekiştirmelerine ve problem çözme yeteneklerini geliştirmelerine katkı sağlamaktadır.

Şekil 13. ÜYZ İle Hata Bul Sayfası

ÜYZ ile Hata Bul

Hataları bulmak istediğiniz kodu buraya yapıştırın...

Hataları Bul Ana Sayfaya Dön

Şekil 14. ÜYZ İle Hata Bul Sayfası

ÜYZ ile Hata Bul

```
def toplama(a, b):  
    return a + b  
sonuc = toplama(10)  
print("Toplama sonucu:", sonuc)
```

Hataları Bul Ana Sayfaya Dön

Bu hata, toplama işlemi için iki sayı verilmesi gerektiği halde sadece bir sayı verildiği için oluşuyor. Makine, eksik bilgiyle toplama yapamıyor ve bize "Bir sayı eksik!" diye hata veriyor.

Geliştirme Basamağı

Tasarlanan modülün geliştirilmesi aşamasında, hem yazılım araçlarının oluşturulması hem de destek materyallerinin hazırlanması hedeflenmiştir. Bu süreçte, öğrencilerin modülü etkili bir şekilde kullanmalarını sağlamak ve öğrenme deneyimlerini zenginleştirmek için çeşitli materyaller tasarlanmıştır.

Geliştirme aşamasında, modülün teknik altyapısına yönelik çalışmalar gerçekleştirilmiştir. Modülün kullanıcı dostu bir yapıya sahip olması ve öğrencilerin kolayca erişim sağlayabilmesi sağlanmıştır.

Öğrencilerin süreç hakkında bilgi sahibi olmalarını sağlamak ve motivasyonlarını artırmak amacıyla tanıtım materyalleri hazırlanmıştır (Ek 7.). Bu materyaller arasında tanıtım metinleri ve afişler yer almıştır. Tanıtım metinleri ve afişler, öğrencilere modülün temel özelliklerini ve faydalarını özetleyen görsel destekler olarak tasarlanmıştır. Bu materyaller, öğrencilerin öğrenme sürecine dair net bir anlayış geliştirmesine katkıda bulunmuştur.

Modülün etkili bir şekilde kullanılabilmesi ve öğrenci ihtiyaçlarına uygun hale getirilmesi amacıyla pilot uygulama yapılmıştır. Pilot uygulama her grupta uygulamaya başlamadan önce 2 ders saati olacak şekilde yapılmıştır. Pilot uygulama sürecinde, öğrencilerin modüle dair deneyimleri, karşılaştıkları zorluklar ve modülün etkili yönleri değerlendirilmiştir. Pilot uygulamadan elde edilen veriler doğrultusunda modül üzerinde tasarım ve erişim kolaylığı açısından düzeltmeler yapılmıştır. Modülün etkililiğini artırmak amacıyla modül tasarımı basitleştirilmiş ve sayfalar arası geçişler pilot uygulama sonrası geliştirilmiştir.

Ayrıca, geliştirme aşamasında öğrencilerin bireysel öğrenme ihtiyaçlarına yönelik olarak etkinlik yaprakları hazırlanmıştır (Ek 8.). Bu yapraklar, öğrencilerin farklı öğrenme hızlarına ve stillerine uygun aktiviteler sunarak kişiselleştirilmiş öğrenme deneyimlerini desteklemiştir. Tanıtım materyalleri ve etkinlik yaprakları, hem öğrenciler hem de öğretmenler için süreç içinde rehberlik sağlayan önemli araçlar olmuştur. Bu kapsamda yapılan çalışmalar, tasarlanan modülün öğrenciler için etkili, erişilebilir ve kullanıcı dostu bir eğitim platformu haline getirilmesine katkı sağlamıştır. Modülün geliştirilmesi aşamasında sağlanan geri bildirimler, hem teknik altyapının hem de içeriklerin sürekli olarak iyileştirilmesini mümkün kılmıştır. Bu sayede, modül, ortaokul öğrencilerinin programlama eğitimi ihtiyaçlarına uygun bir şekilde tasarlanmış ve geliştirilmiştir.

Uygulama Basamağı

Bu aşamada, üretken yapay zekâ destekli öğrenme modülü, deney grubu öğrencileri üzerinde uygulanmış ve modülün etkileri sistematik bir şekilde gözlemlenmiştir.

Deney grubu öğrencileri, üretken yapay zekâ destekli öğrenme modülünü kullanarak Python programlama dilini öğrenmişlerdir. Modül, her öğrenciye kişiselleştirilmiş bir öğrenme deneyimi sunmuş ve öğrenci ihtiyaçlarına uygun geri bildirim mekanizmaları sağlamıştır. Üretken yapay zekâ entegrasyonu, öğrencilerin yazdığı kodları analiz ederek hatalarını tespit etmiş, anında geri bildirim sunmuş ve çözüm önerileriyle öğrenme süreçlerini desteklemiştir.

Bu süreçte, öğrencilerin programlama konusundaki ilerlemeleri ve bilgi işlemsel düşünme becerileri düzenli olarak takip edilmiştir. Deney grubu öğrencilerinin modüle olan adaptasyon süreçleri ve modül kullanım sıklıkları analiz edilmiş, öğrencilerin performanslarına yönelik detaylı veriler toplanmıştır. Bu veriler, öğrencilerin bireysel gelişimlerini ve modülün öğrenme çıktıları üzerindeki etkilerini değerlendirmek için kullanılmıştır.

Kontrol grubu öğrencileri, geleneksel yöntemlerle Python programlama eğitimi almıştır. Bu gruptaki öğrencilerin öğrenme süreçleri, deney grubu ile kıyaslanabilir şekilde yapılandırılmıştır ancak eğitim süreci boyunca üretken yapay zekâ desteği kullanılmamıştır. Kontrol grubunda, öğretim süreci öğretmen merkezli bir yaklaşımla yürütülmüş, öğretmen derste temel programlama kavramlarını açıklamış, örnekler çözmüş ve öğrencilerden bireysel olarak uygulamalar yapmalarını istemiştir. Bu süreçte, öğretim teknikleri arasında açıklayıcı öğretim, problem çözme ve bireysel çalışma yöntemleri kullanılmıştır.

Deney grubunda ise öğrenciler, üretken yapay zekâ destekli öğrenme modülünü kullanarak Python programlama eğitimi almıştır. Bu modül, öğrencilere temel programlama kavramlarını öğrenme, kodlama becerilerini geliştirme ve hatalarını düzeltme süreçlerinde rehberlik etmiştir. Modül, öğrencilere bireysel öğrenme hızlarına göre içerik sunmuş, her bir öğrencinin yazdığı kodları analiz ederek anında geri bildirim sağlamış ve gerekli düzeltme önerilerinde bulunmuştur. Ayrıca, modül içerisinde yer alan interaktif alıştırmalar, öğrencilerin algoritmik düşünme ve problem çözme becerilerini geliştirmelerine olanak tanımıştır. Öğrenciler, modül aracılığıyla üretken yapay zekâyı programlama ile ilgili sorular sormuş, kodlama hatalarını tespit etmiş ve öneriler alarak kodlarını optimize etmişlerdir.

Her iki grubun performansı ve bilgi işlemsel düşünme becerileri, aynı ölçme araçları ve yöntemler kullanılarak değerlendirilmiştir. Ön test ve son test uygulamaları ile deney ve kontrol gruplarının başlangıç ve sonuç düzeyleri karşılaştırılmıştır. Bu yapılandırma, iki grup

arasındaki farklılıkların karşılaştırılabilir bir şekilde analiz edilmesine olanak sağlamış ve üretken yapay zekâ destekli öğrenme modülünün etkisinin daha kapsamlı bir şekilde incelenmesini sağlamıştır.

Her iki grup da haftalık değerlendirmeler ve görevlerle takip edilmiştir. Değerlendirmeler, öğrencilerin öğrenme süreçlerindeki ilerlemeleri ölçmek ve geri bildirim sağlamak amacıyla yapılandırılmıştır. Öğrencilerin ders içindeki etkileşimleri ve öğrenme süreçlerindeki etkinlik düzeyleri düzenli olarak gözlemlenmiştir. Deney grubu öğrencilerinin üretken yapay zekâ destekli modülü kullanım sıklıkları ve etkileşim düzeyleri sistem kayıtları ile takip edilmiş, modülün kullanıcı deneyimine ilişkin önemli veriler sağlamıştır.

Uygulama aşamasında toplanan tüm veriler, modülün etkilerinin değerlendirilmesi ve sonraki aşamalarda yapılacak düzenlemeler için bir temel oluşturmuştur. Bu süreç, hem modülün etkinliğini hem de geleneksel yöntemlere kıyasla sağladığı avantajları ortaya koymayı amaçlamıştır (Yıldırım & Şimşek, 2006). Bu kapsamda, uygulama basamağı, öğrenme modülünün potansiyelini ve eğitimde sağladığı yenilikleri değerlendirmek için kritik bir aşama olarak gerçekleştirilmiştir. Tablo 3'te uygulama sürecinde kullanılan kazanımlar ve etkinliklerin tablosu sunulmuştur.

Tablo 3. *Uygulama Süreci Kazanım ve Etkinlik Tablosu*

Hafta	Kazanımlar	Yapılacak Etkinlikler	Uygulama
1. Hafta	Değişkenlerde tutulacak verileri tanımlar. Bir değişkeni tanımlarken hangi tür veriyi tutacağını belirler.	Değişkenler ve veri türleri hakkında teorik ders. Öğrencilerle interaktif sorular ve cevaplar. Değişken tanımlama ve veri türleri üzerine basit kodlama alıştırmaları.	Geleneksel / Üretken Yapay Zeka ile
2. Hafta	Kullanıcıdan veri alır. Kullanıcıdan alınan veriyi değişkene tanımlar.	Kullanıcıdan veri alma ve değişkene atama üzerine kodlama örnekleri. Grup çalışmaları ve bireysel uygulamalar. Değişkenlere veri atama alıştırmaları.	Geleneksel / Üretken Yapay Zeka ile
3. Hafta	Python programlama dili karar yapılarını içeren programlar geliştirir.	Karar yapıları (if-else) ile ilgili teorik ders ve örnekler. Karar yapıları kullanarak basit projeler geliştirme. Gerçek dünya problemleri üzerine karar yapıları uygulamaları.	Geleneksel / Üretken Yapay Zeka ile

Tablo 3 (Devamı)

4. Hafta	Bir problemi çözmek için kullanacağı döngü yapısına karar verir. Çoklu koşulların mantıksal sonuçlarını değerlendirir.	Döngü yapıları (for, while) ve çoklu koşullar hakkında teorik dersler. Döngüler ve mantıksal koşullar üzerine kodlama alıştırmaları. Basit projelerde döngü yapıları kullanımı.	Geleneksel / Üretken Yapay Zeka ile
5. Hafta	Fonksiyonları program içinde yöntemlerine göre kullanır.	Fonksiyonların tanımlanması ve kullanımı üzerine ders. Fonksiyonlarla ilgili uygulamalı alıştırmalar. Küçük projelerde fonksiyon kullanımı ve modüler programlama teknikleri.	Geleneksel / Üretken Yapay Zeka ile
6. Hafta	Bütün kazanımların tekrarı ve proje çalışması.	Bütün konuların tekrarı ve öğrenci soruları. Öğrencilere verilen kapsamlı bir proje üzerinde çalışma. Proje sunumları ve geri bildirimler.	Geleneksel / Üretken Yapay Zeka ile

Değerlendirme Basamağı

Bu aşamada, uygulamanın etkilerinin sistematik bir şekilde analiz edilmesi ve öğrenme modülünün öğrenciler üzerindeki genel etkilerinin belirlenmesi hedeflenmiştir. Değerlendirme süreci, nicel ve nitel yöntemlerin bir arada kullanıldığı kapsamlı bir yaklaşımla gerçekleştirilmiştir.

Uygulamanın sonunda, öğrencilere son testler uygulanmış ve performansları çok yönlü olarak değerlendirilmiştir. Bu süreçte kullanılan ölçme araçları şunlardır:

- Bilgi İşlemsel Düşünme Öz Değerlendirme Ölçeği: Öğrencilerin bilgi işlemsel düşünme öz değerlendirmelerindeki değişimleri ölçmek amacıyla uygulanmıştır.
- Bilgi İşlemsel Düşünme Bebras Testi: Öğrencilerin bilgi işlemsel düşünme becerilerindeki değişimi ölçmek için kullanılmıştır.
- Programlamaya Yönelik Tutum Ölçeği: Öğrencilerin programlama süreçlerine yönelik tutumlarındaki değişimleri belirlemek amacıyla uygulanmıştır.
- Programlama Başarı Testi: Öğrencilerin programlama becerilerindeki gelişimi ve bilgi düzeylerindeki artışı değerlendirmek amacıyla uygulanmıştır.

Bu testlerden elde edilen veriler, deney ve kontrol gruplarının başlangıç ve sonuç performanslarının karşılaştırılması için kullanılmıştır. Veriler, nicel analiz yöntemleriyle değerlendirilerek modülün öğrencilerin öğrenme çıktıları üzerindeki etkisi incelenmiştir.

Değerlendirme sürecinde, öğrencilerin üretken yapay zekâ destekli öğrenme modülüne yönelik genel deneyimleri, ders sürecindeki gözlemler ve öğrencilerle yapılan görüşmeler aracılığıyla değerlendirilmiştir. Deney grubu öğrencilerinin ders esnasındaki etkileşimleri ve öğrenme süreçleri, öğretmen tarafından yapılan notlar ve sınıf içi gözlemlerle takip edilmiştir. Bu gözlemler, öğrencilerin modülü nasıl kullandıkları, öğrenme süreçlerindeki tepkileri ve karşılaştıkları zorlukları anlamak için bir temel sağlamıştır.

Öğrencilerle yapılan yarı yapılandırılmış görüşmeler, onların modül hakkındaki düşüncelerini, memnuniyet düzeylerini ve modülden öğrendiklerini anlamaya yönelik bilgiler sunmuştur. Görüşmelerden elde edilen nitel veriler, öğrencilerin öğrenme deneyimlerini anlamak ve modülün etkinliğini detaylı bir şekilde analiz etmek için kullanılmıştır.

Tüm bu değerlendirme sürecinden elde edilen veriler, üretken yapay zekâ destekli öğrenme modülünün etkili bir eğitim aracı olarak uygulanabilirliğini ve öğrenciler üzerindeki genel etkilerini ölçmek için bir temel oluşturmuştur. Bu bilgiler, modülün güçlü yönlerini ve geliştirilmesi gereken alanlarını belirlemeye yardımcı olmuştur. Nicel verilerle birlikte nitel verilerin analizi, araştırmanın bulgularına daha geniş bir perspektiften bakılmasını sağlamıştır.

Bu değerlendirme basamağı, araştırmanın tüm aşamalarından elde edilen verilerin bütüncül bir şekilde değerlendirilmesi ve sonuçların hem akademik hem de pratik açıdan yorumlanması için kritik bir süreç olarak tamamlanmıştır.

Nitel Veri Toplama ve Sonuçlandırma

Bu araştırmanın 6. Haftası, nitel veri toplama ve sonuçlandırma aşamasına ayrılmıştır. Bu aşamada, öğrencilerin üretken yapay zekâ destekli öğrenme modülüyle ilgili deneyimlerini derinlemesine anlamak ve modülün etkinliği hakkında kapsamlı bir değerlendirme yapmak amacıyla nitel veri toplama süreci yürütülmüştür. Elde edilen veriler, nicel sonuçlarla birleştirilerek araştırmanın genel amaçlarına ulaşılmasına katkıda bulunmuştur.

Nitel Veri Toplama Süreci

Bu aşamada, öğrencilerle yarı yapılandırılmış görüşmeler gerçekleştirilmiştir. Görüşmeler, modülün kullanım kolaylığı, etkisi ve öğrencilerin genel deneyimleri üzerine odaklanmıştır. Sorular, öğrencilerin modüle dair olumlu ve olumsuz geri bildirimlerini ve

önerilerini belirlemek amacıyla özenle hazırlanmıştır. Öğrencilerin öğrenme süreçleri sırasında karşılaştıkları zorluklar, modülün onlara sunduğu faydalar ve öğrenme süreçlerini nasıl etkilediği hakkında detaylı veriler elde edilmiştir.

Toplanan nitel veriler, içerik analizi yöntemiyle analiz edilmiştir. Görüşmelerden elde edilen veriler, belirli temalar ve kategoriler altında sınıflandırılmıştır. İçerik analizinden elde edilen bulgular, modülün etkili yönlerini, kullanıcı deneyimlerini ve geliştirilmesi gereken alanları kapsamlı bir şekilde ortaya koymuştur.

Nitel analiz sonuçları, araştırmada elde edilen nicel verilerle birleştirilerek değerlendirilmiştir. Bu bütüncül yaklaşım, üretken yapay zekâ destekli öğrenme modülünün öğrenciler üzerindeki etkilerini hem sayısal hem de derinlemesine bir şekilde analiz etmeyi mümkün kılmıştır. Nitel ve nicel sonuçlar birlikte ele alınarak, modülün programlama eğitimindeki rolü ve etkinliği hakkında kapsamlı bir rapor hazırlanmıştır.

Bu aşamada elde edilen bulgular, araştırmanın genel amaçlarına ulaşmasında kritik bir rol oynamıştır. Toplanan nitel veriler, öğrencilerin ve uzmanların modül hakkındaki algılarını ve deneyimlerini derinlemesine anlamaya olanak sağlamış, bu bilgiler modülün gelecekteki kullanım ve geliştirme süreçleri için rehberlik etmiştir. Araştırmanın sonuçları, üretken yapay zekâ destekli öğrenme modülünün eğitimdeki potansiyelini ortaya koymuş ve bu modülün etkili bir öğrenme aracı olarak uygulanabilirliğini desteklemiştir (Polat, 2022).

Verilerin Analizi

Bu araştırmada, hem nicel hem de nitel veriler toplanmış ve bu veriler, farklı analiz yöntemleri kullanılarak kapsamlı bir şekilde değerlendirilmiştir. Veri analizi süreci, araştırmanın amaçlarına uygun olarak sistematik bir şekilde planlanmış ve uygulanmıştır. Elde edilen verilerin geçerliği ve güvenilirliği, analiz süreçlerinin her aşamasında dikkate alınmıştır.

Nicel Verilerin Analizi

Araştırmanın nicel verileri, SPSS gibi istatistiksel analiz yazılımları aracılığıyla analiz edilmiştir. Verilerin analiz süreci, normallik varsayımına dayalı olarak parametrik ya da non-parametrik testlerin seçilmesiyle yapılandırılmıştır. Verilerin normal dağılım gösterip göstermediği çarpıklık ve basıklık değerlerine, ayrıca Shapiro-Wilk ve Kolmogorov-Smirnov testlerine göre değerlendirilmiştir.

Normallik varsayımının sağlandığı durumlarda şu analizler kullanılmıştır:

- Bağımsız Örneklem t-Testi: Deney ve kontrol gruplarının ön test ve son test puanları arasındaki farkların karşılaştırılması amacıyla uygulanmıştır. Bu test ile gruplar arasında başlangıç düzeyinde anlamlı bir fark olup olmadığı belirlenmiş, $p > 0.05$ olduğu durumlarda grupların homojen olduğu sonucuna varılmıştır.
- Bağımlı Örneklem t-Testi: Her grubun kendi içindeki (ön test-son test) gelişim düzeylerini analiz etmek için kullanılmıştır. Deney grubunda anlamlı gelişim gözlemlenirken, kontrol grubunda sınırlı değişim olduğu görülmüştür.

Normallik varsayımı sağlanmayan verilerde ise aşağıdaki non-parametrik testler uygulanmıştır:

- Mann-Whitney U Testi: Deney ve kontrol gruplarının son test puanları karşılaştırılmıştır. Bu test, bağımsız iki grubun sıralı verilerinin karşılaştırılması için uygundur.
- Wilcoxon İşaretli Sıralar Testi: Her bir grubun ön test ve son test puanları arasındaki farkı analiz etmek için uygulanmıştır. Bu test, özellikle küçük örneklem gruplarında ve normallik varsayımı sağlanmayan verilerde güçlü bir analiz sağlamaktadır.

Analizlerde etki büyüklüğü olarak Cohen's d değeri hesaplanmış, bu değerlere göre:

- $0.20 \leq d < 0.50$: Küçük etki
- $0.50 \leq d < 0.80$: Orta düzeyde etki
- $d \geq 0.80$: Büyük etki

Etki büyüklüğü değerleri, öğrenme modülünün deney grubu üzerindeki etkisinin hem istatistiksel hem de pedagojik olarak anlamlı olduğunu göstermektedir.

Ek olarak, analiz sürecinde her ölçek için betimsel istatistikler (ortalama, standart sapma, minimum ve maksimum değerler) sunularak gruplar arası farklılıkların daha iyi anlaşılması sağlanmıştır.

Nitel Verilerin Analizi

Nitel veriler, içerik analizi yöntemi kullanılarak tematik bir yaklaşımla değerlendirilmiştir. Nitel analiz süreci şu şekilde gerçekleştirilmiştir:

Görüşme verileri, belirli temalar ve kategoriler altında kodlanmıştır. Kodlama süreci, araştırma sorularına uygun anlamlı temalar oluşturmak için sistematik bir şekilde yürütülmüştür.

Öğrenci görüşlerinden elde edilen veriler, tematik analize tabi tutulmuştur. Bu analiz, modülün etkili yönlerini, öğrencilerin deneyimlerini ve geliştirilmeye açık alanları ortaya koymuştur.

Elde edilen bulgular, katılımcıların görüşleriyle doğrulanmıştır. Bu süreç, analiz sonuçlarının katılımcıların deneyimlerini doğru bir şekilde yansıttığını garanti altına almıştır.

Araştırmada kullanılan ölçme araçlarının geçerliliği ve güvenilirliği, titizlikle değerlendirilmiş ve bu kapsamda çeşitli yöntemler uygulanmıştır. Karma yöntem araştırmalarında geçerlilik ve güvenilirlik, hem nicel hem de nitel veri toplama süreçlerinde kullanılan araçların doğruluğu ve tutarlılığını sağlamak açısından kritik bir öneme sahiptir. Creswell ve Plano Clark (2011), karma yöntem araştırmalarında geçerlilik ve güvenilirlik sağlamanın, kullanılan yöntemlerin entegrasyonuna, veri kaynaklarının çeşitliliğine ve sonuçların yorumlanmasına dayandığını belirtmektedir. Bu doğrultuda, araştırma yöntemleri kitaplarından alınan ilkeler doğrultusunda çalışma süreci planlanmıştır.

Geçerlilik: Araştırmada kullanılan ölçme araçlarının içerik geçerliliği, uzman görüşleri ve literatür taraması yoluyla değerlendirilmiştir. Her bir ölçme aracının kapsamı, araştırmanın amaçları ve öğrencilerin özellikleri ile uyumlu olacak şekilde tasarlanmıştır. Programlamaya Yönelik Tutum Ölçeği ve Bilgi İşlemsel Düşünme Öz-Değerlendirme Ölçeği, literatürde geçerli ve güvenilir olarak kabul edilen ölçekler arasından seçilmiştir. Ölçeklerin içerik uygunluğu, eğitim teknolojileri ve ölçme değerlendirme alanında uzman olan üç akademisyen tarafından incelenmiş ve uygunluğu onaylanmıştır.

Araştırmanın yapı geçerliliği, ölçeklerin geliştirilme aşamasında yapılan faktör analizleri ve literatürdeki kullanımlarına dayanılarak sağlanmıştır. Nicel verilerin geçerliliğini artırmak için ön test ve son test uygulamalarından elde edilen veriler, istatistiksel analizlerle doğrulanmıştır.

Güvenilirlik: Ölçeklerin iç tutarlılığı, Cronbach Alfa katsayıları ile test edilmiş ve her iki ölçeğin güvenilirlik katsayılarının literatürde kabul edilen minimum değerin (0.70) üzerinde olduğu görülmüştür (Tavşancıl, 2019). Bu bulgu, ölçme araçlarının güvenilir sonuçlar üretebildiğini göstermektedir. Ayrıca, nitel veri toplama sürecinde yarı yapılandırılmış görüşmelerden elde edilen veriler, içerik analizi ile kodlanmış ve iki bağımsız araştırmacı tarafından değerlendirilerek değerlendiriciler arası güvenilirlik sağlanmıştır.

Pilot Çalışmalar: Veri toplama araçlarının pilot uygulamaları gerçekleştirilmiş ve elde edilen sonuçlar doğrultusunda gerekli düzenlemeler yapılmıştır. Pilot çalışmalarda,

öğrencilerin ölçme araçlarına verdikleri yanıtlar analiz edilerek anlaşılabilirlik ve geçerlilik açısından eksiklikler giderilmiştir. Bu süreç, veri toplama araçlarının hedef grup için uygun hale getirilmesini sağlamıştır.

Karma Yöntemlerde Geçerlilik ve Güvenilirlik Yaklaşımları: Creswell ve Plano-Clark (2011) ve Teddlie ve Tashakkori (2009), karma yöntem araştırmalarında geçerlilik ve güvenilirliği sağlamanın, veri triangülasyonu, yöntemsel entegrasyon ve sonuçların doğruluğunu teyit etmek için katılımcılardan geri bildirim almayı içerdiğini vurgulamaktadır. Bu doğrultuda, hem nicel hem de nitel veriler birlikte değerlendirilmiş ve sonuçların doğruluğu artırılmıştır. Triangülasyon yaklaşımı benimsenerek farklı veri kaynaklarından elde edilen bulgular karşılaştırılmış ve tutarlılık sağlanmıştır.

Bu önlemler, araştırma bulgularının geçerli, güvenilir ve genellenebilir olmasını sağlamış, aynı zamanda üretken yapay zekâ destekli öğrenme modülünün etkilerinin daha kapsamlı bir şekilde değerlendirilmesine olanak tanımıştır.

Araştırmacı Rolü

Araştırmacı, çalışmanın tüm aşamalarında aktif bir rol üstlenmiştir. Bu kapsamda, araştırmacı yalnızca veri toplama ve analiz süreçlerini yönetmekle kalmamış, aynı zamanda öğretim sürecinin tasarımı, uygulaması ve değerlendirilmesinde de doğrudan yer almıştır. Araştırmacının çalışmadaki rolleri şu şekilde detaylandırılabilir:

Araştırmacı, üretken yapay zekâ destekli öğrenme modülünün tasarımına liderlik etmiştir. Bu süreçte, modülün pedagojik ve teknik özelliklerinin belirlenmesi, öğrenci ihtiyaçlarının analiz edilmesi ve içerik geliştirme aşamalarında etkin bir rol oynamıştır. Öğretim materyalleri, interaktif etkinlikler ve testlerin geliştirilmesi sırasında, öğrencilerin öğrenme süreçlerini destekleyecek şekilde yapılandırılmış içeriklerin hazırlanmasını sağlamıştır.

Öğretim sürecinde ise araştırmacı, deney grubunda öğretmen rolü üstlenerek üretken yapay zekâ destekli öğrenme modülünün uygulanmasını gerçekleştirmiştir. Araştırmacı, öğrencilere modülün kullanımı konusunda rehberlik etmiş, öğrenme sürecinde karşılaşılan zorlukları belirlemek ve çözüm üretmek için aktif olarak destek sağlamıştır. Ayrıca, öğrencilerin modülden en verimli şekilde faydalanabilmeleri için öğrenme ortamını sürekli olarak gözlemlemiş ve gerekli düzenlemeleri yapmıştır.

Araştırmacı, üretken yapay zekâ destekli öğrenme modülünün teknik tasarım ve geliştirme süreçlerine de katkıda bulunmuştur. Bu kapsamda, modülün algoritmalarının

geliştirilmesi, yazılım araçlarının seçimi ve entegrasyon süreçlerinin planlanması gibi teknik çalışmaları yönetmiştir. Öğrencilerin ihtiyaçlarına uygun işlevsel bir sistemin geliştirilmesi için tasarım süreci boyunca geri bildirimler toplanmış ve sistemin kullanıcı dostu hale getirilmesi sağlanmıştır.

Araştırmacı, deney ve kontrol gruplarından veri toplama sürecini titizlikle yönetmiştir. Programlamaya yönelik tutum ölçeği, bilgi işlemsel düşünme öz-değerlendirme ölçeği ve diğer ölçme araçlarının uygulanması, verilerin toplanması ve analiz edilmesi süreçlerinde doğrudan görev almıştır. Bu süreçte, verilerin doğruluğunu ve güvenilirliğini sağlamak amacıyla detaylı bir planlama ve kontrol mekanizması yürütülmüştür.

Araştırmacı, öğretim sürecini dinamik bir şekilde izleyerek, öğrencilerden ve öğretim ortamından elde edilen geri bildirimleri değerlendirmiştir. Bu geri bildirimler doğrultusunda hem modül içeriği hem de uygulama stratejileri düzenlenmiştir. Bu süreç, araştırmanın tüm paydaşlarının ihtiyaçlarını karşılamaya yönelik esnek ve etkin bir yönetim yaklaşımı sergilenmesini sağlamıştır.

Bu çok yönlü katılım ve liderlik, araştırmanın her aşamasında elde edilen verilerin güvenilirliğini artırmış ve araştırmacının çalışmanın başarıyla yürütülmesinde kritik bir rol üstlenmesini sağlamıştır.

Geçerlik ve Güvenirlik

Araştırmanın geçerli ve güvenilir sonuçlar üretebilmesi amacıyla veri toplama araçlarının yapısı, uygulama süreci, örneklem seçim yöntemi ve analiz süreci dikkatle planlanmış ve açıklanmıştır.

Geçerlik

Bu çalışmada kullanılan yöntem, çalışmanın amacı ve araştırma sorularına uygun olarak sıralı açıklayıcı karma desen temelinde yapılandırılmıştır. Yöntem seçiminde literatür taraması doğrultusunda benzer çalışmalar incelenmiş ve yöntemin uygulanabilirliği pedagojik bağlamda gerekçelendirilmiştir. Örneklem, amaçlı ve ölçüt örnekleme yöntemleriyle belirlenmiş; katılımcıların özellikleri açıkça tanımlanmıştır. Uygulama süreci, nicel ve nitel veri toplama adımlarıyla birlikte detaylı biçimde betimlenmiştir. Veri toplama araçları ile uygulama ortamı araştırma sürecinin bütüncül yapısı içerisinde açıklanmış, kullanılan her aracın araştırma amacına hizmet edecek biçimde yapılandırılmasına özen gösterilmiştir. Araştırmanın varsayımları ve sınırlılıkları açıkça belirtilmiş; araştırmacının sürece yönelik rolü de

tanımlanmıştır. Bu kapsamda araştırma, kapsam geçerliği ve yöntem geçerliği ilkelerine göre yapılandırılmıştır.

Güvenirlik

Araştırmada güvenilirliği sağlamak amacıyla çeşitli stratejiler benimsenmiştir. Veri toplama araçlarının tasarımı aşamasında alan uzmanlarından görüş alınmış ve bu doğrultuda gerekli düzenlemeler yapılmıştır. Nicel veri toplama araçları için pilot uygulama gerçekleştirilmiş; gerekli madde analizleri ve iç tutarlılık katsayıları hesaplanmıştır. Nitel veri toplama sürecinde görüşmeler kayıt altına alınarak veri kaybı önlenmiştir. Nitel veriler iki ayrı araştırmacı tarafından kodlanmış; kodlayıcılar arası uyum oranı hesaplanarak güvenilirlik düzeyi kontrol edilmiştir. Ayrıca çalışmada veri çeşitlemesi (method triangulation) kullanılarak bulguların doğrulanabilirliği artırılmıştır. Araştırmanın her aşaması ayrıntılı olarak raporlanmış ve tekrarlanabilirliği sağlamak amacıyla tüm uygulama adımları belgelenmiştir.

DÖRDÜNCÜ BÖLÜM

Bulgular

Bu bölümde, araştırma kapsamında toplanan nicel ve nitel veriler analiz edilerek sunulmuştur. Araştırmada kullanılan Programlama Başarı Testi, Bilgi İşlemsel Düşünme Öz-Değerlendirme Ölçeği, Bebras Testi, Programlamaya Yönelik Tutum Ölçeği ve yarı yapılandırılmış görüşme formlarından elde edilen bulgular hem betimsel hem de istatistiksel analizlerle değerlendirilmiştir. Bulgular, deney ve kontrol gruplarının farklı değişkenler üzerindeki performanslarını karşılaştırmalı olarak açıklamaktadır. Bu bölümde, araştırma kapsamında toplanan verilerin istatistiksel analizleri sunulmaktadır.

Normallik Analizi Sonuçları

Araştırmada parametrik ya da parametrik olmayan istatistiksel analizlerin belirlenebilmesi için, verilerin normal dağılıma uygun olup olmadıkları Shapiro-Wilk testi ile incelenmiştir. Deney ve kontrol gruplarının çeşitli ölçüm puanlarına ait normallik test puanları Tablo 4’de verilmiştir. Shapiro-Wilk test puanlarına göre;

Deney Grubu’nda: Bebras Son Test Puanı ($p = .048$) ve Tutum Son Test Puanı ($p = .128$) normal dağılıma uygun değildir. Diğer değişkenler (Başarı, Bebras Ön Test, Tutum Ön Test, BİD Ön/Son Test Puanları) için $p > .05$ olduğundan normal dağılım varsayımı sağlanmaktadır.

Kontrol Grubu’nda: Tutum Son Test Puanı ($p = .051$) değeri sınırda olmakla birlikte; Bebras Ön Test ($p = .061$) ve Tutum Ön Test ($p = .079$) değişkenleri de normal dağılmamaktadır. Diğer tüm değişkenler için normallik varsayımı sağlanmıştır ($p > .05$).

Bu bulgular doğrultusunda, özellikle Bebras Son Test Puanı ve Tutum Son Test Puanı gibi değişkenlerde normallik sağlanamadığı için bu değişkenlerin analizinde parametrik olmayan testler (Wilcoxon İşaretli Sıralar Testi) tercih edilmiştir.

Tablo 4. *Değişkenlerin Normallik Analizi Puanları*

Değişkenler	Shapiro-Wilk	
	Deney Grubu	Kontrol Grubu
Başarı Puanı	.500	.845
Bebras Ön Test Puanı	.733	.061

Tablo 4 (Devamı)

Bebras Son Test Puanı	.048	.350
Tutum Ön Test Puanı	.544	.079
Tutum Son Test Puanı	.128	.051
BİD Ön Test Puanı	.068	.240
BİD Son Test Puanı	.445	.180

Bebras Testi değişkeni haricindeki değişkenlerin normal dağılım gösterdiği belirlenmiştir. ($p>0.05$) Ayrıca çarpıklık basıklık değerleri de incelenmiştir. +2 -2 aralığındaki değerler normal kabul edilmiştir (George & Mallery, 2010).

Bilgi İşlemsel Düşünmeye İlişkin Bulgular

Araştırmada üretken yapay zeka destekli öğrenme modülünün öğrencilerin bilgi işlemsel düşünmelerine etkisini ölçmek amacıyla Bilgi İşlemsel Düşünme Öz Değerlendirme Ölçeği kullanılmıştır.

Deney Grubu Bilgi İşlemsel Düşünme Ön-test Son-test Puanlarının Karşılaştırılması

Deney grubunda yer alan öğrencilerin Bilgi İşlemsel Düşünme (BİD) öz değerlendirme düzeylerine ilişkin ön test ve son test puanları karşılaştırılmış, elde edilen veriler bağımlı örneklem için t-testi ile analiz edilmiştir. Elde edilen sonuçlar Tablo 5’de sunulmuştur.

Tablo 5. Deney Grubu Bilgi İşlemsel Düşünme Ön-test Son-test Puanlarına İlişkin t-testi Puanları

Değişkenler	Ön Test		Son Test		$t(16)$	p	Cohen's d
	$\bar{X}$	SS	$\bar{X}$	SS			
Algoritmik Düşünme	3.15	.622	3.89	.524	-7.438	.001	1.287
Paralleleştirme	3.05	.709	3.84	.397	-6.406	.001	1.375
Ayrıştırma	3.75	.810	4.16	.432	-3.510	.003	0.632
Soyutlama	2.94	.669	3.80	.472	-5.913	.001	1.485
Otomasyon	3.29	.823	4.03	.633	-4.797	.001	1.008
Genel Toplam	3.23	.534	3.94	.324	-8.458	.001	1.608

Analiz sonuçları, öğrencilerin uygulama sürecinin ardından BİD öz değerlendirmelerinde istatistiksel olarak anlamlı ve yüksek düzeyde bir artış gösterdiğini, bu gelişimin ise tüm alt boyutlarda gözlemlendiğini ortaya koymaktadır. Ayrıca elde edilen Cohen's d değerleri, uygulamanın etki büyüklüğünün yüksek olduğunu göstermektedir. Algoritmik düşünme becerisine yönelik ön test puan ortalaması $\bar{X} = 3.15$ iken, uygulama sonrası bu ortalama $\bar{X} = 3.89$ 'a yükselmiştir. Bu artış istatistiksel olarak anlamlıdır ($t(16) = -7.438$, $p = .001$). Hesaplanan Cohen's d = 1.287 değeri, çok büyük düzeyde etki büyüklüğüne işaret etmektedir. Parallelistirme alt boyutuna ilişkin ön test ortalaması $\bar{X} = 3.05$, son test ortalaması ise $\bar{X} = 3.84$ olarak hesaplanmıştır. Bu fark anlamlıdır ($t(16) = -6.406$, $p = .001$) ve Cohen's d = 1.375 ile çok yüksek etki büyüklüğü göstermektedir. Ayırıştırma becerisine ait ön test puan ortalaması $\bar{X} = 3.75$, son testte $\bar{X} = 4.16$ 'ya yükselmiştir. Bu fark istatistiksel olarak anlamlı bulunmuştur ($t(16) = -3.510$, $p = .003$) ve etki büyüklüğü $d = 0.632$ olarak hesaplanmıştır. Soyutlama alt boyutunda en dikkat çekici artışlardan biri gözlemlenmiştir. Öğrencilerin ortalama puanı $\bar{X} = 2.94$ 'ten $\bar{X} = 3.80$ 'e yükselmiş, bu fark istatistiksel olarak anlamlı bulunmuştur ($t(16) = -5.913$, $p = .001$). Etki büyüklüğü ise Cohen's d = 1.485 ile çok büyük düzeydedir. Otomasyon becerisine ilişkin ön test ortalaması $\bar{X} = 3.29$, son testte $\bar{X} = 4.03$ 'e yükselmiş ve fark anlamlı bulunmuştur ($t(16) = -4.797$, $p = .001$). Etki büyüklüğü de oldukça yüksektir ($d = 1.008$). Genel toplam puanlarda da anlamlı bir artış görülmüştür. Ön testte $\bar{X} = 3.23$, son testte ise $\bar{X} = 3.94$ olarak ölçülmüş; bu fark yüksek düzeyde anlamlı bulunmuştur ($t(16) = -8.458$, $p = .001$) ve etki büyüklüğü Cohen's d = 1.608 olarak hesaplanmıştır. Bu değer, deney grubunda uygulanan öğretim sürecinin bilgi işlemsel düşünme becerilerine çok güçlü bir katkı sağladığını göstermektedir.

Kontrol Grubu Bilgi İşlemsel Düşünme Ön Test Son Test Verilerine İlişkin t-testi Puanları

Kontrol grubunda yer alan öğrencilerin Bilgi İşlemsel Düşünme (BİD)'ye ilişkin ön test ve son test puanları karşılaştırılmış, analizler bağımlı örneklemeler için t-testi ile gerçekleştirilmiştir. Elde edilen sonuçlar Tablo 6'da sunulmuştur.

Tablo 6. *Kontrol Grubu Bilgi İşlemsel Düşünme Ön Test Son Test Verilerine İlişkin t-testi Puanları*

Değişkenler	Ön Test		Son Test		$t(16)$	p	Cohen's d
	$\bar{X}$	SS	$\bar{X}$	SS			
Algoritmik Düşünme	3.21	.638	3.30	.596	-1.167	.260	-

Tablo 6 (Devamı)

Paralleleştirme	3.22	.932	3.40	.919	-3.665	.002	0.195
Ayrıştırma	3.48	.890	3.58	.809	-2.135	.049	0.118
Soyutlama	2.76	.733	3.21	.754	-5.004	.001	0.605
Otomasyon	3.19	.833	3.45	.781	-2.425	.028	0.322
Genel Toplam	3.20	.648	3.39	.598	-6.168	.001	0.305

Analiz sonuçları, kontrol grubundaki öğrencilerin bazı alt boyutlarda gelişim gösterdiğini ancak bu gelişimin genel olarak deney grubuna kıyasla daha sınırlı düzeyde kaldığını ortaya koymuştur (Tablo 6). Algoritmik düşünme alt boyutunda ön test ($\bar{X} = 3.21$, $SS = .638$) ve son test ($\bar{X} = 3.30$, $SS = .596$) puanları arasında anlamlı bir fark bulunmamıştır ($t(16) = -1.167$, $p = .260$). Bu durum, öğrencilerin adım adım düşünme ve planlı problem çözme becerilerinde belirgin bir gelişim göstermediğini göstermektedir. Paralleleştirme becerisine ilişkin puanlar anlamlı bir artış göstermiştir. Ön test ortalaması ($\bar{X} = 3.22$), son testte 3.40'a yükselmiş ve bu fark istatistiksel olarak anlamlı bulunmuştur ($t(16) = -3.665$, $p = .002$, Cohen's $d = 0.195$). Etki büyüklüğünün düşük düzeyde olması, gelişimin sınırlı kaldığını göstermektedir. Ayrıştırma becerisi açısından da anlamlı bir gelişim kaydedilmiştir. Öğrencilerin ön test puan ortalaması 3.48 iken, son testte bu değer 3.58'e çıkmıştır. Fark anlamlı bulunmuştur ($t(16) = -2.135$, $p = .049$, $d = 0.118$). Soyutlama alt boyutunda anlamlı bir gelişim gözlemlenmiştir. Ön test puan ortalaması ($\bar{X} = 2.76$, $SS = .733$), son testte ($\bar{X} = 3.21$, $SS = .754$) anlamlı bir artış göstermiştir ($t(16) = -5.004$, $p = .001$, $d = 0.605$). Bu, kontrol grubundaki öğrencilerin örüntü tanıma ve genelleme yapma becerilerinde kayda değer bir gelişim yaşadıklarını göstermektedir. Otomasyon becerisi alt boyutunda da istatistiksel olarak anlamlı bir fark tespit edilmiştir. Ön test ortalaması ($\bar{X} = 3.19$), son testte 3.45'e yükselmiştir ($t(16) = -2.425$, $p = .028$, $d = 0.322$). Genel toplam puanlar incelendiğinde, kontrol grubundaki öğrencilerin bilgi işlemsel düşünme becerilerinde anlamlı bir gelişme kaydettikleri görülmektedir. Ön test puan ortalaması ($\bar{X} = 3.20$, $SS = .648$), son testte 3.39'a yükselmiş ve bu fark istatistiksel olarak anlamlı bulunmuştur ($t(16) = -6.168$, $p = .001$, Cohen's $d = 0.305$).

Deney ve Kontrol Gruplarının Bilgi İşlemsel Düşünme Son Test Puanlarının Karşılaştırılması

Deney ve kontrol grubu öğrencilerinin Bilgi İşlemsel Düşünme (BİD)'ye ilişkin son test puanları karşılaştırılmış ve elde edilen veriler bağımsız örneklem için t-testi ile analiz edilmiştir. Elde edilen bulgular Tablo 7'de sunulmuştur.

Tablo 7. *Deney ve Kontrol Gruplarının Bilgi İşlemsel Düşünme Son Test Puanları*

Değişkenler	Deney		Kontrol		t(16)	p	Cohen's d
	$\bar{X}$	SS	$\bar{X}$	SS			
Algoritmik Düşünme	3.89	.524	3.30	.596	3.05	.005	1.051
Paralleleştirme	3.84	.379	3.40	.919	1.84	.075	-
Ayrıştırma	4.16	.432	3.58	.809	2.57	.015	0.894
Soyutlama	3.80	.472	3.21	.754	2.72	.010	0.938
Otomasyon	4.03	.633	3.45	.781	2.41	.022	0.816
Genel Toplam	3.94	.324	3.39	.598	3.30	.002	1.144

Analiz bulguları, deney grubundaki öğrencilerin bilgi işlemsel düşünmelerini genel olarak kontrol grubuna göre daha yüksek düzeyde değerlendirdiklerini göstermektedir. Özellikle algoritmik düşünme, ayrıştırma, soyutlama ve otomasyon alt boyutlarında anlamlı farklılıklar gözlenmiştir. Algoritmik düşünme boyutunda; Deney grubunun algoritmik düşünmeye ilişkin son test puan ortalaması ($\bar{X} = 3.89$), kontrol grubunun ortalamasından ($\bar{X} = 3.30$) anlamlı düzeyde yüksektir ($t(16) = 3.05$, $p = .005$). Etki büyüklüğü ise Cohen's d = 1.051 olarak bulunmuş, bu değer yüksek düzeyde etkiyi işaret etmektedir. Bu bulgu, deney grubu öğrencilerinin problemi adım adım çözme ve planlama becerilerinde gelişim gösterdiğini ortaya koymaktadır. Paralleleştirme alt boyutunda deney grubu öğrencilerinin ortalaması ($\bar{X} = 3.84$), kontrol grubuna göre daha yüksek olmasına rağmen ($\bar{X} = 3.40$), bu fark istatistiksel olarak anlamlı bulunmamıştır ($t(16) = 1.84$, $p = .075$). Bu durum, iki grup arasında paraleleştirme becerilerinde belirgin bir farklılık olmadığını göstermektedir. Ayrıştırma becerisi açısından deney grubu öğrencilerinin puan ortalaması ($\bar{X} = 4.16$), kontrol grubunun ortalamasından ($\bar{X} = 3.58$) anlamlı düzeyde yüksektir ($t(16) = 2.57$, $p = .015$, $d = -0.894$). Bu fark, yüksek düzeyde etki büyüklüğü ile desteklenmiş olup, deney grubunun karmaşık problemleri parçalara ayırma

konusunda daha başarılı olduğunu göstermektedir. Soyutlama alt boyutunda da deney grubunun ortalaması ($\bar{X} = 3.80$), kontrol grubuna göre anlamlı düzeyde daha yüksek bulunmuştur ($\bar{X} = 3.21$; $t(16) = 2.72$, $p = .010$, $d = -0.938$). Bu bulgu, öğrencilerin örüntü tanıma ve genelleme yapma becerilerinde deneysel uygulamanın etkili olduğunu göstermektedir. Otomasyon becerilerinde deney grubunun son test puan ortalaması ($\bar{X} = 4.03$), kontrol grubunun puan ortalamasından ($\bar{X} = 3.45$) anlamlı düzeyde yüksek bulunmuştur ($t(16) = 2.41$, $p = .022$, $d = -0.816$). Bu fark, öğrencilerin tekrar eden işlemleri daha sistematik şekilde otomatikleştirme becerisi kazandıklarını göstermektedir.

Genel toplam puanlar açısından da deney grubundaki öğrenciler ($\bar{X} = 3.94$, $SS = .324$), kontrol grubundaki öğrencilere ($\bar{X} = 3.39$, $SS = .598$) göre istatistiksel olarak anlamlı düzeyde daha yüksek bir performans sergilemiştir ($t(16) = 3.30$, $p = .002$, Cohen's $d = -1.144$). Etki büyüklüğünün yüksek olması, uygulanan yapay zeka destekli programlama modülünün bilgi işlemsel düşünme üzerinde güçlü ve anlamlı bir etkiye sahip olduğunu göstermektedir.

Bebras Testine İlişkin Bulgular

Bu bölümde, öğrencilerin bilgi işlemsel düşünme becerilerini ölçmeyi amaçlayan Bebras Bilge Kunduz Testi'nden elde edilen verilerin analizine yer verilmiştir. Testte yer alan sorular; kolay, orta ve zor düzey olarak sınıflandırılmış ve öğrencilerin her düzeydeki performansları ayrı ayrı analiz edilmiştir.

Deney ve kontrol grubu öğrencilerinin ön test ve son test puanları karşılaştırılmış; ayrıca gruplar arası farklılıkları belirlemek amacıyla bağımsız örneklem analizleri yapılmıştır. Verilerin normal dağılıma uygunluk durumuna göre parametrik veya parametrik olmayan testler tercih edilmiş; anlamlılık düzeyi .05 olarak alınmıştır. Aşağıda, her bir soru düzeyine ve toplam puana ilişkin analiz bulguları sunulmuştur.

Deney Grubu Bebras Ön Test Son Test Puanlarının Karşılaştırılması

Deney grubundaki öğrencilerin Bebras Bilge Kunduz Testi'ne ait ön test ve son test puanları, Wilcoxon işaretli sıralar testi ile analiz edilmiştir. Test, zorluk düzeyine göre kolay (6 puan $\times$ 5 soru = 30 puan), orta (9 puan $\times$ 5 = 45 puan) ve zor (12 puan $\times$ 5 = 60 puan) olmak üzere 3 grupta toplam 15 sorudan ve 135 puan üzerinden oluşmaktadır. Her bir düzeydeki puanlar ayrı ayrı değerlendirilmiş ve yorumlanmıştır. Bulgular Tablo 8'de sunulmuştur.

Tablo 8. Deney Grubu Bebras Ön Test Son Test Puanları

Zorluk Düzeyi	Ön Test		Son Test		Z	p
	$\bar{X}$	SS	$\bar{X}$	SS		
Bebras Kolay	11.76	4.30	13.53	5.80	-1.857	.063
Bebras Orta	11.12	7.48	14.82	7.75	-2.333	.020
Bebras Zor	12.71	9.92	16.24	8.42	-1.890	.059
Genel Toplam	37.94	11.37	47.29	14.09	-2.943	.003

Kolay düzey sorularda öğrencilerin ön test puan ortalaması $\bar{X} = 11.76$ iken, son test ortalaması $\bar{X} = 13.53$ olarak belirlenmiştir. Artış olmasına rağmen fark istatistiksel olarak anlamlı bulunmamıştır ($Z = -1.857$, $p = .063$). Bu durum, kolay sorulara yönelik performansta sınırlı bir gelişimin olduğunu ancak anlamlılık sınırında kaldığını göstermektedir. Orta düzey sorularda ise anlamlı bir artış gözlemlenmiştir. Ön test puan ortalaması $\bar{X} = 11.12 / 45$ (%24.7) iken, son test puanı $\bar{X} = 14.82 / 45$ (%32.9) olarak hesaplanmıştır. Bu fark istatistiksel olarak anlamlıdır ($Z = -2.333$, $p = .020$) ve hesaplanan etki büyüklüğü $r = .40$, orta düzey bir etkiye işaret etmektedir. Bu sonuç, öğrencilerin orta zorluktaki bilgi işlemsel düşünme sorularında anlamlı kazanım elde ettiğini göstermektedir. Zor düzey sorularda öğrencilerin ön test ortalaması $\bar{X} = 12.71 / 60$ (%21.2), son test ortalaması ise $\bar{X} = 16.24 / 60$ (%27.1) olarak belirlenmiştir. Gözlemlenen artışa rağmen fark istatistiksel olarak anlamlı değildir ($Z = -1.890$, $p = .059$). Bu durum, öğrencilerin zor sorularda gelişim gösterdiğini ancak bu farkın istatistiksel anlamlılığa ulaşmadığını göstermektedir. Genel toplamda ise puan ortalaması ön testte $\bar{X} = 37.94 / 135$ (%28.1) iken, son testte $\bar{X} = 47.29 / 135$ (%35.0)'a yükselmiştir. Bu fark istatistiksel olarak anlamlı bulunmuştur ($Z = -2.943$, $p = .003$). Hesaplanan etki büyüklüğü $r = .50$, uygulamanın orta-yüksek düzeyde etkili olduğunu göstermektedir. Bu bulgu, genel başarı açısından öğrencilerin Bebras testinde uygulama sonrasında kayda değer bir ilerleme sağladığını ortaya koymaktadır.

Kontrol Grubu Bebras Ön Test Son Test Puanlarının Karşılaştırılması

Kontrol grubundaki öğrencilerin Bebras Bilge Kunduz Testi'ne ait ön test ve son test puanları Wilcoxon işaretli sıralar testi ile analiz edilmiştir. Analiz sonuçları Tablo 9'da verilmiştir.

Tablo 9. *Kontrol Grubu Bebras Ön Test Son Test Puanları*

Zorluk Düzeyi	Ön Test		Son Test		Z	p
	$\bar{X}$	SS	$\bar{X}$	SS		
Bebras Kolay	9.71	4.13	11.76	5.28	-2.333	.020
Bebras Orta	9.00	9.00	10.59	10.18	-1.732	.083
Bebras Zor	9.88	12.17	10.59	11.91	-.577	.564
Genel Toplam	30.53	20.57	35.29	22.18	-2.534	.011

Kolay düzey sorularda öğrencilerin ön test puan ortalaması $\bar{X} = 9.71$, son test ortalaması ise $\bar{X} = 11.76$ olarak hesaplanmıştır. Bu artış istatistiksel olarak anlamlı bulunmuştur ($Z = -2.333$, $p = .020$). Bu durum, öğrencilerin temel düzeyde bilgi işlemsel düşünme gerektiren sorularda belirgin bir gelişim gösterdiğini ortaya koymaktadır. Orta düzey sorulara ilişkin ön test puanı $\bar{X} = 9.00$, son test puanı ise $\bar{X} = 10.59$ 'dur. Gözlenen artışa rağmen fark istatistiksel olarak anlamlı bulunmamıştır ($Z = -1.732$, $p = .083$). Bu sonuç, öğrencilerin orta düzey sorular karşısında sınırlı düzeyde gelişim gösterdiğini göstermektedir. Zor düzey sorulara yönelik puanlarda neredeyse hiçbir değişim gözlemlenmemiştir. Ön test puanı $\bar{X} = 9.88$, son test puanı ise $\bar{X} = 10.59$ 'dur. İlgili fark istatistiksel olarak anlamlı değildir ($Z = -.577$, $p = .564$). Bu durum, öğrencilerin ileri düzey problem çözme becerilerinde gelişim göstermediklerini ortaya koymaktadır. Genel toplam puanlar açısından anlamlı bir artış kaydedilmiştir. Öğrencilerin ön testteki ortalama puanı $\bar{X} = 30.53$, son testte $\bar{X} = 35.29$ 'a yükselmiştir. Bu fark anlamlı bulunmuştur ($Z = -2.534$, $p = .011$), bu da öğrencilerin genel test performansında anlamlı bir gelişme yaşadığını göstermektedir.

Deney ve Kontrol Gruplarının Bebras Son Test Puanlarının Karşılaştırılması

Deney ve kontrol grubu öğrencilerinin Bebras Bilge Kunduz Testi son test puanları, bağımsız örneklem için Mann-Whitney U testi ile karşılaştırılmıştır. Testin kolay, orta ve zor düzey soruları ayrı ayrı değerlendirilmiş ve ayrıca genel toplam puanlar analiz edilmiştir. Analiz sonuçları Tablo 10'da sunulmuştur.

Tablo 10. *Deney ve Kontrol Gruplarının Bebras Son Test Puanları*

Zorluk Düzeyi	Deney	Kontrol	Z	p
	$\bar{X}$	$\bar{X}$		
Bebras Kolay	19.09	15.91	-.979	.327
Bebras Orta	19.79	15.21	-1.405	.160
Bebras Zor	20.21	14.79	-.1676	.094
Genel Toplam	20.50	14.50	-1.763	.078

Kolay düzey sorularda, deney grubu ortalaması $\bar{X} = 19.09$, kontrol grubu ortalaması ise $\bar{X} = 15.91$ olarak belirlenmiştir. Fark anlamlı bulunmamıştır ($Z = -0.979$, $p = .327$, $r = .17$). Bu bulgu, iki grubun temel düzeydeki bilgi işlemsel düşünme becerileri arasında anlamlı bir farklılık olmadığını göstermektedir. Orta düzey sorulara ait son test puan ortalaması deney grubunda $\bar{X} = 19.79$, kontrol grubunda ise $\bar{X} = 15.21$ olarak ölçülmüştür. Gözlenen fark istatistiksel olarak anlamlı değildir ($Z = -1.405$, $p = .160$, $r = .24$). Ancak etki büyüklüğü $r = .24$ ile küçük-orta düzeyde bir etkiyi işaret etmektedir. Zor düzey sorularda, deney grubunun ortalaması $\bar{X} = 20.21$, kontrol grubunun ortalaması ise $\bar{X} = 14.79$ 'dur. İki grup arasındaki fark istatistiksel olarak anlamlı bulunmamıştır ($Z = -1.676$, $p = .094$, $r = .29$). Etki büyüklüğü açısından değerlendirildiğinde, bu farkın orta düzeyde bir etkiye sahip olduğu söylenebilir. Tüm sorulara ait toplam puanlar karşılaştırıldığında, deney grubunun genel başarı düzeyi ($\bar{X} = 20.50$) kontrol grubuna ($\bar{X} = 14.50$) göre daha yüksek bulunmuştur. Ancak bu fark da istatistiksel olarak anlamlı değildir ($Z = -1.763$, $p = .078$, $r = .30$). Etki büyüklüğü $r = .30$ ile orta düzey bir farka işaret etmektedir. Bu bulgular, deney grubunun son test performanslarının ortalama olarak daha yüksek olmasına rağmen, bu farkların istatistiksel olarak anlamlılık düzeyine ulaşmadığını göstermektedir. Ancak özellikle zor düzey sorularda ve genel toplamda orta düzey etkiler dikkat çekmektedir.

Programlamaya Yönelik Tutum Ölçeğine İlişkin Bulgular

Bu bölümde, öğrencilerin programlamaya yönelik tutumlarını ölçmek amacıyla uygulanan Programlamaya Yönelik Tutum Ölçeği'nden elde edilen verilerin analizine yer verilmiştir. Ölçek, öğrencilerin programlamaya karşı istek, inanç ve ilgi düzeylerini değerlendirmeyi amaçlamaktadır. Deney ve kontrol gruplarındaki öğrencilerin ön test ve son test puanları karşılaştırılmış, ayrıca gruplar arası farklılıklar analiz edilmiştir.

Deney Grubu Programlamaya Yönelik Tutum Ön Test Son Test Puanlarının Karşılaştırılması

Deney grubundaki öğrencilerin programlamaya yönelik tutumlarına ilişkin ön test ve son test puanları karşılaştırılmış, elde edilen veriler bağımlı örneklem için t-testi ile analiz edilmiştir. Analiz sonuçları Tablo 11’de sunulmuştur.

Tablo 11. *Deney Grubu Programlamaya Yönelik Tutum Ön Test Son Test Puanları*

Değişkenler	Ön Test		Son Test		$t(16)$	p	Cohen's d
	$\bar{X}$	SS	$\bar{X}$	SS			
Programlamaya Yönelik İstek	3.89	.474	4.41	.430	-9.826	.001	1.149
Programlamaya Yönelik İnanç	4.00	.754	4.23	.598	-3.165	.006	0.338
Programlamaya Yönelik İlgi	2.94	.637	3.84	.635	-7.707	.001	1.415
Genel Toplam	3.71	.388	4.25	.364	-14.039	.001	1.435

Deney grubundaki öğrencilerin programlamaya yönelik istek düzeyleri anlamlı biçimde artmıştır. Ön testte $\bar{X} = 3.89$, son testte $\bar{X} = 4.41$ olan puanlar arasındaki fark istatistiksel olarak anlamlıdır ($t(16) = -9.826$, $p = .001$) ve Cohen's $d = 1.149$ değeri, büyük düzeyde etki büyüklüğünü göstermektedir. Bu bulgu, öğrencilerin programlama sürecine daha fazla dahil olma isteği kazandıklarını ortaya koymaktadır. Öğrencilerin programlamaya yönelik yeterlilik algılarında da anlamlı bir gelişme gözlemlenmiştir. Ön test puan ortalaması $\bar{X} = 4.00$, son testte $\bar{X} = 4.23$ 'e yükselmiştir. Bu artış istatistiksel olarak anlamlıdır ($t(16) = -3.165$, $p = .006$) ancak Cohen's $d = 0.338$ değeri, düşük-orta düzeyde bir etki büyüklüğü olduğunu göstermektedir. Bu bulgu, öğrencilerin programlamayı başarabileceklerine olan inançlarında olumlu bir değişim olduğunu göstermektedir. Programlamaya olan ilgi düzeyi de anlamlı bir artış göstermiştir. Ön testte $\bar{X} = 2.94$ olan ortalama, son testte $\bar{X} = 3.84$ 'e yükselmiştir. Fark yüksek düzeyde anlamlıdır ($t(16) = -7.707$, $p = .001$) ve Cohen's $d = 1.415$ değeri, çok büyük etki büyüklüğünü işaret etmektedir. Bu durum, öğrencilerin programlamaya olan meraklarının ve içsel motivasyonlarının arttığını göstermektedir. Genel toplam puanlar incelendiğinde, öğrencilerin programlamaya yönelik tutumlarında genel olarak anlamlı ve güçlü bir gelişim olduğu

görülmektedir. Ön testte $\bar{X} = 3.71$, son testte $\bar{X} = 4.25$ 'e yükselmiş; bu fark istatistiksel olarak anlamlı bulunmuştur ($t(16) = -14.039$, $p = .001$) ve Cohen's $d = 1.435$ ile çok büyük düzeyde bir etki gözlemlenmiştir.

Kontrol Grubu Programlamaya Yönelik Tutum Ön Test Son Test Puanlarının Karşılaştırılması

Kontrol grubundaki öğrencilerin programlamaya yönelik tutumlarına ilişkin ön test ve son test puanları karşılaştırılmış, analizler bağımlı örneklemeler için t-testi ile gerçekleştirilmiştir. Elde edilen bulgular Tablo 12'de sunulmuştur.

Tablo 12. *Kontrol Grubu Programlamaya Yönelik Tutum Ön Test Son Test Puanları*

Değişkenler	Ön Test		Son Test		$t(16)$	p	Cohen's d
	$\bar{X}$	SS	$\bar{X}$	SS			
Programlamaya Yönelik İstek	3.77	1.05	3.93	.969	-2.970	.009	0.158
Programlamaya Yönelik İnanç	3.58	.975	3.66	.942	-2.219	.041	0.083
Programlamaya Yönelik İlgi	3.29	.880	3.66	.799	-4.642	.001	0.44
Genel Toplam	3.63	.777	3.81	.772	-5.913	.001	0.232

Kontrol grubundaki öğrencilerin programlamaya yönelik istek düzeylerinde istatistiksel olarak anlamlı bir artış gözlemlenmiştir. Ön test puan ortalaması $\bar{X} = 3.77$, son testte $\bar{X} = 3.93$ 'e yükselmiştir ($t(16) = -2.970$, $p = .009$). Ancak Cohen's $d = 0.158$ değeri, bu farkın etki büyüklüğünün düşük düzeyde olduğunu göstermektedir. Bu bulgu, öğrencilerin programlamaya olan katılım isteğinde sınırlı bir gelişme yaşadığını ortaya koymaktadır. Programlamaya yönelik inanç düzeyinde de anlamlı bir artış olduğu görülmüştür. Ön test ortalaması $\bar{X} = 3.58$, son testte $\bar{X} = 3.66$ 'ya yükselmiştir ($t(16) = -2.219$, $p = .041$). Ancak Cohen's $d = 0.083$, çok düşük düzeyde bir etkiye işaret etmektedir. Bu sonuç, öğrencilerin programlamaya dair başarı algılarının yalnızca küçük bir ölçüde değiştiğini göstermektedir. Programlamaya yönelik ilgi düzeyinde ise diğer alt boyutlara kıyasla daha belirgin bir artış gözlemlenmiştir. Ön test puanı $\bar{X} = 3.29$, son test puanı ise $\bar{X} = 3.66$ olarak ölçülmüş; bu fark istatistiksel olarak anlamlı

bulunmuştur ($t(16) = -4.642$, $p = .001$) ve Cohen's $d = 0.44$ ile orta düzeyde etki büyüklüğü göstermektedir. Genel toplam puanlar incelendiğinde, öğrencilerin programlamaya yönelik genel tutumlarında anlamlı bir artış olduğu görülmektedir. Ön test puan ortalaması $\bar{X} = 3.63$, son testte $\bar{X} = 3.81$ 'e yükselmiş; bu fark anlamlı bulunmuştur ($t(16) = -5.913$, $p = .001$) ancak etki büyüklüğü $d = 0.232$ ile düşük düzeyde kalmıştır.

Deney ve Kontrol Gruplarının Programlamaya Yönelik Tutum Son Test Puanlarının Karşılaştırılması

Deney ve kontrol grubu öğrencilerinin Programlamaya Yönelik Tutum Ölçeği son test puanları bağımsız örneklem için t-testi ile karşılaştırılmıştır. Analiz sonuçları Tablo 13'de verilmiştir.

Tablo 13. *Deney ve Kontrol Gruplarının Programlamaya Yönelik Tutum Son Test Puanları*

Değişkenler	Deney		Kontrol		$t(16)$	p	Cohen's d
	$\bar{X}$	SS	$\bar{X}$	SS			
Programlamaya Yönelik İstek	4.41	.430	3.93	.969	1.858	.072	-
Programlamaya Yönelik İnanç	4.23	.598	3.66	.942	2.100	0.44	-0.722
Programlamaya Yönelik İlgi	3.84	.635	3.66	.799	.712	.481	-
Genel Toplam	4.25	.364	3.81	.772	2.089	.045	-0.729

Deney grubu öğrencilerinin programlamaya yönelik istek puan ortalaması $\bar{X} = 4.41$, kontrol grubu öğrencilerinin ortalaması ise $\bar{X} = 3.93$ olarak belirlenmiştir. Bu fark istatistiksel olarak anlamlı bulunmamıştır ($t(16) = 1.858$, $p = .072$). Cohen's d değeri tabloya yansıtılmamış olmakla birlikte, farkın anlamlı olmaması nedeniyle etki düzeyi de düşüktür. İnanç alt boyutunda deney grubunun ortalaması $\bar{X} = 4.23$, kontrol grubunun ise $\bar{X} = 3.66$ olarak ölçülmüştür. Fark istatistiksel olarak anlamlıdır ($t(16) = 2.100$, $p = .044$) ve Cohen's $d = -0.722$ değeri, yüksek düzeyde etki büyüklüğünü göstermektedir. Bu sonuç, deney grubunun programlamaya yönelik yeterlilik inancının kontrol grubuna göre belirgin şekilde daha yüksek olduğunu ortaya koymaktadır. İlgi alt boyutunda her iki grubun puan ortalamaları birbirine

yakın bulunmuştur. Deney grubunun ortalaması $\bar{X} = 3.84$, kontrol grubunun ise $\bar{X} = 3.66$ 'dır. Bu fark istatistiksel olarak anlamlı değildir ($t(16) = 0.712$, $p = .481$). Genel toplam puanlar açısından deney grubu öğrencilerinin ortalaması $\bar{X} = 4.25$, kontrol grubu öğrencilerinin ortalaması ise $\bar{X} = 3.81$ 'dir. Bu fark istatistiksel olarak anlamlı bulunmuştur ($t(16) = 2.089$, $p = .045$) ve Cohen's $d = -0.729$ olarak hesaplanmıştır. Bu değer, uygulamanın programlamaya yönelik tutuma genel düzeyde orta-üst düzeyde etkili olduğunu göstermektedir.

Programlama Başarı Testine İlişkin Bulgular

Bu bölümde, öğrencilerin Python programlama dilinde algoritma kurma, hata ayıklama ve temel programlama yapılarını kullanma becerilerini ölçmek amacıyla hazırlanan Programlama Başarı Testi'nden elde edilen verilerin analizine yer verilmiştir. Test, toplam 6 açık uçlu sorudan oluşmakta olup öğrencilerin problem çözme, algoritmik düşünme ve kod yazma yetkinliklerini ölçmeyi amaçlamaktadır.

Verilerin analizinde, deney ve kontrol gruplarının ön test ve son test puanları ayrı ayrı karşılaştırılmış; ayrıca gruplar arası son test puanları incelenerek uygulamanın etkisi değerlendirilmiştir.

Deney ve Kontrol Gruplarının Başarı Son Test Puanlarının Karşılaştırılması

Deney ve kontrol grubu öğrencilerinin Programlama Başarı Testi'ne ait son test puanları bağımsız örneklemeler için t-testi ile karşılaştırılmıştır. Analiz sonuçları Tablo 14'de sunulmuştur.

Tablo 14. Deney ve Kontrol Gruplarının Programlama Başarı Son Test Puanları

Değişkenler	Deney		Kontrol		p	t	Cohen's d
	$\bar{X}$	SS	$\bar{X}$	SS			
Programlama Başarı Testi	82.94	12.12	71.47	12.96	.012	2.664	-0.914

Yapılan analiz sonucunda, deney grubundaki öğrencilerin programlama başarı düzeylerinin ($\bar{X} = 82.94$, $SS = 12.12$) kontrol grubuna ($\bar{X} = 71.47$, $SS = 12.96$) göre anlamlı düzeyde daha yüksek olduğu belirlenmiştir ($t(16) = 2.664$, $p = .012$). Bu farkın Cohen's $d = -0.914$ olarak hesaplanan etki büyüklüğü, uygulamanın programlama başarısı üzerinde yüksek düzeyde etkili olduğunu göstermektedir.

ÜYZ Tabanlı Programlama Modülünün Öğrenciler Üzerindeki Etkilerine İlişkin Nicel ve Nitel Bulguların Entegre Edilmiş Sonuçları

Bu bölümde, deney grubundaki öğrencilerle gerçekleştirilen yarı yapılandırılmış görüşmelerden elde edilen nitel veriler, uygulamanın etkilerini daha derinlemesine değerlendirmek amacıyla nicel verilerle ilişkilendirilerek sunulmuştur. Toplamda deney grubundan 10 öğrenciyle bireysel görüşmeler gerçekleştirilmiş ve bu görüşmeler içerik analizi tekniğiyle çözümlenmiştir. Öğrenci görüşlerinden elde edilen bulgular, ilgili ölçeklerin nicel verileriyle birlikte değerlendirildiğinde, öğrencilerin programlamaya yönelik bilişsel becerilerinde ve tutumsal yaklaşımlarında olumlu yönde gelişim gösterdikleri anlaşılmaktadır. Bu gelişim, öğrencilerin problem çözme, algoritmik düşünme, paralel süreçleri planlama, soyutlama ve otomasyon gibi temel bilgi işlemsel düşünme becerilerinde ilerleme kaydettiklerine ilişkin ifadeleriyle de desteklenmiştir. Öğrenciler ayrıca, programlamaya yönelik ilgi, istek ve öz-yeterlik inançlarının arttığını, yapay zekâ destekli uygulamaların öğrenme sürecini kolaylaştırdığını ve daha motive edici hâle getirdiğini belirtmişlerdir. Görüşmelerde öne çıkan ifadeler, öğrencilerin sürece aktif katılım gösterdiklerini ve önceki deneyimlerine kıyasla daha fazla başarı hissi yaşadıklarını göstermektedir. Uygulama sürecinin kalıcı öğrenmeyi destekleyici olduğu, öğrencilerin öğrenme sürecinde daha özgüvenli hissettikleri ve problem çözme stratejilerinde gelişim gösterdikleri sıkça dile getirilmiştir. Bu kapsamda, öğrencilerden elde edilen nitel veriler, ilgili ölçeklerin alt boyutlarıyla ilişkilendirilerek aşağıda tablolar hâlinde sunulmuştur.

ÜYZ Tabanlı Programlama Modülünün Bilgi İşlemsel Düşünmeye Etkisine İlişkin Karma Yöntem Entegre Sonuçları

Bu tema kapsamında öğrencilerin algoritmik düşünme, paralelleştirme, ayrıştırma, soyutlama ve otomasyon becerilerine ilişkin görüşleri analiz edilmiştir. Öğrenciler, kodlama sürecinde problemleri çözmek için adım adım düşünme, aynı anda birden fazla işlemi değerlendirme, karmaşık yapıları parçalara ayırma, ortak örüntülerden genelleme yapma ve tekrar eden işlemleri otomatikleştirme becerilerinde gelişim gösterdiklerini ifade etmişlerdir. Öğrencilerin bir kısmı algoritmik düşünmenin kendilerini yönlendirmede yardımcı olduğunu belirtirken, bazı öğrenciler de kod yazarken artık belirli bir sistematik içinde ilerlediklerini vurgulamıştır. Parallelleştirme konusunda, döngü ve koşullu ifadeleri birlikte kullanabilme becerisinin geliştiği ifade edilmiştir. Özellikle ayrıştırma ve soyutlama alt boyutlarında öğrencilerin problem çözme sürecine daha stratejik yaklaşımları anlaşılmıştır. Bu bulgular,

nicel verilerde gözlenen anlamlı artışları destekler nitelikte olup, öğrencilerin bilgi işlemsel düşünmeye ilişkin temel kavramları sadece öğrenmedikleri, aynı zamanda uygulamaya da dökabildiklerini göstermektedir. İlgili öğrenci görüşlerinden örnekler ve detaylı açıklamalar Tablo 15–19 arasında sunulmuştur.

Tablo 15. *BİD Algoritmik Düşünme Alt Boyutu Öğrenci Görüşleri*

Nicel Veriler	Nitel Veriler	Örnek Öğrenci Görüşleri
Deney grubunun ön test ortalaması orta düzeyden yüksek düzeye çıkmış, bu fark yüksek etki büyüklüğünde ve istatistiksel olarak anlamlı bulunmuştur	Öğrencilerin çoğu, YZ destekli programlama modülüyle bireysel olarak etkileşime geçtikleri etkinlikler sırasında, problem çözme sürecinde adım adım düşünme, sıraya koyma ve sistemli ilerleme gibi becerilerde gelişim yaşadıklarını ifade etmiştir. Özellikle modülde yer alan yönlendirmeler ve kod yazım adımlarına ilişkin açıklamalar sayesinde, “önce ne yapılması gerektiğini düşünme” alışkanlığı kazandıkları vurgulanmıştır. Öğrenciler, ÜYZ’nin sunduğu öneriler ve geri bildirimler doğrultusunda kendi çözüm yollarını yapılandırabildiklerini dile getirmiştir.	Ö1: “Kod yazarken artık önce ne yapacağımı düşünüyorum, kafamda bir sıra oluşuyor.” Ö3: “Eskiden rastgele yazardım, şimdi önce ne yapılacak onu belirliyorum.” Ö6: “Soruları adım adım çözmeyi öğrendim, artık kafam karışmıyor.” Ö9: “Kodlamada artık kafamda bir yol haritası çiziyorum.” Ö12: “Önce ne olacak, sonra hangi işlem gelecek diye düşünerek yazıyorum.”

Tablo 16. *BİD Paralelleştirme Alt Boyutu Öğrenci Görüşleri*

Nicel Veriler	Nitel Veriler	Örnek Öğrenci Görüşleri
Deney grubunun ortalaması orta düzeyden yüksek düzeye çıkmış, bu fark yüksek etki büyüklüğünde ve anlamlı bulunmuştur. Kontrol grubunda da anlamlı fark vardır ancak etki büyüklüğü düşüktür.	Öğrenciler, kodlama sürecinde birden fazla koşul ya da işlemi aynı anda düşünme, mantıksal olarak paralel çalışan komutları planlama becerilerinde gelişme yaşadıklarını belirtmişlerdir. Bu becerinin özellikle, modülde yer alan senaryo tabanlı görevlerde aynı anda birden fazla koşulun sağlanmasını gerektiren problem yapıları sayesinde geliştiği ifade edilmiştir. Ayrıca, modülde örneklerle sunulan döngü ve koşullu ifadelerin birlikte kullanıldığı kod parçaları, öğrencilerin bu yapılar arasındaki ilişkiyi fark etmelerine katkı sağlamıştır.	Ö2: “Hem döngü hem şart kullanmak önce zor geliyordu ama artık birlikte yazabiliyorum.” Ö5: “İki şeyi aynı anda düşündüğümde hangisinin önce çalışacağını görebiliyorum.” Ö8: “Artık aynı anda çalışan kodları anlayabiliyorum, neyin neyi etkilediğini çözüyorum.” Ö13: “Önceden karışıyordu, şimdi aynı anda birden fazla şey yapabilen kodlar yazıyorum.” Ö15: “Şartlı ifadeyle döngüyü bir arada kullanmayı bu modülde öğrendim.”

Tablo 17. BİD Ayrıştırma Alt Boyutu Öğrenci Görüşleri

Nicel Veriler	Nitel Veriler	Örnek Öğrenci Görüşleri
Deney grubunda orta düzeyden çok yakına artış olmuş, fark orta düzeyde etkiyle anlamlıdır. Kontrol grubunda da anlamlı fark vardır ancak etki oldukça düşüktür.	Öğrenciler, büyük ve karmaşık problemleri daha küçük parçalara bölerek çözme becerisi kazandıklarını ifade etmişlerdir. Bu becerinin gelişmesinde, modülde sunulan problemleri bölüm bölüm ilerleyen adımlarla çözmelerine olanak tanıyan yapı, öğrencilerin çözüm sürecini adım adım düşünmesine yardımcı olmuştur. Ayrıca, YZ desteğiyle sunulan parça parça yönlendirme ve açıklamalar, öğrencilerin problemi anlamlandırıp alt parçalara ayırarak hataları daha kolay ayıklamalarını sağlamıştır.	Ö4: “Problemi parçalara ayırınca daha kolay çözüyorum. Hepsine birden bakınca kafam karışıyordu.” Ö7: “Hangi kısmın yanlış olduğunu ayrı ayrı düşündüğümde bulmak kolay oluyor.” Ö10: “Eskiden tek parça düşünüyordum, şimdi önce bölüp sonra çözmeye çalışıyorum.” Ö11: “Soruyu bölümlere ayırınca hataları daha net görebiliyorum.” Ö16: “Karmaşık kodları parçalara bölerek yazmak bana zaman kazandırıyor.”

Tablo 18. BİD Soyutlama Alt Boyutu Öğrenci Görüşleri

Nicel Veriler	Nitel Veriler	Örnek Öğrenci Görüşleri
Deney grubunda orta düzeyden yüksek düzeye belirgin bir artış olmuş, bu fark çok büyük etkiyle anlamlı bulunmuştur. Kontrol grubunda da anlamlı fark vardır ancak etki büyüklüğü orta düzeydedir.	Öğrenciler, benzer problemler arasında örüntüleri fark etme ve önceki çözümlerden genellemeler yapma becerilerinde gelişim yaşadıklarını belirtmişlerdir. Bu becerinin özellikle, modülde yer alan örnek problemlerle çalışma ve önceki çözümlere benzer yeni problemleri çözme fırsatlarının etkisiyle geliştiği ifade edilmiştir. Öğrenciler, yapay zekâ tarafından önerilen çözüm yollarını analiz ederek yeni karşılaştıkları problemleri çözmede bu örüntüleri kullanabildiklerini dile getirmiştir.	Ö3: “Bu kod daha önce yaptığımızın aynısı gibiydi, mantığını çözünce kolay oldu.” Ö6: “Benzer şeyler görünce artık hemen anlayabiliyorum, örnekler işe yarıyor.” Ö9: “Bir kod yapısını başka soruda kullanınca çok kolay çözdüm.” Ö12: “Bazı şeyler birbirine çok benziyor, artık onları ayırt edip genelleştirebiliyorum.” Ö14: “Daha önce yazdığımız kodlar bana yeni sorularda yol gösterdi.”

Tablo 19. BİD Otomasyon Alt Boyutu Öğrenci Görüşleri

Nicel Veriler	Nitel Veriler	Örnek Öğrenci Görüşleri
Deney grubunda orta düzeyden yüksek düzeye çıkmış, fark büyük etkiyle anlamlı bulunmuştur. Kontrol grubunda da anlamlı fark gözlenmiş ancak etki düzeyi düşüktür.	Öğrenciler, tekrar eden işlemleri döngü veya fonksiyon yapılarıyla daha verimli bir biçimde uygulamayı öğrendiklerini ifade etmişlerdir. Bu becerinin gelişiminde, modülde yer alan otomasyon odaklı görevler, aynı işlemi farklı yöntemlerle uygulama gerektiren uygulamalar ve YZ'nin sunduğu önerilerle kodların sadeleştirilmesi gibi özelliklerin etkili olduğu belirtilmiştir. Kodlama sürecinde manuel işlemler yerine otomatik çalışan yapılar kurarak zaman kazandıklarını ve daha kısa kodlarla aynı işlevi gerçekleştirdiklerini vurgulamışlardır.	Ö1: “Artık aynı kodu tekrar tekrar yazmak yerine döngü kuruyorum.” Ö5: “Benzer işlemleri otomatikleştirdiğimde hem kısa sürüyor hem hata olmuyor.” Ö8: “Fonksiyon yazmayı öğrendim, tekrar eden işleri onunla çözüyorum.” Ö13: “Bir komutu birden fazla yerde kullanmam gerekiyorsa otomatik hale getiriyorum.” Ö15: “Önceden elle yapıyordum, şimdi döngüyle kodu sadeleştiriyorum.”

ÜYZ Tabanlı Programlama Modülünün Programlamaya Yönelik Etkisine İlişkin Karma Yöntem Entegre Sonuçları

Bu tema, öğrencilerin programlamaya yönelik istek, inanç ve ilgi düzeylerindeki değişimlere odaklanmaktadır. Öğrencilerin büyük çoğunluğu uygulama süreci boyunca programlamaya olan ilgilerinin arttığını, etkinliklerin eğlenceli ve dikkat çekici olduğunu belirtmiştir. Özellikle üretken yapay zeka ile görev tamamlama ve anında geri bildirim alma, öğrencilerin derse yönelik motivasyonlarını artırmıştır. İstek boyutunda öğrenciler, uygulama yaparken daha istekli olduklarını, ders saatlerini beklediklerini belirtmiştir. İnanç boyutunda, “yapamam” düşüncesinin zamanla yerini “başarıyorum” duygusuna bıraktığı ifade edilmiştir. İlgi boyutunda ise öğrenciler, artık programlamayı sadece okulda değil, kendi başlarına da öğrenmeye çalıştıklarını dile getirmiştir. Bu bulgular, nicel analizlerde görülen tutumsal

gelişimi desteklemektedir. Özellikle deney grubundaki öğrencilerde gözlenen yüksek etki büyüklüğü, görüşme verileriyle paralellik göstermektedir. İlgili öğrenci görüşlerine yer verilen örnekler Tablo 20–22’de sunulmuştur.

Tablo 20. *Programlamaya Yönelik Tutum İstek Alt Boyutu Öğrenci Görüşleri*

Nicel Veriler	Nitel Veriler	Örnek Öğrenci Görüşleri
Deney grubunun ön test ortalaması yüksek düzeye çıkmış, fark büyük etkiyle ve istatistiksel olarak anlamlı bulunmuştur. Kontrol grubunda da anlamlı fark gözlenmiş, ancak etki büyüklüğü düşüktür.	Öğrencilerin büyük çoğunluğu, programlamaya yönelik isteklerinin arttığını ifade etmiştir. Bu artışın özellikle, modülde yer alan uygulamalı etkinliklerin öğrenciyi aktif katılıma teşvik etmesi, görsel ve etkileşimli içeriklerle öğrenme sürecini desteklemesi ve üretken yapay zekâ desteğiyle anında geri bildirim alma imkânı sunması sayesinde gerçekleştiği belirtilmiştir. Öğrenciler, bu özelliklerin dersi daha ilgi çekici ve anlamlı kıldığını; etkinlikleri tekrar etmek ve derse daha istekli katılmak istediklerini ifade etmişlerdir.	Ö3: “Derslerde kod yazacağımız zaman daha çok istekle derse geliyorum.” Ö4: “Bazen tekrar yapmak istiyorum, çünkü kod yazmak eğlenceli hale geldi.” Ö7: “Eskiden sadece izlerdim, artık ben de yapmak istiyorum.” Ö11: “Kodlama etkinlikleri dersin en sevdiğim kısmı oldu.” Ö15: “Yapay zekayla bir şey üretince hevesim artıyor.”

Tablo 21. *Programlamaya Yönelik Tutum İnanç Alt Boyutu Öğrenci Görüşleri*

Nitel Veriler	Nitel Veriler	Örnek Öğrenci Görüşleri
Deney grubunun ortalaması anlamlı biçimde artmış, bu fark düşük-orta düzeyde bir etki göstermiştir. Kontrol grubunda da anlamlı bir fark vardır, ancak etki çok düşüktür.	Öğrencilerin önemli bir kısmı, programlama yapabileceklerine dair inançlarının arttığını dile getirmiştir. Bu inanç gelişiminde, modülde yer alan basitten karmaşığa yapılandırılmış görev akışı, öğrencilerin küçük başarılar yaşayarak ilerlemeleri, ve üretken yapay zekânın yapıcı geri bildirim ve ipuçları sunarak süreci desteklemesi belirleyici olmuştur. Öğrenciler, görevleri başarıyla tamamladıkça kendilerine olan güvenlerinin arttığını ve kod yazmaya karşı daha olumlu bir bakış geliştirdiklerini ifade etmişlerdir.	Ö2: “Eskiden kod yazamam sanıyordum ama artık yapabileceğimi biliyorum.” Ö6: “Zorlandığım yerlerde yardım alıp devam ettiğimde kendime güvenim arttı.” Ö8: “Kodları çalıştırabildikçe ‘ben bunu yapabiliyorum’ dedim.” Ö10: “En başta çok zordu ama sonra kendi kendime de çözümler buldum.” Ö14: “Programlamanın bana uygun olmadığını düşünüyordum ama artık yapabileceğime inanıyorum.”

Tablo 22. *Programlamaya Yönelik Tutum İlgili Alt Boyutu Öğrenci Görüşleri*

Nicel Veriler	Nitel Veriler	Örnek Öğrenci Görüşleri
Deney grubunun ortalaması orta düzeyden yüksek düzeye çıkmış, fark çok büyük etkiyle anlamlı bulunmuştur. Kontrol grubunda da anlamlı fark vardır, ancak etki orta düzeydedir.	Öğrencilerin çoğu, programlamaya yönelik ilgilerinin belirgin biçimde arttığını belirtmiştir. Bu artışın temelinde, modülün öğrencileri sadece bilgi veren değil, üretmeye teşvik eden etkileşimli bir yapıda olması, ayrıca üretken yapay zekâ aracılığıyla kişiselleştirilmiş içerik ve öneriler sunması etkili olmuştur. Öğrenciler, bu süreçte programlamaya dair meraklarının arttığını, ders dışında da araştırmalar yaptıklarını ve farklı programlama dillerine karşı ilgi geliştirdiklerini ifade etmişlerdir.	Ö1: “Kod yazmak artık ilgimi çekiyor, boş zamanlarımda da araştırıyorum.” Ö5: “Python dışında başka diller de var mı diye bakmaya başladım.” Ö9: “Oyun yapmak için hangi dilleri bilmem gerektiğini araştırdım.” Ö12: “Bu dersten sonra yazılım kulübüne katılmak istedim.” Ö13: “Yapay zekayla birleşince programlama çok daha ilginç geldi.”

Programlama Başarısı Teması

Bu temada öğrencilerin programlama başarılarına yönelik öz değerlendirmeleri yer almaktadır. Öğrenciler, uygulama süreci boyunca açık uçlu programlama sorularını daha kolay çözebildiklerini, kod yazma becerilerinin geliştiğini ve hata ayıklama konusunda daha dikkatli olduklarını ifade etmişlerdir. Bazı öğrenciler daha önce sadece hazır örnekleri izlediklerini ancak şimdi kendi başlarına çözüm üretebildiklerini belirtmiş; bazıları ise önceki hatalarını analiz ederek yeniden doğru çözüme ulaşabildiklerini söylemiştir. Öğrencilerin kodlama konusundaki öz yeterlik algılarında artış olduğu gözlemlenmiştir. Bu bulgular, Programlama Başarı Testi’ne ilişkin nicel verilerle örtüşmektedir. Deney grubunun başarı düzeyinin kontrol grubuna göre anlamlı derecede yüksek olması, öğrencilerin görüşleriyle desteklenmektedir. İlgili öğrenci alıntıları ve bulgular Tablo 23’de ayrıntılı biçimde sunulmuştur.

Tablo 23. *Programlama Başarısı Öğrenci Görüşleri*

Nicel Veriler	Nitel Veriler	Örnek Öğrenci Görüşleri
Deney grubunun son test puanı, kontrol grubuna göre anlamlı düzeyde yüksek bulunmuş, fark büyük etki büyüklüğünde ve istatistiksel olarak anlamlıdır.	Öğrenciler, kodlama sorularını çözme konusunda uygulama süreci boyunca daha fazla deneyim kazandıklarını ve başarı düzeylerinin arttığını belirtmiştir. Özellikle yapay zeka desteğiyle verilen ipuçlarının ve yapılandırılmış adımların, soruları bağımsız çözmelerine katkı sağladığı ifade edilmiştir.	Ö4: “Kod yazarken artık ne yapmam gerektiğini daha hızlı anlıyorum.” Ö7: “Soruyu görünce ne tür yapı kullanacağımı tahmin edebiliyorum.” Ö10: “Eskiden yardım almadan çözemiyordum, şimdi çoğunu kendim yapıyorum.” Ö11: “En başta hata bulmak çok zordu ama artık nerede sıkıntı var anlayabiliyorum.” Ö16: “Açık uçlu sorularda kendi başıma çözüm üretebildiğimi görmek hoşuma gidiyor.”

BEŞİNCİ BÖLÜM

Tartışma ve Sonuç

Ortaokul programlama eğitiminde üretken yapay zeka destekli öğrenme modülünün; öğrencilerin bilgi işlemsel düşünceleri, programlamaya yönelik tutumları ve programlama başarıları açısından toplanan verilerin analizlerine ait bulgular, alan yazında yer alan çalışmalar ile birlikte tartışılarak detaylı bir şekilde bu bölümde sunulmuştur.

Öğrencilerinin Bilgi İşlemsel Düşüncelerine Yönelik Sonuçları ve Tartışma

Araştırma kapsamında elde edilen bulgular, deney grubundaki öğrencilerin bilgi işlemsel düşünme becerilerinde uygulama öncesine kıyasla anlamlı ve yüksek düzeyde bir gelişim gösterdiğini ortaya koymuştur. Öğrencilerin öz değerlendirme düzeyleri orta düzeyden yüksek düzeye çıkmıştır. Bu gelişim, sadece bilişsel değil aynı zamanda öğrencilerin düşünme süreçlerini yapılandırma biçimlerinde de değişim yaratmıştır. Özellikle algoritmik düşünme boyutunda öğrencilerin sistematik düşünme, adım adım planlama ve kodlama sürecini bir akışa yerleştirme becerilerinde anlamlı ilerleme kaydettikleri görülmüştür. Nitel verilerde öğrenciler, kod yazarken artık rastgele değil, planlı ve sıralı ilerlediklerini; neyin önce, neyin sonra yapılması gerektiğini daha net kavradıklarını belirtmişlerdir. Bu durum, üretken yapay zeka destekli modülün sunduğu aşamalandırılmış yönlendirmeler, hata analizleri ve geribildirimli görev yapısı sayesinde gelişmiş olabilir. Özellikle ÜYZ sistemlerinin öğrenmeyi kişiselleştirmesi, bireysel eksiklere göre yönlendirmesi ve düşünce sürecine aktif müdahil olması, öğrencilerin algoritmik düşünme becerilerini derinleştirmektedir (Park & Jo, 2023; Jiang vd., 2023). Paralleştirme becerisi açısından da öğrenciler, birden fazla süreci aynı anda düşünebilme, döngülerle koşulları birlikte kullanabilme ve zihinsel olarak komutları gruplayarak çalıştırma konusunda daha başarılı olduklarını ifade etmişlerdir. Bu gelişimin, modülde yer alan çok adımlı senaryolar, görselleştirilmiş işlem akışları ve anında öneriler sunan ÜYZ etkileşimleri sayesinde desteklendiği düşünülmektedir. Öğrenciler, daha önce karmaşık gelen çoklu yapıları artık daha rahat çözümleyebildiklerini belirtmişlerdir. Bu, ÜYZ'nin kodlama öğretiminde kavramsal yükü azaltma ve karmaşık yapıları sadeleştirerek sunma gücünün bir yansımasıdır (Almutairi vd., 2024; Shin vd., 2023). Kontrol grubundaki öğrencilerde ise bazı alt boyutlarda sınırlı gelişimler gözlenmiş, ancak bu gelişimlerin uygulamadaki bireysel yönlendirme ve aktif etkileşim eksikliği nedeniyle daha düşük düzeyde kaldığı söylenebilir. Bu durum, sadece içerik sunumunun değil, öğrenme sürecine kişisel katkı

sunan destekleyici teknolojilerin önemini bir kez daha ortaya koymaktadır. Bu bulgular, ÜYZ teknolojilerinin yalnızca teknik bir araç değil; aynı zamanda bilişsel gelişimi yönlendiren pedagojik bir bileşen olarak da kullanılabileceğini göstermektedir. Literatürde de ÜYZ sistemlerinin algoritmik düşünme, örüntü tanıma ve çoklu süreç yönetimi gibi temel bilgi işlemsel düşünme becerilerinde anlamlı katkılar sunduğu ifade edilmektedir (Bali vd., 2023; Kim & Lee, 2023). Bu sayede öğrenciler, programlamada farklı yapıların birlikte çalıştığını yalnızca öğrenmekle kalmamış, aynı zamanda bu yapıları içselleştirerek zihinsel şemalarına entegre etmişlerdir. Paralleleştirme becerisindeki bu gelişim, öğrencilerin sistematik düşünmenin ötesine geçerek çok boyutlu problem çözme yetilerini geliştirdiklerini göstermektedir. Bu, sadece programlama değil, matematiksel düşünme, mühendislik uygulamaları ve günlük yaşamda karar alma gibi alanlara da doğrudan katkı sağlayabilecek nitelikte bir beceridir.

Deney grubundaki öğrencilerin ayrıştırma (decomposition) becerilerinde istatistiksel olarak anlamlı bir gelişme gösterdiğini ortaya koymuştur. Nicel veriler incelendiğinde, deney grubunun ön test ortalaması son test ortalamasına göre yükselmiş; bu fark anlamlı bulunmuştur. Hesaplanan Cohen's d değeri, uygulamanın bu beceri üzerinde orta-üst düzeyde etkili olduğunu göstermektedir. Kontrol grubunda da anlamlı bir fark tespit edilmiş olsa da, bu farkın etki büyüklüğü oldukça düşüktür, bu da gelişimin sınırlı düzeyde kaldığını göstermektedir. Bu durum, kontrol grubundaki öğrencilerin yalnızca temel düzeyde içerik sunumuna maruz kalmaları ve bireysel yönlendirme ya da yapay zekâ destekli etkileşimli ortamdan yoksun olmalarıyla açıklanabilir. Öğrenciler, kendi başlarına ilerlemek zorunda kaldıkları bu süreçte, hatalarını anlama ve düzeltme konusunda yeterince destek alamamış olabilirler. Bu da öğrenme sürecinin derinliğini ve sürdürülebilirliğini sınırlamış olabilir. Nitel veriler de bu bulguları güçlü biçimde desteklemektedir. Görüşme yapılan öğrenciler, karmaşık problemleri daha küçük parçalara bölerek çözüm üretmeyi uygulamalar sayesinde öğrendiklerini belirtmişlerdir. Öğrenciler, özellikle kodlama sürecinde hataları daha rahat fark edebilmek için kodu bölümlere ayırma, önce sorunun “küçük parçalarını çözme”, ardından bütünü birleştirme eğiliminde olduklarını ifade etmişlerdir. Bu, ayrıştırma becerisinin hem hata ayıklama sürecinde hem de mantıksal bütünlük kurmada etkin rol oynadığını göstermektedir. Ayrıştırma becerisi, Wing (2006) tarafından bilgi işlemsel düşünmenin temel unsurlarından biri olarak tanımlanmış; karmaşık yapıların daha küçük, yönetilebilir alt yapılara bölünmesi olarak açıklanmıştır. Ayrıca Brennan ve Resnick (2012), bu becerinin erken yaşlardan itibaren kazandırılmasının öğrencilerin bilişsel yükünü azalttığını ve problem çözme stratejilerini daha etkili kullanmalarını sağladığını ifade etmektedir. Bu araştırmada kullanılan yapay zekâ destekli modül, öğrencilere problemleri sadece çözme değil, aynı zamanda analiz etme, bölümlendirme

ve yeniden yapılandırma fırsatları sunmuştur. Öğrenciler bir problemi çözmeden önce hangi bölümlerin kod içinde nasıl temsil edileceğini düşünme alışkanlığı kazanmıştır. Bu sayede yalnızca çözüm değil, çözümün yapısı üzerine de düşünceleri teşvik edilmiştir. Ayırıştırma becerisine ilişkin bu gelişim, öğrencilerin problem çözme yaklaşımında derinleştiğini, sadece sonucu değil, çözüm sürecini yönetme yetisi kazandıklarını göstermektedir. Bu beceri, programlama dışında matematiksel modelleme, sistem analizi, mühendislik tasarımı gibi birçok alanda da kullanılabilecek kritik bir düşünme biçimidir. Öğrencilerin bu farkındalığı erken yaşta kazanması, onları karmaşık problemleri yönetebilen bireyler hâline getirme yolunda önemli bir adımdır.

Nicel veriler incelendiğinde, deney grubunun ön test ortalaması son test ortalamasına göre yükselmiş; bu fark anlamlı bulunmuştur. Hesaplanan Cohen's d değeri, uygulamanın bu beceri üzerinde orta-üst düzeyde etkili olduğunu göstermektedir. Kontrol grubunda da anlamlı bir fark tespit edilmiş olsa da, Cohen's d değeri, bu gelişimin oldukça sınırlı olduğunu göstermektedir. Nitel veriler de bu bulguları güçlü biçimde desteklemektedir. Görüşme yapılan öğrenciler, karmaşık problemleri daha küçük parçalara bölerek çözüm üretmeyi uygulamalar sayesinde öğrendiklerini belirtmişlerdir. Öğrenciler, özellikle kodlama sürecinde hataları daha rahat fark edebilmek için kodu bölümlere ayırma, önce sorunun "küçük parçalarını çözme", ardından bütünü birleştirme eğiliminde olduklarını ifade etmişlerdir. Bu, ayırıştırma becerisinin hem hata ayıklama sürecinde hem de mantıksal bütünlük kurmada etkin rol oynadığını göstermektedir. Ayırıştırma becerisi, Wing (2006) tarafından bilgi işlemsel düşünmenin temel unsurlarından biri olarak tanımlanmış; karmaşık yapıların daha küçük, yönetilebilir alt yapılara bölünmesi olarak açıklanmıştır. Ayrıca Brennan ve Resnick (2012), bu becerinin erken yaşlardan itibaren kazandırılmasının öğrencilerin bilişsel yükünü azalttığını ve problem çözme stratejilerini daha etkili kullanmalarını sağladığını ifade etmektedir. Bu araştırmada kullanılan yapay zekâ destekli modül, öğrencilere problemleri sadece çözme değil, aynı zamanda analiz etme, bölümlendirme ve yeniden yapılandırma fırsatları sunmuştur. Öğrenciler bir problemi çözmeden önce hangi bölümlerin kod içinde nasıl temsil edileceğini düşünme alışkanlığı kazanmıştır. Bu sayede yalnızca çözüm değil, çözümün yapısı üzerine de düşünceleri teşvik edilmiştir. Ayırıştırma becerisine ilişkin bu gelişim, öğrencilerin problem çözme yaklaşımında derinleştiğini, sadece sonucu değil, çözüm sürecini yönetme yetisi kazandıklarını göstermektedir. Bu beceri, programlama dışında matematiksel modelleme, sistem analizi, mühendislik tasarımı gibi birçok alanda da kullanılabilecek kritik bir düşünme biçimidir. Öğrencilerin bu farkındalığı erken yaşta kazanması, onları karmaşık problemleri yönetebilen bireyler hâline getirme yolunda önemli bir adımdır.

Deney grubundaki öğrencilerin otomasyon becerilerinde anlamlı ve güçlü bir gelişme gösterdiğini ortaya koymuştur. Nicel verilere göre, deney grubunun ön test puan ortalaması son teste göre yükselmiş; bu fark istatistiksel olarak anlamlı bulunmuştur. Hesaplanan Cohen's d değeri, uygulamanın büyük düzeyde etki yarattığını göstermektedir. Kontrol grubunda da anlamlı bir artış tespit edilmiştir, ancak etki büyüklüğü düşük düzeydedir. Nitel veriler de bu bulgularla paraleldir. Öğrencilerle yapılan görüşmelerde, tekrar eden işlemleri daha kısa ve etkili yollarla çözmeyi öğrendikleri; döngüler, fonksiyonlar ve yapılandırılmış bloklarla kodlarını otomatikleştirebildikleri görülmüştür. Öğrenciler, başlangıçta aynı kod parçalarını manuel olarak yazarken, uygulama süreci sonunda bu işlemleri tek bir komut veya döngü aracılığıyla çözebildiklerini ifade etmişlerdir. Bu ifadeler, öğrencilerin verimlilik odaklı düşünme ve optimizasyon becerilerinde gelişim gösterdiğini ortaya koymaktadır. Otomasyon becerisi, özellikle yazılım geliştirme süreçlerinde en temel yetkinliklerden biri olarak görülmektedir. Selby ve Woollard (2013), otomasyonun öğrencilerin kod tekrarlarını fark etmesini ve daha verimli kod yazmalarını sağladığını belirtmektedir. Bocconi vd. (2016) ise, bu becerinin öğrencilerin hem bilişsel hem de üretimsel süreçlerde hız kazandığını ve daha az hata yaptığını ortaya koymuştur. Ayrıca otomasyon, öğrencinin bilgiyi yapılandırma, yeniden kullanma ve genelleyebilme kapasitesiyle doğrudan ilişkilidir. Bu araştırmada kullanılan yapay zekâ destekli modül, öğrencilerin yazdıkları kodlara ilişkin anlık öneriler ve optimizasyon tavsiyeleri sunarak, onları sadece sonuca değil, daha iyi ve etkili çözüme yönlendirmiştir. Öğrenciler bu önerileri değerlendirerek tekrar eden yapıları otomatikleştirme konusunda farkındalık kazanmışlardır. Öğrencilerin otomasyon becerilerindeki bu gelişim, onların sadece kod yazmakla kalmayıp kodun niteliğini, verimliliğini ve sürdürülebilirliğini de düşünmeye başladıklarını göstermektedir. Bu durum, bilgi işlemsel düşünmenin sadece teknik değil, analitik ve stratejik düşünmeye dayalı bir süreç olduğunu kanıtlamaktadır. Ayrıca, öğrencilere bu yaşta otomasyon farkındalığı kazandırmak, onları yalnızca yazılım dünyasına değil, her alanda daha sistemli düşünen bireyler hâline getirme potansiyeline sahiptir.

Bebras Bilge Kunduz Testi Sonuçlarına İlişkin ve Tartışma

Araştırmada kullanılan ölçme araçlarından biri olan Bebras Bilge Kunduz Testi, öğrencilerin bilgi işlemsel düşünme becerilerini özellikle mantıksal akıl yürütme, örüntü tanıma, algoritmik yaklaşım ve problem çözme gibi alanlarda ölçmeye yönelik yapılandırılmıştır. Test, üç ayrı zorluk düzeyinde (kolay, orta, zor) çoktan seçmeli sorular içermekte ve standart okul müfredatının dışında, genel zihinsel esneklik ve algoritmik düşünmeye dayalı bilişsel süreçleri değerlendirmektedir. Araştırma bulgularına göre, deney grubunun testin genel toplamında ön test ortalaması son teste göre yükselmiş ve bu fark

istatistiksel olarak anlamlı bulunmuştur. Ancak alt boyut düzeyinde incelendiğinde; kolay ve zor düzey sorularda fark istatistiksel olarak anlamlılık düzeyine ulaşmamış, yalnızca orta düzey sorularda anlamlı fark elde edilmiştir. Kontrol grubunda ise genel bir artış gözlemlenmekle birlikte, özellikle orta ve zor düzey sorularda anlamlı fark oluşmamıştır. Deney ve kontrol gruplarının son test puanları karşılaştırıldığında, her üç zorluk düzeyinde de deney grubunun lehine bir fark görünmekle birlikte bu farklar istatistiksel olarak anlamlı bulunmamıştır. Bu sonuçlar, Bebras Testi'nin doğrudan uygulama sürecinde öğretilen becerilerle bire bir eşleşmemesi ve öğrencilerin bu test türüne aşinalık düzeyinin düşük olmasıyla açıklanabilir. Testin içeriği, öğrencilerin alışık olduğu okul sınavlarından farklı olarak daha soyut ve zihinsel akıl yürütmeye dayalı sorular içerdiğinden, öğrencilerin bu yapıya uyum sağlamaları sınırlı kalmış olabilir. Ayrıca öğrencilerin sosyoekonomik düzeylerinin düşük olması, teknoloji okuryazarlığı, bilgisayarlı düşünme ile erken tanışma ve dış kaynaklara erişim gibi konularda yetersizlik yaşadıklarını düşündürmektedir. Bu durum, test performansını dolaylı olarak etkilemiş olabilir. Literatürde de benzer şekilde, Bebras testlerinin öğrencilerin doğrudan öğretimle değil, bilişsel altyapı, alışkanlık ve problem türlerine maruz kalma sıklığıyla daha fazla ilişkili olduğunu gösteren çalışmalar bulunmaktadır. Dagienė ve Sentance (2016), Bebras testlerinin bilgi işlemsel düşünmenin geniş bir yelpazede ölçülmesini sağladığını, ancak bu sorulara alışkın olmayan öğrencilerin performansının öğretimden bağımsız olarak düşük kalabileceğini belirtmiştir. Kultusministerkonferenz (2018) raporuna göre ise, Bebras sorularında başarıyı artırmak için öğrencilerin bu tür sorulara düzenli olarak maruz kalmaları, düşünme stratejileri üzerinde çalışmaları ve bol uygulama yapmaları gerekmektedir. Bebras soruları öğrenciler için farklı bir zihinsel düzlemde kodlama bilgisi talep etmektedir. Bu çalışmada kullanılan yapay zeka destekli öğrenme modülü, öğrencilerin algoritma yazma ve yapı kurma becerilerini desteklemiş olsa da, testin çoktan seçmeli, görsel-analitik mantık içeren yapısına doğrudan karşılık gelen stratejiler eğitim sürecinde sınırlı kalmış olabilir. Ayrıca öğrencilerin günlük yaşamda çok adımlı akıl yürütme, örüntü tanıma ve soyut modelleme gerektiren deneyimlerden uzak olması, başarı düzeyini sınırlamış olabilir. Bununla birlikte, Bebras testinde gözlenen sınırlı gelişim, uygulamanın etkisizliğinden ziyade testin yapısal farklılıkları ve öğrencilerin hazırbulunuşluk düzeyleriyle daha yakından ilişkilidir. Bu nedenle, ileriki uygulamalarda Bebras benzeri soru yapılarının sınıf içi etkinliklerle daha fazla desteklenmesi, öğrencilerin bu tür düşünme biçimlerine aşinalık kazanması açısından yararlı olacaktır.

Öğrencilerinin Programlamaya Yönelik Tutumlarına İlişkin Sonuçları ve Tartışma

Araştırma kapsamında elde edilen veriler, deney grubundaki öğrencilerin programlamaya yönelik istek düzeylerinde anlamlı ve güçlü bir artış olduğunu ortaya koymuştur. Nicel bulgulara göre, deney grubunun ön test puan ortalaması yükselmiş; bu fark istatistiksel olarak anlamlı bulunmuştur. Hesaplanan Cohen's d değeri, uygulamanın yüksek düzeyde etki büyüklüğüne sahip olduğunu göstermektedir. Kontrol grubunda da anlamlı bir artış tespit edilmiştir, ancak etki büyüklüğü çok düşüktür. Bu durum, üretken yapay zekâ destekli modülün özellikle deney grubundaki öğrencilerin programlamaya yönelik istekliliğini artırmada etkili olduğunu göstermektedir. Nitel veriler de bu bulgularla uyumludur. Öğrencilerle yapılan görüşmelerde, derslere daha hevesli geldikleri, kodlama yapmayı bekledikleri ve programlamayla ilgilenmeye başladıkları görülmüştür. Özellikle yapay zekâ destekli araçların süreci eğlenceli hâle getirmesi, başarı hissini artırması ve öğrenciye doğrudan katkı sağlaması, motivasyon üzerinde önemli rol oynamıştır. Bazı öğrenciler derse katılım isteğinin arttığını, bazıları ise ders dışı zamanlarda bile kodlama etkinlikleri yapmak istediklerini belirtmiştir. Programlamaya yönelik istek, öğrencinin öğrenme sürecine aktif katılımı açısından oldukça kritik bir değişkendir. Papert (1980), programlamanın öğrencilerde yaparak öğrenmeyi teşvik ettiğini ve bu durumun içsel motivasyonu artırdığını savunur. Korkmaz ve Altun (2014) da benzer şekilde, öğrencilerin programlamaya yönelik isteklerinin arttıkça, derse katılım düzeylerinin ve kalıcı öğrenmenin de arttığını ortaya koymuştur. Programlama ortamlarının etkileşimli ve geri bildirim odaklı olması, öğrencilerde motivasyonu tetikleyen temel faktörlerden biri olarak değerlendirilir. Bu çalışmada uygulanan yapay zekâ destekli modül, öğrencilerin sadece teknik yeterliliklerini değil, derse olan bağlılıklarını ve ilgi düzeylerini de desteklemiştir. Yapay zekâdan alınan öneriler, anında geri bildirimler ve başarılı kod çalışmaları, öğrencilerin “başarabildiklerini görmelerine” katkı sunmuş ve bu da istekliliklerini doğrudan etkilemiştir. İstek düzeyindeki bu gelişim yalnızca derse olan ilgiyi değil, öğrencinin kendini gerçekleştirme sürecini de desteklemiştir. Öğrenciler, öğrenme sürecine sadece “öğrenci” olarak değil, aynı zamanda bir “üretici” ya da “tasarımcı” olarak dâhil olduklarında, bu öğrenme isteği içselleşmiş bir hâl almıştır. Bu yönüyle üretken yapay zekâ desteği, sadece öğretimsel değil, duyuşsal gelişim açısından da önemli bir rol üstlenmiştir.

Deney grubundaki öğrencilerin programlamaya yönelik inanç düzeylerinde anlamlı bir artış olduğunu göstermektedir. Deney grubunun ön test puan ortalaması son testte yükselmiş, fark istatistiksel olarak anlamlı bulunmuştur. Ancak bu artışın Cohen's d değeriyle düşük-orta düzeyde bir etkiye sahip olduğu görülmektedir. Kontrol grubunda da anlamlı bir artış gözlemlenmiştir, ancak etki büyüklüğü oldukça düşük düzeydedir. Bu bulgular, her iki grupta

da öğrencilerin programlamaya yönelik inançlarının geliştiğini, fakat bu gelişimin özellikle deney grubunda daha belirgin olduğunu ortaya koymaktadır. Nitel veriler, öğrencilerin bu inanç gelişimini kendi ifadeleriyle de yansıttığını göstermektedir. Öğrenciler, kodlama yapabileceklerine dair özgüvenlerinin arttığını, başlangıçta “yapamam” düşüncesine sahipken uygulama süreci sonunda “yapabiliyorum” hissini kazandıklarını belirtmişlerdir. Yapay zekâ desteği ile karşılaştıkları zorluklarda yönlendirilmiş olmaları, başarısızlık korkularını azaltmış ve kendi çabalarıyla ilerleyebileceklerine olan inançlarını güçlendirmiştir. Bu bulgu, literatürde yer alan birçok çalışmayla da paralellik göstermektedir. Örneğin Bandura’nın (1997) öz-yeterlik kuramına göre bireylerin belirli bir alanda başarılı olacaklarına dair inançları, o alandaki davranışlarını, azimlerini ve öğrenmeye karşı tutumlarını doğrudan etkiler. Tükel (2019) ise öğrencilerin programlamaya yönelik inançlarının, başarının temel belirleyicilerinden biri olduğunu ve olumlu inanç geliştiren öğrencilerin hata yapmaktan daha az korktuklarını, daha cesur stratejiler denediklerini belirtmektedir. Araştırma kapsamında geliştirilen üretken yapay zekâ destekli programlama modülü, öğrencilere düzenli geri bildirim sağlaması ve hatalarını öğretici biçimde analiz etmeleri yönünde destek vermesi sayesinde, öğrencilerin programlamaya dair olumlu inanç geliştirmelerinde etkili olmuştur. Öğrencilerin hata yaptıklarında sistemi suçlamak yerine süreci analiz etmeleri ve hatadan öğrenme eğilimi göstermeleri bu gelişimi desteklemektedir. Öğrencilerdeki inanç artışı, sadece teknik beceri kazancıyla değil, aynı zamanda kendi öğrenme süreçlerine güvenmeleriyle ilgilidir. Bu, onların bağımsız öğrenen bireyler olma yolunda attıkları önemli bir adımdır. Eğitim teknolojilerinde amaç yalnızca bilgi aktarmak değil, bireyin öğrenme kapasitesine olan inancını da güçlendirmektir. Bu çalışmanın bu yönde önemli bir etki yarattığı söylenebilir.

Deney grubundaki öğrencilerin programlamaya yönelik ilgilerinde anlamlı ve dikkat çekici bir artış olduğunu göstermektedir. Deney grubunun ön test puan ortalaması son testte yükselmiş ve bu fark istatistiksel olarak anlamlı bulunmuştur. Hesaplanan Cohen’s d değeri, uygulamanın bu alt boyutta çok büyük düzeyde bir etki yarattığını ortaya koymaktadır. Kontrol grubunda da anlamlı bir artış gözlenmiş, ancak etki büyüklüğü orta düzeydedir. Bu sonuçlar, uygulama sürecinde deney grubundaki öğrencilerin programlamaya olan ilgisinin çok daha fazla geliştiğini göstermektedir. Nitel veriler, bu gelişimi destekleyici yönde açıklamalar sunmaktadır. Öğrenciler, kodlama sürecinden keyif aldıklarını, boş zamanlarında da araştırma yaptıklarını, farklı programlama dillerine ve yapay zekâyâ yönelik meraklarının oluştuğunu ifade etmişlerdir. Ayrıca, bazı öğrencilerin ders dışı zamanlarda yazılım kulüplerine katılmak istediklerini ya da oyun tasarımı gibi alanlara yöneldiklerini belirtmeleri, bu ilgi artışının sadece derse özgü değil, bireysel öğrenme motivasyonuna da yansıdığını göstermektedir. Bu bulgu, literatürde programlamaya yönelik ilginin, öğrencilerin derse katılım, akademik başarı ve

mesleki yönelimlerinde belirleyici rol oynadığını vurgulayan çalışmalarla örtüşmektedir. Bennedsen ve Caspersen (2007), öğrencilerin bilgisayar bilimlerine yönelik ilgisinin düşük olmasının öğrenme motivasyonunu olumsuz etkilediğini belirtirken; Yukselturk ve Altıok (2017), programlamaya yönelik ilgi düzeyi yüksek olan öğrencilerin daha kalıcı öğrenmeler gerçekleştirdiğini göstermiştir. Özellikle etkileşimli, uygulamalı ve destekleyici öğrenme ortamları, öğrencilerin bu alana olan ilgisini pekiştirmektedir.

Araştırma sonuçları, öğrencilerin programlamaya yönelik tutumlarında olumlu bir değişim yaşandığını göstermektedir. Özellikle üretken yapay zekâ destekli dönütlerin, öğrencilerin öğrenme sürecinde kendilerini daha yeterli hissetmelerini sağladığı gözlemlenmiştir. Ogunleye vd. (2024) yürüttüğü sistematik derleme çalışmasında, üretken yapay zekânın bireysel öğrenmeyi desteklemesinin öğrencilerin öz güvenini artırdığı ve tutumlarını pozitif yönde etkilediği ifade edilmiştir. Aynı şekilde, Papadakis (2023) de üretken yapay zekâ destekli modüllerin öğrencilerin programlama sürecine daha aktif katılımını sağladığını vurgulamaktadır.

Bu çalışmada kullanılan ÜYZ destekli öğrenme ortamı, öğrencilerin klasik kod yazma süreçlerinden farklı olarak daha yaratıcı, etkileşimli ve özgün üretimler yapmalarına olanak sağlamıştır. Öğrenciler sadece belirli görevleri tamamlamakla kalmamış, aynı zamanda kendi kodlarını geliştirme, fikir üretme ve keşfetme sürecine de aktif şekilde dâhil olmuşlardır. Bu durum, onların öğrenmeyi sadece zorunluluk değil, merak ve keşif süreci olarak görmeye başlamalarını sağlamıştır. Öğrencilerin programlamaya duyduğu ilginin bu denli artması, onların sadece bir konuyu öğrenmekle kalmayıp kendilerini bu alanda görmek istemeleri ile doğrudan ilişkilidir. Bu ilgi, ilerleyen dönemlerde yazılım alanına yönelme, STEM kariyer planlaması yapma ve bireysel projeler geliştirme gibi birçok olumlu çıktının öncüsü olabilir. Eğitimde sürdürülebilir bir öğrenme ortamı oluşturmak istiyorsak, ilgiyi sadece çekmekle kalmamalı, beslemeli ve dönüştürmeliyiz. Bu çalışmanın bunu başarabildiği söylenebilir.

Programlama Başarı Testine İlişkin Sonuç ve Tartışma

Araştırmada kullanılan Programlama Başarı Testi, öğrencilerin algoritma oluşturma, hata ayıklama ve problem çözme becerilerini ölçmeye yönelik olarak yapılandırılmıştır. Çoktan seçmeli ve uygulamaya dayalı açık uçlu sorulardan oluşan bu test, öğrencilerin yalnızca kuramsal bilgi düzeylerini değil, aynı zamanda gerçek bir problem durumunda uygulama ve çözüm üretme becerilerini de değerlendirmeyi amaçlamaktadır. Araştırma bulguları, deney grubundaki öğrencilerin programlama başarısında anlamlı ve güçlü bir artış gösterdiğini ortaya koymuştur. Deney grubunun son test puan ortalaması $\bar{X} = 82.94$, kontrol grubunun ise $\bar{X} =$

71.47 olarak hesaplanmıştır. Bu fark istatistiksel olarak anlamlı bulunmuştur. Hesaplanan Cohen's d değeri, programlama başarı düzeyinde yüksek düzeyde bir etki büyüklüğüne işaret etmektedir. Bu bulgu, uygulanan yapay zekâ destekli öğrenme modülünün öğrencilerin kodlama becerilerine doğrudan katkı sunduğunu ortaya koymaktadır. Nitel veriler de bu durumu desteklemektedir. Öğrenciler, uygulama sürecinde kodlama sorularını çözme becerilerinin geliştiğini, hata bulma konusunda daha dikkatli ve sistemli çalıştıklarını belirtmişlerdir. Ayrıca, ilk başta açık uçlu soruları çözerken zorlandıklarını fakat zamanla bu sorulara karşı yaklaşım biçimlerinin değiştiğini ifade etmişlerdir. Bu tür açık uçlu ve uygulama tabanlı değerlendirme türlerinin, öğrencilerin düşünme derinliğini artırdığı ve kalıcı öğrenmeye katkı sunduğu anlaşılmaktadır. Bu sonuçlar, literatürde de benzer şekilde ifade edilen görüşlerle örtüşmektedir. Grover ve Pea (2013), programlama başarısının yalnızca sentaks bilgisiyle değil, problem çözme sürecini planlama, adım adım düşünme ve kodların çalışmasını analiz edebilme becerisiyle ilişkili olduğunu vurgular. Falkner ve Vivian (2015), proje tabanlı ve açık uçlu sorularla yapılan programlama öğretiminin öğrencilerin gerçek dünya problem çözme becerilerini daha iyi geliştirdiğini belirtmektedir. Özellikle yapay zekâ desteğiyle verilen anında geri bildirimler, öğrencilerin hatalarını fark etme ve düzeltme sürecini hızlandırmakta, böylece bilişsel yükü azaltarak öğrenmeyi derinleştirmektedir. Bu araştırmada uygulanan üretken yapay zekâ destekli modül, öğrencilerin kod yazma sürecinde karşılaştıkları sorunları analiz etme, çözüm yollarını karşılaştırma ve en uygun yapıyı seçme gibi yüksek düzey beceriler geliştirmelerini sağlamıştır. Öğrenciler yalnızca “doğru cevabı” değil, aynı zamanda “neden doğru olduğunu” da düşünmeye başlamışlardır. Programlama başarısındaki bu gelişim, öğrencilerin yalnızca teknik bilgi kazanımı değil, aynı zamanda bağımsız problem çözücü bireyler olma yolunda ilerlediklerini göstermektedir. Bu süreçte öğrenciler, yapay zekâyı pasif bir yardımcı olarak değil, öğrenme sürecine aktif katkı sunan bir “akıllı öğrenme partneri” olarak görmeye başlamışlardır. Bu da geleneksel kodlama öğretiminin ötesinde bir öğrenme deneyimi sunulmasını sağlamıştır.

Programlama başarı testi sonuçları, öğrencilerin uygulama sonrasında başarılarında anlamlı bir artış olduğunu ortaya koymuştur. Bu durum, üretken yapay zekânın öğrencilerin yalnızca mevcut bilgi düzeyini değil, aynı zamanda öğrenme süreçlerindeki ilerlemelerini de desteklediğini göstermektedir. Waseem vd. (2024) yaptığı araştırmada, ChatGPT gibi üretken yapay zekâ araçlarının, yazılım geliştirme sürecindeki performansı artırdığı ve öğrencilerin projelerinde daha az hata yapmalarını sağladığı belirtilmiştir. Bu bulgular, araştırmamızda gözlemlenen öğrenme çıktıları ile tutarlıdır.

Genel Sonuç ve Değerlendirme

Bu araştırmada geliştirilen ÜYZ destekli programlama öğrenme modülü, ortaokul düzeyindeki öğrencilerin hem bilişsel hem de duyuşsal gelişimlerini desteklemeyi hedefleyen yenilikçi bir öğretim süreci sunmuştur. Uygulama sonucunda elde edilen bulgular, modülün öğrenciler üzerinde çok boyutlu ve bütüncül etkiler yarattığını göstermektedir. Öncelikle, öğrencilerin bilgi işlemsel düşünme becerilerinde anlamlı bir gelişme gözlemlenmiştir. Özellikle algoritmik düşünme, soyutlama, otomasyon gibi alt boyutlarda yüksek etki büyüklükleriyle elde edilen gelişmeler; öğrencilerin yalnızca kod yazmakla kalmadığını, aynı zamanda sistemli düşünme, problem çözme, akıl yürütme ve analiz etme becerilerini de kazandıklarını ortaya koymuştur. Bu gelişim, öğrencilerin programlama yoluyla üst düzey düşünme becerilerini içselleştirdiğine işaret etmektedir. Programlamaya yönelik tutum açısından, öğrencilerin derse olan isteklerinin, kendilerine olan inançlarının ve programlamaya duydukları ilginin olumlu yönde değiştiği belirlenmiştir. Öğrenciler, yapay zekâ desteğiyle daha fazla başarı hissi yaşamış, hata yapma korkusu azalmış, ders dışı etkinliklere yönelme eğilimleri artmıştır. Bu durum, öğrencilerin öğrenmeye karşı içsel motivasyonlarının güçlendiğini, programlamayı sadece okul içi bir zorunluluk değil, kişisel gelişim alanı olarak benimsediklerini göstermektedir. Programlama başarısı açısından elde edilen bulgular da dikkat çekicidir. Açık uçlu sorulardan oluşan başarı testi, öğrencilerin düşünme derinliğini, mantıksal tutarlılığını ve algoritmik uygulama becerisini ölçmüştür; deney grubundaki öğrencilerin kontrol grubuna göre anlamlı düzeyde daha yüksek başarı gösterdiği görülmüştür. Bu, öğretim sürecinin öğrencilerde gerçek anlamda uygulama yetkinliği geliştirdiğini, yalnızca teorik bilgi değil, pratik beceri kazandırdığını göstermektedir. Buna karşın, Bebras Bilge Kunduz Testi sonuçları sınırlı düzeyde değişim göstermiştir. Bu durum, testin yapısal özelliklerinin uygulamada kazandırılan becerilerle bire bir örtüşmemesi, öğrencilerin bu tür sorulara alışık olmamaları ve düşük sosyoekonomik koşullar nedeniyle alternatif düşünme stratejilerine yeterince maruz kalmamalarıyla açıklanabilir. Ancak bu bulgu, çalışmanın değerini azaltmaktan çok, eğitsel müdahalelerle ölçme araçları arasındaki uyumun ne kadar önemli olduğunu hatırlatmaktadır. Genel olarak değerlendirildiğinde, bu araştırma üretken yapay zekânın öğretim sürecinde yalnızca teknik bir araç değil, aynı zamanda öğrenmeyi kolaylaştıran, özelleştiren ve kişiselleştiren bir öğrenme destekçisi olarak etkili biçimde kullanılabileceğini göstermektedir. Öğrenciler bu süreçte yalnızca bilgi tüketicisi değil, aynı zamanda bilgi üreticisi olmuş; hem düşünsel hem de duyuşsal düzeyde güçlenmişlerdir. Araştırmanın en güçlü yönlerinden biri, ölçme ve değerlendirme sürecinde hem nicel hem de nitel verilerle çok boyutlu bir analiz sunmasıdır. Bu yaklaşım, öğrencilerin gelişimini sadece sayısal verilerle değil, gerçek ifadeleri ve öğrenme deneyimleri üzerinden anlamaya olanak

sağlamıştır. Ayrıca geliştirilen öğretim modülünün ortaokul düzeyine uygunluğu, sahaya özgü ve uygulanabilir bir örnek sunması açısından önemli bir katkı niteliğindedir. Bu çalışma, öğrencilere yalnızca kodlama öğretmek değil; düşünmeyi, üretmeyi ve öğrenmeyi öğretmek için yapay zekâ gibi güçlü teknolojilerin pedagojik çerçevede nasıl dönüştürülebileceğine dair değerli bir örnek sunmaktadır.

Genel olarak değerlendirildiğinde, üretken yapay zekâ destekli modülün, ortaokul öğrencileri üzerinde hem bilişsel hem de duyuşsal düzeyde anlamlı etkiler yarattığı söylenebilir. ChatGPT, Gemini ve Copilot gibi araçların öğrenme sürecinde sunduğu anlık geri bildirim, hata analizi ve örnek çözümler; öğrencilerin hem başarılarını artırmakta hem de motivasyonlarını yükseltmektedir (Brown vd., 2020; DeepMind, 2023; GitHub, 2021). Ayrıca, bu araçlar öğrencilerin bireysel öğrenme hızlarına uyum sağlayarak öğretim sürecini daha kapsayıcı ve erişilebilir hale getirmektedir.

Öneriler

Araştırma bulguları ve sonuçlar göz önüne alındığında etkinliklere ve gelecek araştırmalara yönelik olan öneriler aşağıda verilmiştir.

Uygulayıcılara (Eğitimcilere) Yönelik Öneriler

- Üretken yapay zekâ destekli programlama modülleri, ortaokul düzeyinde öğrencilerin hem bilgi işlemsel düşünme becerilerini hem de programlamaya yönelik tutumlarını geliştirmede etkili olabilmektedir. Bu nedenle, ChatGPT gibi dil tabanlı yapay zekâ araçlarının entegrasyonu, rehberlik eden dijital asistan olarak aktif biçimde yapılandırılmalı, yönlendirici sorular, örnekler ve açıklayıcı dönütlerle desteklenebilir.
- Uygulama sürecinde, ChatGPT'nin zaman zaman karmaşık veya bağlama uygun olmayan yanıtlar ürettiği görülmüştür. Bu durumun önüne geçebilmek için, sistem içine gömülü, sınıf düzeyine uygun ön tanımlı komut setleri ve cevap sınırlandırmaları yapılması önerilmektedir. Aynı zamanda öğretmenlerin bu sistemin çıktıları üzerinde denetim sağlayabileceği ara yüzlerin geliştirilmesi önemlidir.
- Kullanılan modül web tabanlı olarak tasarlanmış olsa da, mobil uyumlu versiyonlarının geliştirilmesi erişilebilirliği artıracaktır. Öğrencilerin ev ortamında da kullanabileceği bu sistem, ders dışı öğrenme süreçlerini destekleyecek şekilde asistan modunda çalışabilir.
- Sisteme, öğrencilerin ilgi ve dikkatini artırmak amacıyla etkileşimli anlatım videoları, sürükle-bırak blok tabanlı örnekler ve oyunlaştırılmış görev yapıları entegre edilebilir.

Bu tür görsel-işitsel öğeler, özellikle algoritmik düşünme ve soyutlama becerilerinin gelişimini kolaylaştırabilir.

- Modülün güçlü yönlerinden biri, öğrencilerin adım adım öğrenmesini destekleyen yönlendirmeli yapı ve yapay zekâdan alınan anlık geri bildirimler olmuştur. Ancak, bazı öğrenciler karmaşık komutları anlamlandırmakta zorlanmıştır. Bu nedenle, farklı zorluk düzeylerine sahip görevlerin sunulduğu kademeli yapılar modüle entegre edilebilir.
- Bebras testi gibi farklı düşünme yapıları gerektiren uygulamalara yönelik, öğrencilerin daha fazla maruz kalmasını sağlayacak YZ destekli mantık yürütme, örüntü fark etme ve problem çözme senaryoları modüle entegre edilebilir. Böylece Bebras testinin yapısına daha yakın beceriler doğrudan modül içinde geliştirilebilir.
- Modül, farklı sınıf düzeylerine ve kazanım hedeflerine göre uyarlanabilir biçimde genişletilmelidir. Özellikle 5. ve 6. sınıf gibi alt düzeylere blok tabanlı sadeleştirilmiş versiyonlar, üst düzeylere ise metin tabanlı Python uygulamaları entegre edilerek kullanılabilir.
- Ders dışı öğrenme süreçlerini desteklemek adına modül, öğrencilerin kendi hızlarında çalışabilecekleri, yapay zekâdan destek alabilecekleri bir "kişisel öğrenme asistanı" olarak yapılandırılabilir. Bu sistemde öğrenciye günlük küçük görevler atanabilir, ilerleme takibi yapılabilir ve öneriler sunulabilir.

Gelecek Araştırmalara Yönelik Öneriler

- Benzer modüllerin farklı yaş gruplarında (ilkokul, lise) uygulanmasıyla yaşa bağlı etki düzeyleri karşılaştırmalı olarak incelenebilir.
- Bebras gibi bilişsel derinlik gerektiren test türlerinin etkilerini daha net ortaya koyabilmek için uzun süreli uygulamalar veya aşamalı öğretim modelleri tercih edilebilir.
- Üretken yapay zekâ destekli öğrenme ortamlarının öğrencilerin üstbilişsel farkındalıkları, öğrenme stratejileri ve öz düzenleme becerileri üzerindeki etkisi nitel yöntemlerle daha derinlemesine incelenebilir.

Geliştiricilere ve Sistem Tasarımcılarına Yönelik Öneriler

- Geliştirilen modülün farklı düzeydeki öğrencilere (başlangıç, orta, ileri) uygun uyarlanabilir içerikler sunması sağlanmalıdır. Yapay zekâ, öğrencinin öğrenme hızına ve başarımlarına göre içerik zorluk düzeyini dinamik olarak ayarlayacak şekilde yapılandırılabilir.

- Modül içerisine öğrencilerin kendi kodlarını test edebileceği, küçük projeler üretebileceği, görsel simülasyonlar ve animasyonlarla çıktıları gözlemleyebileceği daha etkileşimli alanlar entegre edilmelidir. Bu sayede öğrenme hem somutlaştırılır hem de kalıcı hâle gelir.
- Sistem, sadece bireysel öğrenmeye değil, işbirlikli öğrenme süreçlerine de olanak tanımalıdır. Öğrencilerin aynı ortamda birlikte kod yazabilecekleri, tartışabilecekleri veya bir görev üzerinde birlikte çalışabilecekleri bir paylaşım ve işbirliği alanı geliştirilebilir.
- Modül, yalnızca kodlama temelli değil, öğrencilerin algoritmik düşünme becerilerini günlük yaşam problemleriyle ilişkilendirebileceği açık uçlu etkinliklerle de desteklenebilir.



KAYNAKÇA

- Ahmadzadeh, M., Elliman, D., & Higgins, C. (2007). An analysis of patterns of debugging among novice computer science students. *ACM SIGCSE Bulletin*, 39(3), 82–86.
- Akdeniz, M., & Özdiñ, F. (2021). Eğitimde yapay zeka konusunda Türkiye adresli çalışmaların incelenmesi. *YYÜ Eğitim Fakültesi Dergisi (YYU Journal of Education Faculty)*, 18(1), 912–932. <https://doi.org/10.33711/yyuefd.938734>
- Alanoğlu, M., & Karabatak, S. (2020). Eğitimde yapay zekâ. In F. Güçlü Yılmaz & M. Naillioğlu Kaymak (Eds.), *Eğitim araştırmaları 2020* (pp. 175–196). Eğitim Yöneticileri ve Uzmanları Derneği Yayınları. E-ISBN: 978-975-2490-47-5.
- Alsancak Sırakaya, D. (2019). Programlama öğretiminin bilgi işlemsel düşünme becerisine etkisi. *Türkiye Sosyal Araştırmalar Dergisi*, 23(2), 575–590.
- Altay, G., & Kışla, T. (2018). Programlamaya yönelik tutum ölçeği ve psikometrik özellikleri. *Ege Eğitim Dergisi*, 19(2), 559–574.
- Altın, R. (2021). *Secondary school students' programming and computational thinking skills: Traditional and interdisciplinary approaches to teaching programming* [Yayımlanmamış doktora tezi]. Orta Doğu Teknik Üniversitesi.
- Altun, E. (2024). Yapay zekâ ve pedagoji: Eğitimde fırsatlar ve zorluklar. *Dijital Teknolojiler ve Eğitim Dergisi*, 3(1), 80–95. <https://doi.org/10.5281/zenodo.12637335>
- Anagnostopoulos, C.-N. (2023). *ChatGPT impacts in programming education: A short list of input questions and output answers from ChatGTP*. University of the Aegean, Cultural Technology and Communication Department, Intelligent Systems Lab.
- Apostolellis, P., Stewart, M., Frisina, C., & Kafura, D. (2014). Bringing computational thinking to K–12: What is involved and what is needed. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*. ACM.
- Arabacıoğlu, T., Bülbül, H. İ., & Filiz, A. (2007). Bilgisayar programlama öğretiminde yeni bir yaklaşım. *IX. Akademik Bilişim Konferansı Bildirileri* (ss. 193–197). Dumlupınar Üniversitesi.
- Araujo, J., Andrade, W., Guerrero, M., & Melo, F. (2019). Computational thinking in K-12 education: A review. *Educational Technology Research and Development*, 67(3), 711–730.
- Armağan, N., Efe, R., Güzel, S., & Uluişik, M. (2024). The use and development of artificial intelligence in education. *Eurasian Education & Literature Journal*, 19(4), 59–70. <https://doi.org/10.17740/eas.edu.2024-V19-04>
- Arslan, K. (2020). Eğitimde yapay zeka ve uygulamaları. *Batı Anadolu Eğitim Bilimleri Dergisi*, 11(1), 71–88.
- Arslan, K. (2023). Eğitimde yapay zeka ve uygulamaları. *Türk Eğitim Dergisi*, 10(2), 45–68.
- Ateş, Ş. (2021). Oyunlaştırılmış Python programlama dili öğretiminin öğrencilerin akademik başarılarına, öğrenmenin kalıcılığına ve programlamaya yönelik tutumlarına etkisi (Tez No. 697071) [Yüksek lisans tezi, Bahçeşehir Üniversitesi-İstanbul]. Yükseköğretim Kurulu Ulusal Tez Merkezi.
- Atun, H. (2024). Üretken yapay zeka destekli programlama eğitiminde sorgu mühendisliğinin çeşitli değişkenler açısından değerlendirilmesi (Tez No. 883715) [Doktora tezi, Ankara Üniversitesi-Ankara]. Yükseköğretim Kurulu Ulusal Tez Merkezi.

- Avcı, E. (2024). Akıllı şehirler için üretken yapay zeka kavramsal çerçevesi. *Kent Akademisi Dergisi*, 17(5), 1654–1675. <https://doi.org/10.35674/kent.1490925>
- Aydın, C. (2021). Yapay zeka destekli eğitim sistemlerinin öğrenci performansına etkisi. *Eğitim Teknolojileri Dergisi*, 15(2), 45–59.
- Bağra, A., & Kılınç, H. H. (2021). Ortaokul öğrencilerinin kodlama eğitimine ilişkin görüşleri. *Maarif Mektepleri Uluslararası Sosyal ve Beşerî Bilimler Dergisi*, 4(1), 36–51. <https://doi.org/10.47155/mamusbbd.946241>
- Baker, R. S., & Inventado, P. S. (2014). Educational data mining and learning analytics. In J. A. Larusson & B. White (Eds.), *Learning analytics: From research to practice* (pp. 206–210). Springer.
- Baki, A., & Gökçek, T. (2012). Karma araştırma yöntemleri üzerine bir inceleme. *Eğitim ve Bilim*, 37(164), 1–16.
- Balanskat, A., & Engelhardt, K. (2015). *Computing our future: Computer programming and coding priorities, school curricula and initiatives across Europe*. European Schoolnet.
- Bandura, A. (1997). *Self-efficacy: The exercise of control*. W. H. Freeman.
- Batal, M. S. (2019). *Kodlama (programlama) öğretimi*. Uluslararası Sunhill Üniversitesi.
- Benzer, A. İ., & Erümit, A. K. (2017). Programlama öğretimine yönelik lisansüstü tezlerin incelenmesi. *Journal of Instructional Technologies & Teacher Education*, 6(3), 99–110.
- Best, J. W., & Kahn, J. V. (2017). *Research in education*. Pearson Education.
- Binns, R. (2018). Fairness in machine learning: Lessons from political philosophy. In *Proceedings of the 2018 Conference on Fairness, Accountability, and Transparency* (pp.149–159). Association for Computing Machinery. <https://doi.org/10.1145/3287560.3287598>
- Bozkurt, A. (2023). ChatGPT, üretken yapay zekâ ve algoritmik paradigma değişikliği. *Alanyazın: Eğitim Bilimleri Eleştirel İnceleme Dergisi (CRES Journal)*, 4(1), 63–72. <https://doi.org/10.59320/alanyazin.1283282>
- Börü, N. (2019). Eğitimde sosyal adalet için yerel yönetimler. *14. Uluslararası Eğitim Yönetimi Kongresi Tam Metin Bildiri Kitabı* (ss. 752–760). Orta Doğu Teknik Üniversitesi. <https://doi.org/978-605-037-012-6>
- Brennan, K., & Resnick, M. (2012, April). *New frameworks for studying and assessing the development of computational thinking*. Paper presented at the Annual Meeting of the American Educational Research Association (AERA), Vancouver, BC, Canada.
- Bringula, R. (2023). ChatGPT in a programming course: Benefits and limitations. *Frontiers in Education*, 8(1), 125–142.
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901.
- Bucaioni, A., Ekedahl, H., Helander, V., & Nguyen, P. T. (2023). Programming with ChatGPT: How far can we go?. *Computational Intelligence Journal*, 29(4), 345–359.
- Büyüköztürk, Ş. (2007). *Deneyisel desenler: Öntest-sontest kontrol grubu deseni ve veri analizi*. Pegem A Yayıncılık.
- Cao, J., Li, M., Wen, M., & Cheung, S.-C. (2023). A study on prompt design, advantages and limitations of ChatGPT for deep learning program repair. *Software Engineering Journal*, 38(5), 220–235.

- Ceylan, V. K., & Akar Vural, R. (2023). Bilgi işlemsel düşünme becerisinin öz-değerlendirilmesi: Bir ölçek geliştirme çalışması. *Rumeli Eğitim Dergisi*, 4(Haziran), 1–22.
- Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., & Bharath, A. A. (2018). Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1), 185–192.
- Creswell, J. W. (2008). *Educational research: Planning, conducting, and evaluating quantitative and qualitative research*. Pearson Education.
- Çanakcı, Y. E. (2022). Programlama eğitime başlamada blok temelli programlama ile metin temelli programlamanın karşılaştırılması (Tez No. 762489) [Yüksek lisans tezi, Ondokuz Mayıs Üniversitesi-Samsun]. Yükseköğretim Kurulu Ulusal Tez Merkezi.
- Çavuş, M. N. (2024). Eğitimde yapay zekâ tabanlı ölçme ve değerlendirme üzerine bir derleme. *International Journal of English for Specific Purposes (JOINESP)*, 2(1), 39–54. <https://doi.org/10.3390/joinesp.1492424>
- Çetin, E. (2012). Bilgisayar programlama eğitiminin çocukların problem çözme becerileri üzerine etkisi (Tez No. 349116) [Yüksek lisans tezi, Gazi Üniversitesi-Ankara]. Yükseköğretim Kurulu Ulusal Tez Merkezi.
- Çetin, M., & Aktaş, A. (2021). Yapay zekâ ve eğitimde gelecek senaryoları. *OPUS Uluslararası Toplum Araştırmaları Dergisi*, 18(Eğitim Bilimleri Özel Sayısı), 4227–4248. <https://doi.org/10.26466/opus.911444>
- Çetin, T., & Sönmez, Ö. F. (2009). Sosyal bilgiler öğretmen adaylarının topluma hizmet uygulamaları dersinin amaç ve içeriğine yönelik görüşlerinin değerlendirilmesi. *Eğitim ve Bilim*, 34(151), 40–46.
- Çimen Çoşğun, Ü., & Çoşğun, V. (2018, Nisan). Programlama öğretiminin ortaokul öğrencilerinin öz-düzenleme stratejileri ve motivasyonel inançları üzerindeki etkisi [Sözlü bildiri]. 27. Uluslararası Eğitim Bilimleri Kongresi, Antalya, Türkiye.
- Çobanoğlu, R., & Yıldırım, A. (2021). Türkiye’de program geliştirme çalışmaları: Cumhuriyet’in ilanından günümüze tarihsel bir analiz. *Türk Eğitim Bilimleri Dergisi (TEBD)*, 19(2), 810–830. <https://doi.org/10.37217/tebd.912329>
- Dagienė, V., & Futschek, G. (2008). Bebras international contest on informatics and computer literacy: Criteria for good tasks. In *Proceedings of the International Conference on Informatics in Secondary Schools – Evolution and Perspectives (ISSEP)*. Springer.
- Dagienė, V., & Sentance, S. (2016). It’s computational thinking! Bebras tasks in the curriculum. *Informatics in Education*, 15(1), 39–52.
- Dagienė, V., & Stupurienė, G. (2016). Bebras – a sustainable community building model for the concept based learning of informatics and computational thinking. *Informatics in Education*, 15(1), 25–44.
- Demirer, V., & Sak, N. (2016). Dünyada ve Türkiye’de programlama eğitimi ve yeni yaklaşımlar. *Eğitimde Kuram ve Uygulama*, 12(3), 521–546.
- Demirer, V., & Sak, N. (2023). Programming education and new approaches around the world and in Turkey. *Turkish Journal of Computer and Mathematics Education*, 14(2), 225–241.
- Demirkaya, H., & Sarpel, E. (2018). Eğitim ve geliştirme uygulamalarında yeni nesil bilişim teknolojilerinden sanal gerçeklik, bulut bilişim ve yapay zekâ. *Karadeniz Uluslararası Bilimsel Dergi*, 40, 231–245. <https://doi.org/10.17498/kdeniz.460145>

- Denning, P. J. (2009). The profession of IT: Beyond computational thinking. *Communications of the ACM*, 52(6), 28–30.
- DiMaggio, P., & Hargittai, E. (2001). *From the "digital divide" to "digital inequality": Studying Internet use as penetration increases* (Working Paper No. 15). Princeton University, Center for Arts and Cultural Policy Studies.
- Durak, H., Karaoğlu-Yılmaz, F. G., Yılmaz, R., & Seferoğlu, S. S. (2017). Erken yaşta programlama eğitimi: Araştırmalardaki güncel eğilimlerle ilgili bir inceleme. In H. F. Odabaşı, B. Akkoyunlu, & A. İşman (Eds.), *Eğitim teknolojileri okumaları 2017* (Bölüm 12, ss. 205–236). TOJET ve Sakarya Üniversitesi.
- Ekinci, M., Saraç Öztıp, N., Demir Kaymak, Z., & Canan Güngören, Ö. (2023). The relationship between secondary school students' attitudes towards coding and computational thinking skills. *International Journal of Düzce Educational Sciences*, 1(1), 81–90.
- Elgammal, A., Liu, B., Elhoseiny, M., & Mazzone, M. (2017). CAN: Creative adversarial networks, generating "art" by learning about styles and deviating from style norms. In *Proceedings of the 8th International Conference on Computational Creativity* (pp. 248–256).
- Erdem, E. (2018). Blok tabanlı ortamlarda programlama öğretimi sürecinde farklı öğretim stratejilerinin çeşitli değişkenler açısından incelenmesi (Tez No. 509354) [Yüksek lisans tezi, Başkent Üniversitesi -Ankara]. Yükseköğretim Kurulu Ulusal Tez Merkezi.
- Eryılmaz, S., & Deniz, G. (2019). Türkiye’de programlama eğitimi ile ilgili yapılan çalışmaların incelenmesi: Bir betimsel analiz çalışması. *Eğitimde Kuram ve Uygulama*, 15(4), 319–338. <https://doi.org/10.17244/eku.645387>
- Esling, P., & Devis, N. (2020). Creativity in the era of artificial intelligence. *arXiv*. <https://arxiv.org/abs/2008.05959>
- Fayer, S., Lacey, A., & Watson, A. (2017). *STEM occupations: Past, present, and future*. U.S. Bureau of Labor Statistics.
- Fırat, M., Sakar, A. N., & Yurdakul, I. K. (2014). Açıklayıcı karma yöntem deseninin eğitim araştırmalarında kullanımı. *Eğitim ve Bilim*, 39(173), 25–36.
- Geçitli, E., & Bümen, N. (2020). Kademelendirilmiş etkinlik yoluyla metin tabanlı programlama öğretimi: İşbirlikli bir eylem araştırması. *Asya Öğretim Dergisi*, 8(2), 64–85. <https://doi.org/10.47215/aji.791967>
- Geng, C., Zhang, Y., Pientka, B., & Si, X. (2023). Can ChatGPT pass an introductory level functional language programming course? *Journal of Functional Programming Education*, 17(3), 85–102.
- Gezgin, D. M., Azaz, E., & Atabay, E. (2022). Ortaokul öğrencilerinin robotik ve kodlama eğitimi başarılarına işbirlikli öğrenme tutumu, problem çözme becerisi algısı ve kişilik tiplerinin etkisi: Bir nedensel karşılaştırma araştırması. *Journal of Instructional Technologies & Teacher Education*, 11(2), 28–42. <https://doi.org/10.51960/jitte.1165083>
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2672–2680.
- Gökçe, M. (2023). Yapay zekâ: Tanım, gelişim ve gelecek. *Akademik Bilişim Dergisi*, 12(1), 15–28.

- Gözet, M., Filiz, U., & Yılmaz, A. E. (2023). Üretken yapay zekâ. *International Journal of Multidisciplinary Studies and Innovative Technologies*, 7(1), 32–40. <https://doi.org/10.36287/ijmsit.7.1.32>
- Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, 42(1), 38–43. <https://doi.org/10.3102/0013189X12463051>
- Gülbahar, Y., Kalelioğlu, F., Doğan, D., & Karataş, E. (2020). Bilge Kunduz: Enformatik ve bilgi işlemsel düşünmeyi kavram temelli öğrenme yaklaşımı. *Ankara Üniversitesi Eğitim Bilimleri Fakültesi Dergisi*, 53(1), 241–272.
- Gülbahar, Y., Kert, S. B., & Kalelioğlu, F. (2019). Bilgi işlemsel düşünme becerisine yönelik öz yeterlik algısı ölçeği: Ortaokul öğrencileri için geçerlilik ve güvenirlik çalışması. *Eğitim Teknolojisi Kuram ve Uygulama*, 9(1), 16–30.
- Gürsoy, K., & Çekmez, E. (2019). Ortaokul öğrencilerinin programlamaya yönelik tutumlarının ve görüşlerinin incelenmesi. *Türk Bilgisayar ve Matematik Eğitimi Dergisi*, 10(3), 757–777. <https://doi.org/10.16949/turkbilmat.466047>
- Güzey, C., Çakır, O., Athar, M. H., Yurdaöz, E., & Saad, S. (2023). Eğitimde yapay zeka konusunda yapılmış çalışmaların içerik analizi. *Bilgi ve İletişim Teknolojileri Dergisi*, 5(1), 66–77. <https://doi.org/10.53694/bited.1060730>
- Heintz, F., Mannila, L., & Färnqvist, T. (2017). A review of models for introducing computational thinking, computer science and computing in K–12 education. *Frontiers in Education*, 2, 1–18. <https://doi.org/10.1109/FIE.2016.7757410>
- Holmes, W., Bialik, M., & Fadel, C. (2019). *Artificial intelligence in education: Promises and implications for teaching and learning*. Center for Curriculum Redesign.
- Humble, N., Boustedt, J., Holmgren, H., Milutinovic, G., Seipel, S., & Östberg, A.-S. (2023). Cheaters or AI-enhanced learners: Consequences of ChatGPT for programming education. *Journal of Educational Integrity*, 13(1), 33–51.
- İncemen, S., & Öztürk, G. (2024). Farklı eğitim alanlarında yapay zekâ: Uygulama örnekleri. *International Journal of Computers in Education*, 7(1), 27–49. <https://doi.org/10.5281/zenodo.12600022>
- İpek, Z., Can-Gözüm, A. İ., Papadakis, S., & Kallogiannakis, M. (2023). Educational applications of the ChatGPT AI system: A systematic review research. *Journal of Artificial Intelligence in Education*, 15(2), 92–108.
- İşler, B., & Kılıç, M. Y. (2021). Eğitimde yapay zekâ kullanımı ve gelişimi. *e-Journal of New Media*, 5(1), 1–11. https://doi.org/10.17932/IAU.EJNM.25480200.2021/ejnm_v5i1001
- İyigün, N. Ö., & Yılmaz, M. K. (2021). *Yapay zeka: Güncel yaklaşımlar ve uygulamalar* (1. baskı). Beta Basım Yayım Dağıtım.
- Jebreen, A., & Husain, A. (2023). Potentials of ChatGPT in computer programming: Insights from programming instructors. *Journal of Information Technology*, 21(3), 109–127.
- Jukiewicz, M. (2023). The future of grading programming assignments in education: The role of ChatGPT in automating the assessment and feedback process. *International Journal of Computer Science Education*, 37(2), 201–218.
- Kalelioğlu, F., & Gülbahar, Y. (2014). The effects of teaching programming via Scratch on problem solving skills: A discussion from learners' perspective. *Informatics in Education*, 13(1), 33–50.
- Karaca, F., & Erdem, M. (2020). Yapay zeka tabanlı eğitim araçlarının öğretmenler tarafından kullanımı ve eğitimdeki yeri. *Eğitim Bilimleri Araştırmaları Dergisi*, 9(3), 123–140.

- Karanlık, G. (2024). Algoritma ve programlama eğitiminin ortaokul öğrencilerinin bilgi işlemsel düşünme ve yaratıcı düşünme becerisine etkisi (Tez No. 855496) [Yüksek lisans tezi, Afyon Kocatepe Üniversitesi -Afyonkarahisar]. Yükseköğretim Kurulu Ulusal Tez Merkezi.
- Karataş, H. (2021). 21. yy. becerilerinden robotik ve kodlama eğitiminin Türkiye ve dünyadaki yeri. *21. Yüzyılda Eğitim ve Toplum*, 10(30), 693–729.
- Keskin, D., & Seveli, O. (2024, Mayıs). *Eğitim alanında yapay zekâ ve etik* [Konferans bildirisi]. Topkapı Conference on Artificial Intelligence in Education, İstanbul, Türkiye.
- Krajcik, J. S., & Blumenfeld, P. C. (2006). Project-based learning. In R. K. Sawyer (Ed.), *The Cambridge handbook of the learning sciences* (pp. 317–334). Cambridge University Press.
- Kulik, J. A., & Fletcher, J. D. (2016). Effectiveness of intelligent tutoring systems: A meta-analytic review. *Review of Educational Research*, 86(1), 42–45.
- Kutlucan, E., & Seferoğlu, S. S. (2024). Eğitimde yapay zekâ kullanımı: ChatGPT'nin KEFE ve PEST analizi. *Türk Eğitim Bilimleri Dergisi (TEBD)*, 22(2), 1059–1083. <https://doi.org/10.37217/tebd.1368821>
- Kutlucan, E., Çakır, R., & Ünal, Y. (2017, Ekim 11–13). *Kodlama öz yeterliliğinin problem çözme ve bilgisayarca düşünme üzerine etkisi* [Konferans bildirisi]. 5th International Instructional Technologies & Teacher Education Symposium (ITTES'17), İzmir, Türkiye.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., ... & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, 2(1), 32–37.
- Lent, R. W., Brown, S. D., & Hackett, G. (2000). Contextual supports and barriers to career choice: A social cognitive analysis. *Journal of Counseling Psychology*, 47(1), 36–49.
- Liu, C., McKelroy, E., Corliss, S., & Carrigan, J. (2020). Investigating the efficacy of AI-powered learning systems for high school students' mathematics learning. *Educational Technology Research and Development*, 68(3), 1081–1100.
- Lo, C. K. (2023). What is the impact of ChatGPT on education? A rapid review of the literature. *Educational Technology Review*, 25(2), 120–138.
- Łodzikowski, K., Foltz, P. W., & Behrens, J. T. (2023). Generative AI and its educational implications. *arXiv*. <https://doi.org/10.48550/arxiv.2303.12712>
- Luckin, R., Holmes, W., Griffiths, M., & Forcier, L. B. (2016). *Intelligence unleashed: An argument for AI in education*. Pearson Education.
- Ma, B., Li, C., & Shin'ichi, K. (2024). Enhancing programming education with ChatGPT: A case study on student perceptions and interactions in a Python course. *arXiv*. <https://arxiv.org/abs/2403.15472>
- Margolis, H., & McCabe, P. P. (2006). Improving self-efficacy and motivation: What to do, what to say. *Intervention in School and Clinic*, 41(4), 218–227.
- Meço, G., & Coştu, F. (2022). Eğitimde yapay zekânın kullanılması: Betimsel içerik analizi çalışması. *Karadeniz Teknik Üniversitesi Sosyal Bilimler Enstitüsü Sosyal Bilimler Dergisi*, 12(23), 171–191.
- Mello, R. F., Freitas, E., Pereira, F. D., Cabral, L. S., Tedesco, P., & Ramalho, G. (2024). *Education in the age of generative AI: Context and recent developments*. CESAR Innovation Center.

- Memarian, B., & Doleck, T. (2023). ChatGPT in education: Methods, potentials, and limitations. *Education and Information Technologies*, 28(1), 67–84.
- Mıhçı Türker, P., & Pala, F. K. (2018). Ortaokul öğrencilerinin, öğretmenlerin ve öğrenci velilerinin kodlamaya yönelik görüşleri. *İlköğretim Online*, 17(4), 2013–2029.
- Millî Eğitim Bakanlığı (MEB). (2018). *Bilişim teknolojileri ve yazılım dersi öğretim programı*. Millî Eğitim Bakanlığı Yayınları.
- Nakajima, T. (2020). Programming education in Japan: Current status and future direction. *Japanese Journal of Educational Technology*, 64(2), 123–131.
- Ogunleye, B., Zakariyyah, K. I., Ajao, O., Olayinka, O., & Sharma, H. (2024). A systematic review of generative AI for teaching and learning practice. *Education Sciences*, 14(6), 636. <https://doi.org/10.3390/educsci14060636>
- Olgun, K. B. (2024). Programlamanın ortaokul öğrencilerinin düşünme stilleri üzerine etkisi (Tez No. 380549) [Yüksek lisans tezi, İstanbul Üniversitesi-İstanbul]. Yükseköğretim Kurulu Ulusal Tez Merkezi.
- Ouh, E. L., Gan, B. K. S., Shim, K. J., & Wlodkowski, S. (2023). ChatGPT, can you generate solutions for my coding exercises? An evaluation on its effectiveness in an undergraduate Java programming course. *Journal of Information Technology Education*, 20(1), 45–63.
- Pirim, H. (2006). Yapay zeka. *Journal of Yasar University*, 1(1), 81–93.
- Polat, E. (2022). *Nitel araştırma yöntemleri*. Seçkin Yayıncılık.
- Qureshi, B. (2023). Exploring the use of ChatGPT as a tool for learning and assessment in undergraduate computer science curriculum: Opportunities and challenges. *Computing Education Research*, 40(1), 56–70.
- Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv*. <https://arxiv.org/abs/1511.06434>
- Rahman, M. M., & Watanobe, Y. (2023). ChatGPT for education and research: Opportunities, threats, and strategies. *Educational Research Review*, 32(1), 89–107.
- Ramazanoğlu, M., & Ayaz, B. (Eds.). (2023). *Geleceğin eğitimi: Yapay zeka ve dijital öğrenme*. BİDGE Yayınları.
- Rapaport, W. J. (2017). What is computer science? *American Philosophical Quarterly*, 54(1), 29–41.
- Rasul, T., Nair, S., Kalendra, D., Robin, M., Ladeira, F. O. W. J., Sun, M., Day, I., Rather, R. A., & Heathcote, L. (2023). The role of ChatGPT in higher education: Benefits, challenges, and future research directions. *Journal of Higher Education Policy and Management*, 45(2), 111–128.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60–67.
- Rovshenov, A. (2020). Programlama eğitimi ile ilgili yapılan araştırmaların içerik analizi (Tez No. 643094) [Yüksek lisans tezi, Ege Üniversitesi -İzmir]. Yükseköğretim Kurulu Ulusal Tez Merkezi.
- Russell, S. J., & Norvig, P. (2020). *Artificial intelligence: A modern approach* (4th ed.). Pearson.

- Saygıner, Ş., & Tüzün, H. (2016). Programlama eğitiminde yaşanan zorluklar ve çözüm önerileri. *Mustafa Kemal Üniversitesi Eğitim Fakültesi Dergisi*, 13(26), 1–15.
- Saygıner, Ş., & Tüzün, H. (2018). Programlama eğitimi üzerine bir inceleme: Yaşanan zorluklar, mevcut uygulamalar ve güncel yaklaşımlar. In H. F. Odabaşı, B. Akkoyunlu, & A. İşman (Eds.), *Eğitim teknolojileri okumaları 2018* (ss. 693–710). Eğiten Kitap.
- Schunk, D. H., & Pajares, F. (2009). The development of academic self-efficacy. In A. Wigfield & J. S. Eccles (Eds.), *Development of achievement motivation* (pp. 35–40). Academic Press.
- Selby, C. C., & Woollard, J. (2013). *Computational thinking: The developing definition* [E-print]. University of Southampton. <https://eprints.soton.ac.uk/356481/>
- Sun, D., Boudouaia, A., Zhu, C., & Li, Y. (2023). Would ChatGPT facilitated programming mode impact college students' programming behaviors, performances, and perceptions? An empirical study. *Journal of Educational Technology Research*, 22(2), 77–96.
- Şanlı, M., & Alper, A. (2024). Investigation of postgraduate theses on coding and robotics education in Türkiye. *Ankara University Journal of Faculty of Educational Sciences*, 57(2), 759–803. <https://doi.org/10.30964/auebfd.1282226>
- Şimşek, H. (2021). *Eğitimde program geliştirme ve değerlendirme*. Nobel Yayın Dağıtım.
- Şişman, G., & Küçük, S. (2018). Robotik tutum ölçeği: Geçerlilik ve güvenirlik çalışması. *Eğitim ve Bilim*, 43(195), 129–148.
- Tanrıkulu, H. (2024). *Python ile programlamaya giriş* (1. baskı). MCNA | Pitonca Yayınları.
- Tekin, N. (2023). Eğitimde yapay zekâ: Türkiye kaynaklı araştırmaların eğilimleri üzerine bir içerik analizi. *Necmettin Erbakan Üniversitesi Ereğli Eğitim Fakültesi Dergisi*, 5(Özel Sayı), 387–411. <https://doi.org/10.51119/ereegf.2023.49>
- Tian, H., Lu, W., Li, T.-O., Tang, X., Cheung, S.-C., Klein, J., & Bissyandé, T. F. (2023). Is ChatGPT the ultimate programming assistant: How far is it? *arXiv*. <https://arxiv.org/abs/2304.11938>
- UNESCO. (2024). *Eğitim ve araştırmada üretken yapay zeka kılavuzu*. UNESCO Yayınları. <https://www.unesco.org/en/open-access/cc-sa>
- Uyar, A., Öztürk, G., & Öztürk, Y. (2022). Ortaokul öğrencilerinin kodlamaya yönelik tutumları. *İnönü Üniversitesi Eğitim Bilimleri Enstitüsü Dergisi*, 9(18), 81–92. <https://doi.org/10.29129/inuigse.1166046>
- Uzun, Y., Tümtürk, A. Y., & Öztürk, H. (2021). Günümüzde ve gelecekte eğitim alanında kullanılan yapay zekâ. In *1st International Conference on Applied Engineering and Natural Sciences (ICAENS)*, Konya, Türkiye. <https://doi.org/10.5281/zenodo.355809268>
- Ünal, E. (2023). Üretken yapay zekâ çağı ve ChatGPT. *Bilim ve Teknik*, 58(9), 44–52.
- Üstün, A. B. (2024). Eğitim alanında yapay zekâ uygulamaları: Lisansüstü tezlerin sistematik incelemesi. *Bilgi ve İletişim Teknolojileri Dergisi*, 6(2), 95–112. <https://doi.org/10.53694/bited.1593139>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems* (pp. 5998–6008).
- Vukojevic, M., & Krstic, J. (2023). ChatGPT in programming education: ChatGPT as a programming assistant. *Journal for Contemporary Education*, 12(4), 67–84.

- Waseem, M., Das, T., Ahmad, A., Liang, P., Fahmideh, M., & Mikkonen, T. (2024). ChatGPT as a software development bot: A project-based study. In *Proceedings of the International Conference on Advanced Computing and Applications* (pp. 1–20). <https://zenodo.org/doi/10.5281/zenodo.10457831>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>
- Yeşilyurt, S., Dündar, R., & Demir, R. Z. (2024). Türkiye’de yapay zekâ ve eğitim ilişkisini inceleyen lisansüstü tezlerin analizi: Bir meta sentez çalışması. *Journal of Innovative Research in Social Studies*, 7(1), 47–73. <https://doi.org/10.47503/jirss.1484848>
- Yıldırım, A., & Şimşek, H. (2006). *Sosyal bilimlerde nitel araştırma yöntemleri*. Seçkin Yayıncılık.
- Yıldız, M., Çiftçi, E., & Karal, H. (2017). Bilişimsel düşünme ve programlama. In A. İşman, H. F. Odabaşı, & B. Akkoyunlu (Eds.), *Eğitim teknolojileri okumaları 2017* (ss. 75–86). TOJET.
- Yılmaz, R., & Karaoğlu Yılmaz, F. G. (2020). The effect of generative artificial intelligence (AI)-based tool use on students’ computational thinking skills, programming self-efficacy and motivation. *Journal of Educational Technology*, 45(3), 123–145.
- Yiğit, B. (2021). Programlama eğitimi araştırmalarında bilimsel eğilimler: Bir içerik analizi örneği (Tez No. 691759) [Yüksek lisans tezi, Yıldız Teknik Üniversitesi -İstanbul]. Yükseköğretim Kurulu Ulusal Tez Merkezi.
- Yuill, S., & Halpin, H. (2006). *Guide to Python programming*. GNU Free Documentation License.
- Yükseköğretim Kurumları. (2024). *Bilimsel araştırma ve yayın faaliyetlerinde üretken yapay zekâ kullanımına dair etik rehber*. Yükseköğretim Kurumları Yayınları.
- Zawacki-Richter, O., Marín, V. I., Bond, M., & Gouverneur, F. (2019). Systematic review of research on artificial intelligence applications in higher education – Where are the educators? *International Journal of Educational Technology in Higher Education*, 16(1), 1–27.
- Zheng, Y. (2023). ChatGPT for teaching and learning: An experience from data science education. *Data Science Education Journal*, 12(4), 98–115.
- Zhu, C., Sun, M., Luo, J., Li, T., & Wang, M. (2023). How to harness the potential of ChatGPT in education? *International Journal of Educational Technology*, 19(3), 78–95.

EKLER

EK-1. Programlama Başarı Testi

Adı:	ORTAOKUL PROGRAMLAMA EĞİTİMİNDE ÜRETKEN YAPAY ZEKA DESTEKLİ BİR ÖĞRENME MODÜLÜNÜN TASARLANMASI, UYGULANMASI VE DEĞERLENDİRİLMESİ YÜKSEK LİSANS ÇALIŞMASI	<u>Not</u>
Soyadı:		
PROGRAMLAMA BAŞARI TESTİ		

1.Soru) Bir okul, öğrencilerin sınav sonuçlarını değerlendirmek için bir sistem oluşturmak istiyor. Her öğrencinin sınav notlarını (0-100 arası) bir liste olarak alacak ve şu çıktıları üretecek bir algoritma yazınız:

- Ortalama not.
- Geçen (notu 50'nin üzerinde olan) ve kalan öğrenci sayısı.
- En yüksek ve en düşük notlar.

2.Soru) Bir dizi içindeki tüm çift sayıları ve toplamalarını ekrana yazdıran bir Python kodu yazınız. Kodunuzun doğru çalıştığını test etmek için [3, 6, 8, 10, 15, 20] dizisini kullanınız.

3.Soru) Aşağıdaki Python kodunda bazı hatalar bulunmaktadır. Kodun hatalarını bulun ve düzeltiniz. Hataları nasıl düzelttiğinizi açıklayınız.

```
python

def calculate_area(radius):
    pi = 3.14
    area == pi * radius ** 2
    return area

print("The area is: " + calculate_area(5))
```

4.Soru) Bir alışveriş uygulamasında, her müşterinin satın aldığı ürünlerin toplam fiyatını hesaplayan ve şu çıktıları üreten bir Python programı yazınız:

- Her müşteri için toplam fiyat.
- 100 TL üzerinde harcama yapan müşterilerin sayısı.

5.Soru) Bir oyun için bir Python programı yazınız. Program, kullanıcının girdiği bir kelimenin harflerini rastgele bir sırada yeniden düzenleyecek ve yeni bir kelime oluşturacaktır. Kullanıcı bu yeni kelimeyi tahmin etmeye çalışacaktır. Programın her yanlış tahminde ipucu vermesini sağlayınız.

6.Soru) Aşağıdaki Python kodu, kullanıcıdan alınan bir sayının faktöriyelini hesaplamayı amaçlamaktadır. Ancak, kodda birkaç hata bulunmaktadır. Kodun hatalarını bulun, düzeltin ve düzeltilmiş haliyle doğru bir şekilde çalıştırınız. Hataları nasıl düzelttiğinizi açıklayınız.

```
def factorial(n):  
    result = 0  
    for i in range(n):  
        result = result * i  
    return result
```

EK-2. Bilge Kunduz Bebras Testi

1 Su Baskını

Kunduzların su deposunun beş girişi ve bir çıkışı vardır. Normal zamanda bu girişler ve çıkış kapalıdır. Kemal'in işi yüksek su işaretlerini (yanan uyarı lambaları) saymak ve su baskını riski olduğunda çıkışı açmaktır. Kemal geniş olan C ve D nehirlerinin her birini iki yüksek su işareti olarak saymaktadır, çünkü bu nehirler özellikle taşkınlar sırasında depoyu hızla doldurmaktadır. Toplamda dört yüksek su işareti saydığında, örneğin A, B ve C ya da D'yi iki yüksek su işareti olarak sayarak çıkışı açmalıdır?

A) B) C) D)

2 Hediye Seçimi

Meltem'in sınıfına Filiz adında yeni bir öğrenci katılır. Filiz'e hoş geldin sürprizi yapmak isteyen Meltem, ona hediye almak ister. Fakat Filiz'in nelerden hoşlandığını bilmemektedir. Doğru hediyeyi seçmek için Filiz'e sorular sormaya ve yanıtlarından yola çıkarak ona en uygun hediyeyi seçmeye karar verir. Meltem aşağıdaki bulunan akış diyagramını kullanarak Filiz'e sorular sormaya başlar.

En sonunda Meltem, Filiz'in müzikten çok hoşlandığını fakat şarkı söylemekten ve enstrüman çalmaktan hoşlanmadığını öğrenir.

Filiz'in verdiği yanıtlardan yola çıkarak, Meltem aşağıdaki hediyelerden hangisini almalıdır?

A) B) C) D)

3 Çiçek Bahçesi

Bilge Kunduz'un, etrafı çit ile örülü, köşelerinde direklerin olduğu bir bahçesi vardır (Görsel A). Bahçesine çiçek ekmek isteyen Bilge Kunduz, bahçesini aşağıdaki kullarallara göre 7 parçaya ayırır.

- Ayırıldığı bölgeler yalnızca üçgen olabilir (Görsel B).
- İpler birbirleri ile kesişemez (Görsel C-1).
- İpler çiti geçemez (Görsel C-2).
- İpler köşelerdeki direklere sabitlenmelidir (Görsel C-3).

Bilge Kunduz, yukarıda verilen bahçeyi kullaralla uygun bir şekilde 7 bölgeye ayırıp çiçek ekerse bahçe aşağıdakilerden hangisi gibi görünür?

A) B) C) D)

4 Kahlvaltı

Aslı, Baran, Cem ve Deniz adında dört kardeş anne ve babası için güzel bir kahlvaltı hazırlamak isterler. Hazırlayacakları kahlvaltı için dört ayrı malzemeye ihtiyaçları vardır; yumurta, sos, peynir ve domates. Bu malzemeleri almak için markete giderler. Marketten iki kutu yumurta, iki paket sos, iki kalıp peynir ve iki adet domates olmak üzere toplamda sekiz adet malzeme alırlar.

Marketten eve döndüklerinde her kardeşin iki farklı malzeme satın aldığı fark ederler. Ayrıca;

- Aslı yumurta satın almıştır. Fakat Baran yumurta almamıştır.
- Cem ve Baran peynir satın almıştır.
- Deniz ve Cem aldıkları malzemelerden biri aynıdır.
- Baran sos almamıştır.

Eğer Aslı ve Deniz'in aldıkları malzemelerden biri aynı ise bu malzeme aşağıdakilerden hangisidir?

A) Yumurta B) Sos C) Peynir D) Domates

5 Yürüyüş Parkuru

Bilge Kunduz evinin yakınında olan bir yürüyüş parkurunda yürüyüş yapmaktan çok hoşlanmaktadır. Bir gün yanına fotoğraf makinesini alır ve yürüyüş yaparken parkurun fotoğraflarını çekmek ister.

Bilge Kunduz, yürüyüş sırasında çekmiş olduğu fotoğrafları nereye ait olduğunu anlamak için aşağıdaki tabloda verilen semboller kullanır. Her fotoğraf tüm nesneleri 3 x 3'lük ızgara şeklinde göstermektedir.

Bir gün kunduz yürüyüş sırasında çekmiş olduğu fotoğrafları nereye ait olduğunu anlamak için şöyle bir not alır.

Düz İlerle Sağa Dön Sola Dön Fotoğraf çek

Aşağıdaki seçeneklerden hangisinde kunduzun o gün çektiği fotoğraflar yer almaktadır?

A) B) C) D)

6 Oyuncak Fabrikası

Bir oyuncak fabrikasında, her biri oyuncak üretiminde belirli bir görevi yerine getiren üç makine vardır.

İlk makine oyuncakın üretiminden sorumludur. Makine, rastgele bir harf ve sayı kombinasyonuna bağlı bir oyuncak oluşturma komutu alır.

A-8ir kunduz yap B-8ir tren yap C-8ir yo-yo yap

1-Onu mavî çizgilerle boyay 2-Onu kırmızı beneklerle boyay 3-Onu yeşile boyay

Örneğin: [A,2] komutu aşağıdaki oyuncakı oluşturacaktır.

İkinci makine aşağıda gösterilen oyuncakları ayırıp çıkarır, çünkü pazarlama araştırmaları çocukların bu oyuncaklardan hoşlanmadığını göstermektedir.

Üçüncü makine oyuncakları kendi kutularına koyarak paketler. Yo-yolar kutuda 1 birim, trenler 2 birim ve kunduzlar 3 birim yer kaplar. Kutularda tam olarak 8 birimlik alan bulunmaktadır.

Bir kutuyu tamamen dolduran bir oyuncak seti oluşturmak isteniyor. Bu durumu sağlayan komut dizisi aşağıdaki seçeneklerden hangisidir?

A) [B,3], [C,2], [B,2], [A,2] B) [A,1], [A,2], [B,1], [A,3]
C) [A,3], [C,2], [A,1], [B,2] D) [B,1], [A,2], [A,3], [C,1]

7 İki Robot

Esma Kunduz iki robot (A ve B) ile deney yapıyor. Robotlar karelerden oluşan bir zemin üzerinde bir karede başlıyor ve kareye hareket edebiliyorlar.

Esma her iki robotu da tek bir denetleyiciyle kontrol ediyor: Komutlarının her biri, her iki robotun da aynı hareketi gerçekleştirmesini sağlıyor. Denetleyicinin üç komutu vardır:

- Bir sonraki kareye geçin (robotun o anda baktığı yönde bir kare ilerler)
- 90 derece sola dönmün (robot bulunduğu karede kalır).
- 90 derece sağa dönmün (robot bulunduğu karede kalır).

Örneğin: Robotlar yandaki gibi konumlandırıldığında, komut dizisi robotların renkli hücreler boyunca hareket etmesini sağlar.

Her iki robotun da ortadaki eve doğru hareket etmesini sağlayacak komut dizisi nasıl olmalıdır? Komutların, robotlardan herhangi birinin engelleri (gri hücrelere) çarpmasına veya ızgaranın dışına çıkmasına neden olmadığında emin olun.

A) B) C) D)

8 Yorgun Sayı

Dijital saat, zamanı dört set (rakam olarak adlandırılır) ışıklı çubuk kullanarak gösterir. Her rakam 7 tane ışıklı çubuğa sahiptir. Her bir rakam şöyle görünür:

Bu çubuklar yandığında aşırır. Bu sebeple hafta boyunca toplamda en uzun süre yanan çubuğun ilk önce değiştirilmesi gerekir.

Aşağıda isimlendirilen 28 parçadan hangisinin ilk önce değiştirilmesi gerekir?

A) 18 B) 2A C) 3A D) 1C

9

Örüntü Bulma

Kunduzlardan Aslı ve Berat yandaki gibi dekore edilmiş bir masada oturmaktadır:

Aslı, Berat'a şöyle der: Şu desene bak, Bu deseni döndürebilirsin ancak yansıtamayabilirsin. Buna uyan iki tane örnek bulabilirim:



Büyük masada bu desenden kaç tane bulabilirsin?



Aslı'nın Berat'a vereceği yanıt kaçtır?

- A) 3 B) 4 C) 5 D) 6

10

Mektup Postalama

Peşami beş kunduz mektup dağıtır. Kunduzlar birbirlerine mektup göndermek istediklerinde, mektubu Peşami'nin teslim etmesi için kendi posta kutularına bırakırlar. Peşami kunduzların posta kutularını tek tek ziyaret eder. Her posta kutusuna vardığında iki şey yapar: Posta kutusundaki tüm mektupları toplar ve çantasına koyar. Sonrasında o kunduzun çantasındaki tüm mektupları posta kutusuna koyarak teslim eder. Peşami herhangi bir mektubu teslim etmeden önce kunduzların posta kutularında aşağıdaki mektupların bulunduğunu biliyor.



Peşami boş bir çantayla güne başlar ve her posta kutusunu yalnızca bir kez ziyaret eder. Bu nedenle, tüm mektupları aynı gün içerisinde teslim edebilmek için posta kutularını ziyaret etme sırası konusunda mantık yürütmesi gerekir.

Peşami'nin tüm mektupları teslim edebilmesi için posta kutularını hangi sırayla ziyaret etmesi gerekir?

- A) Gizem → Caner → Leman → Sude → Teoman
B) Gizem → Sude → Caner → Teoman → Leman
C) Gizem → Caner → Sude → Leman → Teoman
D) Caner → Gizem → Sude → Leman → Teoman

13

Emine'nin Görevi

Aşağıdaki harita Emine'nin köyünün haritasıdır. Haritadaki rakamlar, Emine'nin bir yerden başka bir yere yürümesinin ne kadar sürdüğünü (dakika cinsinden) gösterir.

Bir günde Emine'nin aşağıdaki görevleri yapması gerekiyor:

- Demirciden bir at nali al
- Eczaneden ilaç al
- Pazardan meyve al

Bu görevler herhangi bir sırayla gerçekleştirilebilir. Emine görevleri tamamlamaya evinden başlar ve görevleri tamamladıktan sonra tekrar evine döner. Her bir görevin tamamlanması için ilgili yerde 1 dakika harcar.

Emine aşağıdaki rotayı izlerse, bu onun 41 dakikasını alır.

Emine'nin evi → Pazar (meyve) → Fırıncı → Demirci (at nali) → Park → Eczane (ilaç) → Okul → Emine'nin evi (Aslında: 4+6+3+9+7+3+6 38 dakikadır ve bu süreye görevler için 3 dakika ekleriz.)

Emine'nin tüm görevlerini yerine getirebileceği ve tekrar eve dönebileceği en kısa süre nedir?

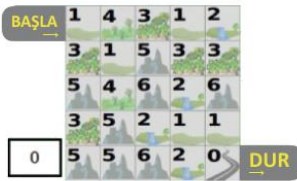
- A) 37 dakika B) 38 dakika C) 39 dakika D) 40 dakika

14

Labirent Oyunu

Bir kunduz oyun oynamaya karar verir. Oyun, karelere ayrılmış arazide oynanır. Karelereki sayılar, kunduzun o karedeyken ne kadar enerji harcadığını gösterir. Toplam harcadığı enerji yola başladığı sol üst kare ile yolu bitirdiği sağ alt kare de dahil olmak üzere geçtiği tüm kareleri kapsar. Kunduz sadece aşağı ve sağa doğru hareket edebilir.

Kunduzun BAŞLA noktasından DUR noktasına ulaşırken harcayacağı minimum enerji kaç olabilir?



- A) 14 B) 15 C) 16 D) 17

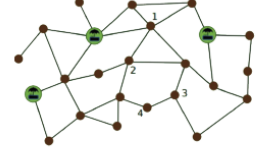
Meyve Suyu Arabaları

11

Yaz aylarında Kunduz Dışarıda hava çok sıcak olduğundan belediye başkanı şehrin bazı köşelerine meyve suyu arabaları yerleştirmeye karar verdi. Meyve suyu arabalarının konumlandırılması gereken köşeler, her vatandaşın şehrin herhangi bir köşesinden en fazla 2 sokak yürüyerek meyve suyunu erişebilmelerini sağlayacak şekilde seçildi. Aşağıdaki şehir haritasında kahverengi noktalar köşeleri, noktaları veya daireleri birleştiren çizgiler sokakları ve meyve suyu arabası simgeleri, meyve suyu arabasının hali hazırda yerleştirildiği köşeleri temsil eder.

Ne yazık ki belediye başkanı, köşelerden birinden (hangisi olduğunu hatırlamıyordu) en fazla 2 sokak yürüyerek meyve suyu arabasına ulaşmanın mümkün olmadığına dair bir şikâyet alır. Bu sorunu çözmek için belediye başkanı, yandaki haritada gösterilen köşelerden birine ilave meyve suyu arabası yerleştirmesini talep eder.

Şehrin herhangi bir köşesinden en fazla iki sokak yürüyerek meyve suyu arabasına ulaşabilmesi için yeni meyve suyu arabası nereye konumlandırılmalıdır?



- A) Köşe 1 B) Köşe 2 C) Köşe 3 D) Köşe 4

Nakliye

12

Kunduz Nakliye şirketi çalışanları, depolar arasında belgeleri taşımaktadır. İş için gerekli belgeleri küçük bir kamyon kullanarak bir depodan başka bir depoya taşımak zorundadırlar. Belgeler, kamyonu yüklenmeden önce kollarına yerleştirilir. Kolların kamyonu yerleştirilmesi ve kamyonun indirilmesi her kolu için 1 dakika sürmektedir. İki depo arasındaki toplam yolculuk süresi 60 dakikadır. Kamyon bir seferde en fazla 20 koli taşıyabilir. Çalışanlar 20'den fazla koli taşımak isterse, kamyonun teslim edilmesi gereken depoya gidip geri gelmesini beklemek zorundadır.

Belgeler kollarına her zaman düzenli bir şekilde yerleştirilebilir. Belgeleri aşağıdaki 3 yöntemden birini kullanarak kollarına düzenli bir şekilde yerleştirerek toplam koli sayısı azaltılabilir.

Yöntem A: Koli başına 3 dakika vakit ayırır ve kutu sayısı yarı yarıya azalır.

Yöntem B: Koli başına 1 dakika vakit ayırır ve kutu sayısı üçte bir oranında azalır.

Yöntem C: Düzenli bir yerleştirme için vakit harcanmaz.

Çalışanlar 36 koliyi 4 saatten daha kısa bir sürede taşımak istiyor. Bunun için üç yöntemden hangisini kullanabilirler?

- A) Yalnız Yöntem A B) Yalnız Yöntem B C) Yalnız Yöntem C D) Yalnız Yöntem A veya B

Basit Kelime İşlemci

15

Basit bir kelime işlem aracı düşünün. Bir harfe basmak, ilgili karakteri imlecin soluna ekler. Özel tuşlar, imlecin başlangıç konumunun 1 ile gösterildiği bu tabloda açıklandığı gibi çalışır:

"←" tuşuna basmak, imlecin solundaki bir karakteri hareket ettirir.	bebras	"→" tuşuna basmak imlecin bir karakter sağına gider.	bebras
"backsp" tuşuna (backspace) basmak, imlecin solundaki karakteri siler.	bebras (backsp)	"del" tuşuna (delete) basmak, imlecin sağındaki karakteri siler.	bebras (del)
"home" tuşuna basmak imlecin metnin başına taşır.	bebras (home)	"end" tuşuna basmak imlecin metnin sonuna taşır.	bebras (end)

Bir iletim hatası nedeniyle bazı harfler "*" olarak değiştirilmiş ve orijinal "abcde*ghijklm" metni geri yüklenmek isteniyor. Aşağıda gösterilen iki ayrı işlem için sırasıyla minimum kaç tuş vuruşu gerekir? Başlangıç olarak imleç kelimenin sonundadır.

abcde*ghijklm |
abcde*ghij*** |

- A) 4-9 B) 5-11 C) 8-13 D) 7-12

Tebrikler!



Sınavınız Bitmiştir.
Sınav Kitapçığı ve
Yanıt kağıdınızı sorumlu
öğretmene teslim ediniz.

EK-3. Bilgi İşlemsel Düşünme Öz Değerlendirme Ölçeği

22 / Rumeli Journal of Education Studies 2023.4 (June)

Bilgi işlemsel düşünme öz değerlendirme ölçeği: Ölçek geliştirme çalışması / Ceylan, V. K. & Akar Vural, R.

Appendix 1. Self-Assessment of Computational Thinking Skill Scale (Turkish Version)

Sevgili Öğrenciler, bu ölçek sizlerin bilgi işlemsel düşünme becerilerinizin özdeğerlendirmesine yönelik bilimsel bir çalışmada kullanılmak üzere tasarlanmıştır. Lütfen her bir maddeyi özenle ve dikkatle okuyunuz ve kendinize en uygun cevabı "tamamen katılıyorum", "katılıyorum", "kısmen katılıyorum", "katılmıyorum", "hiç katılmıyorum" şeklinde kodlayınız. Katılımınızdan dolayı şimdiden çok teşekkür ederim.

Cinsiyetiniz: Kız () Erkek ()

Sınıfınız: 6. Sınıf (). 7. Sınıf ()

Günlük olarak Ekran (Akıllı Telefon, Tablet, Bilgisayar, Oyun konsolu vb diğer araçları) Kullanım süreniz:
0-1 saat () 1-2 saat () 2-3 saat () 3-4 Saat () 4 saat ve daha fazlası ()

		Kesinlikle Katılmıyorum	Katılmıyorum	Kısmen Katılıyorum	Katılıyorum	Kesinlikle Katılıyorum
		(1)	(2)	(3)	(4)	(5)
1	Problemin çözümünde takip etmem gereken adımları kendim belirlerim.	()	()	()	()	()
2	Uygulamalarımnda kullandığım döngülerin sınırlarını belirlerim.	()	()	()	()	()
3	Uygulamalarımnda hangi işlemlerin döngü sınırları içerisinde olması gerektiğini hesaplarım.	()	()	()	()	()
4	Bir problemin çözümünden yola çıkarak yeni bir problemi çözerim.	()	()	()	()	()
5	Çalışan bir kod bloğunda değişiklikler yaparım.	()	()	()	()	()
6	Uygulamalarımnda karşılaştığım hataları kolaylıkla bulurum.	()	()	()	()	()
7	Uygulamalarımnda birden fazla nesneyi aynı sahnede kullanırım.	()	()	()	()	()
8	Uygulamalarımnda birden fazla olay kullanırım.	()	()	()	()	()
9	Uygulamalarım çalıştığı anda arka planda gerçekleşen olayları (işlemleri) açıklarım.	()	()	()	()	()
10	Problemlerin çözümü için farklı programlama dilleri kullanırım.	()	()	()	()	()
11	Uygulamamdaki gereksiz (önemsiz) adımları çıkartırım.	()	()	()	()	()
12	Uygulamalarımndaki işlem adımlarını önem sırasına göre sıralarım.	()	()	()	()	()
13	Problemlerin çözümünü formül ile gösterebilirim.	()	()	()	()	()
14	Problemlerin çözümünde operator (+, -, *, /), ifade ve eşitlikleri kullanarak çözüm üretirim.	()	()	()	()	()
15	Problemin çözüm sürecini daha küçük parçalara ayırırım.	()	()	()	()	()
16	Problemlerin çözümünde programlama dili ya da programlama kavramı kullanmadan da çözümü gerçekleştirebilirim.	()	()	()	()	()
17*	Problemlerin çözümü için programlama dili kullanma zorunluluğu hissetmem.	()	()	()	()	()
18	Problem çözümünde en kısa çözüm yolunu kullandığımı düşünürüm.	()	()	()	()	()
19	Problem çözümünde bilgisayar veya diğer araçları kullanırım.	()	()	()	()	()
20	Tekrar eden ve birbirine benzer işlemleri bilgisayarlara ve diğer araçlara (Robot, tablet, telefon vb.) yaptırabilirim.	()	()	()	()	()

Not: * ters kodlu madde

Adres	Address
Rumeli Eğitim Araştırmaları Dergisi	Rumeli Journal of Education Studies
İstanbul Medeniyet Üniversitesi, Eğitim Bilimleri Fakültesi, Türkçe ve Sosyal Bilimler Bölümü, Türkçe Eğitimi ABD	İstanbul Medeniyet University, Faculty of Education Sciences, Dep. of Turkish and Social Sciences Education, Turkish Education Program
Cevizli, Kartal - İSTANBUL / TÜRKİYE 34865	Cevizli, Kartal - İSTANBUL / TÜRKİYE 34865
e-posta: rumelieducation@yandex.com	e-mail: rumelieducation@yandex.com
tel: +90 505 7958124, +90 216 773 0 616	phone: +90 505 7958124, +90 216 773 0 616

EK-4. Programlamaya Yönelik Tutum Ölçeği

ÖĞRENCİLERİN PROGRAMLAMAYA YÖNELİK TUTUMLARININ BELİRLENMESİ						
Madde No	Madde	Kesinlikle Katılmıyorum	Katılmıyorum	Kararsızım	Katılıyorum	Kesinlikle Katılıyorum
1	Bir programlama dilinde komut yazarken yazım (sintaks) hatalarıyla uğraşmayı sevmiyorum.	<i>Programlamaya yönelik ilgi</i>				
2	Programlama diline özgü kavramlar (değişken, döngü, dizi, fonksiyon vb) soyut olduğu için bana karmaşık görünüyor.					
3	Programlamadaki hataları ayıklama işlemi çok sıkıcıdır.					
4	Programlamanın, problemlere farklı yönlerden bakabilme yeteneğimi geliştireceğini düşünüyorum.	<i>Programlamanın Yararına Yönelik İnanç</i>				
5	Programlamanın, karar verme becerilerimi geliştireceğini düşünüyorum.					
6	Programlamanın, matematiksel düşünme becerilerimi geliştireceğine inanıyorum.					
7	Programlama yaparken çok eğlenirim.	<i>Programlamaya Yönelik İstek</i>				
8	Programlama yaparken kendimi iyi hissediyorum.					
9	Programlama öğrenmek çok zevklidir.					
10	Programlama derslerinde eğlenceli zaman geçiririm.					
11	Programlama yaparken olaylar arasındaki ilişkileri görebilme bana zevk verir.					
12	Programlama sorunlarını çözmekten zevk alıyorum.					
13	Programlama ile yeni bir ürün geliştirmek hoşuma gider.					
14	Bir programlama problemini çözmek için gerekli olan bilgiyi araştırırken zevk alırım.					

EK-5. Yarı Yapılandırılmış Görüşme Formu

Üretken Yapay Zeka Destekli Programlama Öğrenme Modülü Öğrenci Değerlendirme Formu

Bu görüşme, programlama ve python programlama dilini öğrenmenize yardımcı olan üretken yapay zeka modülünü nasıl bulduğunuzu anlamamıza yardımcı olmak için tasarlanmıştır. Düşüncelerinizi, sevdiklerinizi ve belki de zorlandığınız yerleri öğrenmek istiyoruz. Modül hakkındaki düşünceleriniz ve katılımınız için teşekkür ederim.

Katılımcı Bilgileri:

- İsim:
- Yaş:
- Sınıf: 7

- Bu modülü kullanmaya başladığında neler hissettin? İlk olarak ne düşündün?
- Modülde en çok hangi bölümleri veya özellikleri sevdin? Neden bunları sevdin?
- Modülü kullanırken herhangi bir zorluk yaşadın mı? Eğer yaşadıysan, bunlar nelerdi ve bunların nasıl daha iyi olabileceğini düşünüyorsun?
- Bu modül, programlama öğrenirken sana nasıl yardımcı oldu? Öğrendiğin yeni şeyler oldu mu?
- Modül sana geri bildirim verdiğinde bu bilgiler öğrenmene nasıl yardımcı oldu? Bu geri bildirimler sana yeterli geldi mi?
- Bu modülün daha sık kullanılmasını ister miydin? Neden daha fazla veya daha az kullanmak istersin?
- Modülün kullanımı kolay mıydı? Arayüzü hakkında ne düşünüyorsun, daha nasıl iyileştirilebilir?
- Bu tür bir yapay zeka modülünü başka hangi derslerde kullanmak isterdin? Hangi konularda bu modülün yardımcı olabileceğini düşünüyorsun?

Bize bu bilgileri verdiğin için çok teşekkür ederiz! Düşüncelerin ve fikirlerin, bu eğitim aracını daha iyi hale getirecek ve gelecekte nasıl kullanılacağına dair fikirler verecek.

EK-6. Bireysel Bilgi Formu

Bireysel Bilgi Formu

Adı Soyadı:

Sınıfı:

Cinsiyeti: ☐ Kız ☐ Erkek ☐ Belirtmek istemiyorum

Okuduğunuz Okul Türü: ☐ Devlet Ortaokulu ☐ Özel Ortaokul

Daha önce herhangi bir programlama dili ile çalıştınız mı? ☐ Evet ☐ Hayır

Eğer “Evet” yanıtladıysanız, hangi programlama dil(ler)ini kullandınız?

.....

Programlama eğitimini nereden öğrendiniz?

- ☐ Okulda ders olarak
- ☐ Kurs / Etkinlik
- ☐ Kendi başıma (internet / kitap)
- ☐ Ailemden / çevremden destek alarak
- ☐ Diğer:

Programlama konusunda kendinizi nasıl değerlendirirsiniz?

- ☐ Hiç bilgim yok
- ☐ Temel seviyede biliyorum
- ☐ Orta seviyede biliyorum
- ☐ İleri düzeyde program yazabiliyorum

Python programlama dilini duydunuz mu? ☐ Evet ☐ Hayır

Eğer “Evet” dediyseniz, ne düzeyde bildiğinizi işaretleyiniz:

- ☐ Yalnızca adını biliyorum
- ☐ Basit kodlar yazabiliyorum
- ☐ Karmaşık kodlar yazabiliyorum

Daha önce algoritma ve akış şemalarıyla çalıştınız mı? ☐ Evet ☐ Hayır

Gelecekte programlama öğrenmeye devam etmeyi düşünüyor musunuz?

- ☐ Evet ☐ Hayır ☐ Emin değilim

PROGRAMLAMA ÖĞRENME SÜRECİNE HOŞ GELDİNİZ!

Sevgili Öğrencimiz,

Bu süreçte eğlenerek öğrenmenizi, teknoloji ile yaratıcı işler yapmanızı ve geleceğin dünyasına hazırlanmanızı amaçlıyoruz.

Bu Süreçte Neler Yapacağız?

- Python programlama dilinin temel yapılarını öğreneceğiz.
- Gerçek hayat problemlerine kod yazarak çözümler üreteceğiz.
- ChatGPT gibi üretken yapay zekâ araçlarını nasıl kullanabileceğimizi göreceğiz.
- Kendi algoritmamızı tasarlayacak, hataları tespit edip düzelteceğiz.
- Birlikte eğlenecek, birlikte geliştireceğiz! 

Bu Süreç Sana Ne Katacak?

- ✓ Problem çözme ve algoritmik düşünme becerileri
- ✓ Kendi başına program yazma deneyimi
- ✓ Teknolojiye daha bilinçli bakış açısı
- ✓ Bilgisayar bilimlerine olan ilginin artması
- ✓ Takım çalışması ve proje üretme alışkanlığı

Unutma!

Hiç kimse bir günde programlama uzmanı olmadı.

 Soru sormaktan çekinme.

 Hata yapmaktan korkma.

 Denemekten vazgeçme.

 **Öğretmeninden Mesaj:** “Bu süreçte sana rehberlik edeceğim. Elinden gelenin en iyisini yapmaya çalışman yeterli. Kod yazmayı birlikte öğrenecek, birlikte başaracağız!”

Etkinlik Yaprakları – Programlama Öğrenme Süreci

Etkinlik Yaprağı 1 – Başlangıç Seviyesi

Konu: Değişkenler ve Ekrana Yazdırma

Amaç: Python’da değişken tanımlamayı ve print() komutu ile veri yazdırmayı öğrenmek.

Etkinlik:

1. Aşağıdaki kodu inceleyin. Bu kod ne yapar? Açıklayın.

```
isim = "Ali"  
yas = 13  
print("Merhaba", isim)  
print("Yaşınız:", yas)
```

2. Kendi adınızı ve yaşınızı değişkene atayarak kodu yeniden yazın.

3. Ek olarak bir “şehir” değişkeni oluşturun ve “Ben [isim], [şehir]’de yaşıyorum.” yazdırın.

Etkinlik Yaprağı 2 – Orta Seviye

Konu: Koşullu İfadeler (if-else)

Amaç: Karar yapılarıyla basit kontrol mekanizmaları oluşturabilmek.

Etkinlik:

1. Kullanıcıdan bir sayı girmesini isteyin.

2. Sayı çift mi tek mi olduğunu ekrana yazdıran bir program yazın:

```
sayi = int(input("Bir sayı girin: "))  
if sayi % 2 == 0:  
    print("Bu sayı çift.")  
else:  
    print("Bu sayı tek.")
```

3. Programınızı geliştirerek sayının negatif mi pozitif mi olduğunu da yazdırın.



Etkinlik Yaprağı 3 – İleri Seviye

Konu: Döngüler ve Liste İşlemleri

Amaç: Liste oluşturma ve for döngüsüyle işlemler yapabilme.

Etkinlik:

1. Aşağıdaki kod ne yapar? Tahmin edin, sonra çalıştırıp kontrol edin:

```
meyveler = ["elma", "muz", "çilek"]  
for meyve in meyveler:  
    print("Sevdiğim meyve:", meyve)
```

2. Kendi “sevdiğiniz dersler” listesini oluşturun ve her ders için “Bu ders eğlenceli!” yazdırın.

3. Bonus: Bir sayı listesi oluşturun. Bu listedeki sayıların toplamını ekrana yazdıran bir kod yazın.

EK-9. Etik Kurul İzni

T.C.
ATATÜRK ÜNİVERSİTESİ
SOSYAL VE BEŞERİ BİLİMLER ETİK KURULU
Eğitim Bilimleri Birim Etik Kurul Başkanlığı
PROJE DEĞERLENDİRME FORMU

Projenin/Çalışmanın Adı: Danışmanlığını Prof.Dr. Sevdâ KÜÇÜK'ün yürüttüğü yüksek lisans öğrencisi Volkan KAYA'nın "Ortaokul Programlama Eğitiminde Üretken Yapay Zeka Destekli Bir Öğrenme Modülünün Tasarlanması, Uygulanması ve Değerlendirilmesi" isimli yüksek lisans tez çalışması ile ilgili Etik Kurul uygunluk onay belgesi talep edilmesi.

Gönderildiği Tarih: 30.12.2024

Teslim Tarihi: 30.12.2024

Değerlendirenin Adı Soyadı: Sosyal ve Beşeri Bilimler Etik Kurulu Başkanlığı Adına

Prof. Dr. Ufuk ŞİMŞEK
Etik Kurul Başkanı
e-imza

Bu Yüksek Lisans Tez uygundur: +

Gerekçe/Açıklama: Yapılacak çalışmada yaklaşım ve yöntemler dikkate alınarak, konu ile ilgili çalışmaların gerçekleştirilmesinde etik ve bilimsel açıdan herhangi bir sakınca bulunmamaktadır.

Bu Yüksek Lisans Tez uygun değildir: ☐
Gerekçe/Açıklama



EK-10. İl Milli Eğitim Müdürlüğü Uygulama İzin Belgesi



Muştakbaba Ortaokulu Müdürlüğüne



Başvuru No: MEB.TT.2025.020748

Uygulama Yapılacak MEB Teşkilatının Kurum Kodu: 714933

T.C. Kimlik No: [REDACTED]

Adı Soyadı: VOLKAN KAYA

Araştırmanın Adı: Ortaokul Programlama Eğitiminde Üretken Yapay Zeka Destekli Bir Öğrenme Modülünün Tasarlanması, Uygulanır ve Değerlendirilmesi

Araştırmanın Niteliği: Yüksek Lisans Tezi

Araştırmanın Örneklem / Çalışma Grubu: Öğrenci

Uygulama Yapılacak MEB Teşkilatı: Muştakbaba Ortaokulu

Uygulama Yapılacak Birim: Ortaokul

Uygulama Yapılacak İl: BİTLİS

Veri Toplama Aracının Başlığı: Bilgi İşlemsel Düşünme Öz Değerlendirme Ölçeği, Bilgi İşlemsel Düşünme Bebras Testi, Programlamaya Yönelik Tutum Ölçeği, Öğrenci Görüşme Formu, Programlama Başarı Testi

Araştırma Uygulama İzninin Kabul Tarihi: 14.03.2025

Araştırmanın Uygulama İzninin Bitiş Tarihi: 14.03.2026

Yukarıda kimliği yazılı araştırmacı "Araştırma Uygulama İzinleri Genelgesine (2024/41)" göre belirtilen kapsamda araştırmasını yapmayı taahhüt etmiştir. Araştırmacının bilgi ve belgelerinin uygunluğu kontrol edilmiş olup araştırma uygulama izni **BİTLİS İl Milli Eğitim Müdürlüğü** tarafından onaylanmıştır.

NOT: Okul/kurum yöneticileri tarafından "Araştırma Uygulama İzni" belgesinin ve veri toplama araçlarının (araçlardaki maddelerinin) modülde yer alan belge ve araçlarla aynı olduğu kontrol edilmelidir. Belgeler aynı olmadığı durumda araştırma uygulama izni verilmeyecektir.

* Başvuru detayını görüntülemek ve belgeyi doğrulamak için '<https://arastirmaizninleri.meb.gov.tr/belge-dogrula>' bağlantısını kullanınız.

- Araştırma Uygulama İzinleri Başvuru ve Değerlendirme Sistemi -

EK 11. Ebeveyn Bilgilendirilmiş Gönüllü Onam Formu



EBEVEYN BİLGİLENDİRİLMİŞ GÖNÜLLÜ ONAM FORMU

Sayın Velimiz,

“Ortaokul Programlama Eğitiminde Üretken Yapay Zeka Destekli Bir Öğrenme Modülünün Tasarlanması, Uygulanması ve Değerlendirilmesi” başlıklı çalışma, Volkan Kaya tarafından yürütülmektedir. Bu araştırma, öğrencilerin Python programlama dilini öğrenirken üretken yapay zeka destekli bir öğrenme modülünün etkilerini incelemek amacıyla yapılmaktadır.

Araştırma kapsamında, öğrencinizden bazı testler ve anketleri doldurması istenecektir. Bu test ve anketler, öğrencilerin programlama becerilerini, bilgi işlemsel düşünme yeteneklerini ve programlama derslerine yönelik tutumlarını değerlendirmek için hazırlanmıştır. Çalışma öncesinde ve sonrasında bu değerlendirme araçları kullanılacaktır. Uygulamanın toplam süresi yaklaşık 30-40 dakikadır.

Bu çalışmaya katılım tamamen gönüllülük esasına dayanmaktadır. Öğrenciniz çalışmaya katılmayı kabul ettikten sonra, dilediği zaman çalışmadan ayrılma hakkına sahiptir. Araştırma sırasında toplanan tüm bilgiler yalnızca bilimsel amaçlarla kullanılacak ve öğrencinizin kimlik bilgileri kesinlikle gizli tutulacaktır. Araştırma sonuçları, kimlik belirtici herhangi bir bilgi içermeyecektir.

Çalışmanın herhangi bir aşamasında sorularınız veya endişeleriniz olursa, aşağıda iletişim bilgileri yer alan araştırmacıya ulaşabilirsiniz.

Çocuğunuzun bu çalışmaya katılmasına izin veriyorsanız, lütfen aşağıdaki alanları doldurup imzalayınız.

Sorumlu Araştırmacı:

Volkan Kaya

Telefon Numarası: [REDACTED]

E-posta: [REDACTED]

Velinin Adı Soyadı:

İmza:

EK 12. Çocuk Bilgilendirilmiş Gönüllü Onam Formu



ÇOCUK BİLGİLENDİRİLMİŞ GÖNÜLLÜ ONAM FORMU

Merhaba,

“Ortaokul Programlama Eğitiminde Üretken Yapay Zeka Destekli Bir Öğrenme Modülünün Tasarlanması, Uygulanması ve Değerlendirilmesi” adlı bu çalışma, Volkan Kaya tarafından yapılmaktadır. Bu çalışmada, Python programlama dilini öğrenmenizi kolaylaştıracak bir modül kullanılacak ve bunun nasıl bir etki yaptığını anlamaya çalışacağız.

Bu çalışmaya katılmanız tamamen gönüllülük esasına dayanmaktadır. İsterseniz çalışmaya katılabilir, katılmayı kabul ettikten sonra bile istediğiniz zaman çalışmadan ayrılabilirsiniz. Çalışmaya katıldığınızda, sizden bazı testler ve anketler doldurmanız istenecek. Bunlar programlama ile ilgili becerilerinizi ve düşüncelerinizi ölçmek içindir.

Test ve anketlerden elde edilen bilgiler yalnızca bu çalışma için kullanılacaktır. Hiçbir yerde adınız veya kimliğinizle ilgili bir bilgi paylaşılmayacaktır. Sizin bilgileriniz gizli tutulacak ve sadece bilimsel amaçlarla kullanılacaktır.

Test ve anketleri tamamlamanız toplamda yaklaşık 45-60 dakikanızı alacaktır. Çalışma sırasında herhangi bir sorunuz olursa ya da bir şey anlamazsanız, yüz yüze veya aşağıdaki iletişim bilgileriyle bana ulaşabilirsiniz.

Çalışmamıza katkı sağladığınız için şimdiden teşekkür ederim!

Sorumlu Araştırmacı:

Volkan Kaya

Telefon Numarası: [REDACTED]

E-posta: [REDACTED]

Öğrenci Adı Soyadı:

İmza:

EK 13. Ölçek Kullanım İzinleri

Bilgi İşlemisel Düşünme Öz Değerlendirme Ölçeği Kullanım İzni Talebi

Gelen Kutusu x

x 0 0



Volkan Kaya

26 Kas Sal 10:05 (7 gün önce)

☆ 😊 ↩ ⋮

Sayın Veysel Karani Ceylan ve Ruken Akar Vural,

Ben Volkan Kaya, Erzurum Atatürk Üniversitesi Eğitim Bilimleri Enstitüsü'nde yüksek lisans öğrencisiyim. Tez çalışmamda "Ortaokul Programlama Eğitiminde Üretken Yapay Zeka Destekli Bir Öğrenme Modülünün Tasarlanması, Uygulanması ve Değerlendirilmesi" konusunu araştırmaktayım. Ayrıca, Bilişim Teknolojileri öğretmeni olarak çalışmaktayım.

Tez çalışmam kapsamında öğrencilerin bilgi işlemisel düşünme becerilerini değerlendirmek amacıyla sizin tarafınızdan geliştirilmiş olan "Bilgi İşlemisel Düşünme Öz Değerlendirme Ölçeği"ni kullanmak istiyorum. Çalışmamda ölçeğin etik kurallara çerçevesinde ve yalnızca bilimsel amaçlarla kullanılacağını garanti ederim. Ölçeğe ilgili uygun akademik atıfları yapacağımı ve kaynak gösterimini sağlayacağımı taahhüt ederim.

İzniniz doğrultusunda, ölçeği tez çalışmamda kullanmak ve katılımcılar üzerinde uygulamak istiyorum. Çalışmamın sonuçlarını sizinle paylaşmaktan memnuniyet duyuyorum.

İzin talebimle ilgili olumlu veya olumsuz geri dönüşlerinizi bekliyorum. Gerekli ek bilgi veya belgelere ihtiyaç duymazsanız halinde, sizinle iletişime geçmekten mutluluk duyuyorum.

Şimdiden desteğiniz için teşekkür eder, iyi çalışmalar dilerim.

Saygılarımla,
Volkan KAYA
Yüksek Lisans Öğrencisi
Erzurum Atatürk Üniversitesi
E-posta: [Redacted]
Telefon: [Redacted]



veysel ceylan

Alıcı: ben

26 Kas Sal 10:50 (7 gün önce)

☆ 😊 ↩ ⋮

Merhaba volkan hocam, öncelikle çalışmanızda kolaylıklar dilerim. Elbette ölçeğinizi yürüteceğiniz çalışmada kullanmanız için rıza göstermekteyiz. Ölçeğin tam halini ve kullanımı hakkında detayları aşağıda ki linkte bulabilirsiniz.

İyi çalışmalar dilerim.

Veysel Ceylan

Olcek izni

Gelen Kutusu x



Gönül ALTAY

Alıcı: ben

Merhaba hocam

İlgili ölcek ektedir. Başarılar dilerim.

Basari testi C# diline gore gelistirildi. Ortaokulda bu testi kullanabilecek misiniz?

Saygılarımla

Bir ek • Gmail tarafından tarandı



← Yanıtla

→ Yönlendir



Programlamaya Yönelik Tutum Ölçeği ve Programlama Yönelik Başarı Testi Kullanım İzni Talebi

0 0



Volkan Kaya

Alıcı: tarik.kisla, tarik.kis

26 Kas Sal 09:52 (7 gün önce)

☆ 😊 ↩ ⋮

Sayın Tarık Kışla,

Ben Volkan KAYA, Erzurum Atatürk Üniversitesi Eğitim Bilimleri Enstitüsü'nde yüksek lisans öğrencisiyim. Tez çalışmamda "Ortaokul Programlama Eğitiminde Üretken Yapay Zeka Destekli Bir Öğrenme Modülünün Tasarlanması, Uygulanması ve Değerlendirilmesi" konusunu araştırmaktayım. Aynı zamanda bir Bilişim Teknolojileri öğretmeni olarak çalışıyorum.

Tez çalışmam kapsamında öğrencilerin programlamaya yönelik tutumlarını ölçmek amacıyla sizin tarafınızdan geliştirilen "Programlamaya Yönelik Tutum Ölçeği" ve "Programlama Temelleri Dersine Yönelik Başarı Testi"ni kullanmak istiyorum. Çalışmamda ölçeğin etik kurallara çerçevesinde ve yalnızca bilimsel amaçlarla kullanılacağını belirtmek isterim. Ayrıca, ölçeğe ilgili akademik atıfları doğru bir şekilde yapacağımı ve gerekli kaynak gösterimlerini sağlayacağımı garanti ederim.

İzniniz doğrultusunda, ölçeği tez çalışmamda kullanmak ve katılımcılarım üzerinde uygulamak istiyorum. Çalışma sonuçlarını sizinle paylaşmaktan da mutluluk duyuyorum.

İzin talebimle ilgili olumlu veya olumsuz geri dönüşlerinizi bekliyorum. Gerekli ek bilgi veya belgelere ihtiyaç duymazsanız halinde, sizinle iletişime geçmekten memnuniyet duyuyorum.

Şimdiden yardımlarınız için teşekkür eder, iyi çalışmalar dilerim.

Saygılarımla,
Volkan KAYA
Yüksek Lisans Öğrencisi
Erzurum Atatürk Üniversitesi
E-posta: [Redacted]
Telefon: [Redacted]



ÖZ GEÇMİŞ

Volkan KAYA 1987 yılında Bitlis ilinin Merkez ilçesinde doğmuştur. İlkokul, ortaokul ve lise eğitimini burada tamamladıktan sonra 2013 yılında Afyon Kocatepe Üniversitesi Bilgisayar ve Öğretim Teknolojileri Öğretmenliği bölümüne başlamıştır. 2018 yılında Bitlis ili Merkez ilçesinde bilişim teknolojileri öğretmeni olarak çalışmaya başlamıştır. Mesleği boyunca Tübitak bölge ve Türkiye derecesi, Teknofest Türkiye derecesi ve çeşitli robot yarışmalarında dereceler elde etmiştir. Yaptığı projelerden 1 patent, 3 faydalı modeli ve 2 tasarım tescili bulunmaktadır. Ayrıca 2023 yılında Bitlis ilinde yılın öğretmeni seçilmiştir. Halen Bitlis ilinde bilim ve sanat merkezinde bilişim teknolojileri öğretmeni olarak çalışmaktadır.

