

Authentic programming activities: The effect of students on computational thinking and programming self-efficacy beliefs

Mücahit Öztürk, Aksaray University, Faculty of Education, Department of Computer and Instructional Technologies, ORCID ID: 0000-0003-4293-9086

Abstract

In this study, the effect of Introduction to programming course designed with authentic learning activities on students' computational thinking and programming self-efficacy beliefs was examined. This study is a mixed method research in which quantitative and qualitative data are used together. The study group consisted of 17 students studying at a state university, a vocational school, and the electronics and automation department. Before and after the study, the students completed the computer thinking scale and the programming self-efficacy beliefs scale. At the end of the study, students were interviewed to evaluate their development in Computational thinking and Programming self-efficacy beliefs. The results showed that authentic programming activities significantly improved Creativity, Algorithmic thinking, Critical thinking and the overall scores of Computational thinking. Students associated the development of computational thinking skills with engaging activities, continuous support, daily life problems, and learning to code. It was revealed that students' programming self-efficacy beliefs developed significantly. In future research, different activities can be planned to encourage students to do it repeat at home. Students can be crossed between groups each week to give them the opportunity to work with different people.

Keywords: *Authentic learning, computational thinking, programming learning, self-efficacy*



Inonu University
Journal of the Faculty of
Education
Vol 22, No 2, 2021
pp. 1611-1640
DOI:10.17679/inuefd.773764

Article type:
Research article

Received : 25.07.2020
Accepted : 26.08.2021

Suggested Citation

Öztürk, M. (2021). Authentic programming activities: The effect of students on computational thinking and programming self-efficacy beliefs, *Inonu University Journal of the Faculty of Education*, 22(2), 1611-1640. DOI: 10.17679/inuefd.773764

EXTENDED ABSTRACT

Introduction

Nowadays, students are expected to be able to solve problems and produce products by using information and communication technologies effectively. In this context, it is important that students have computer thinking skills in relation to 21st century skills (Wong & Cheung, 2018). Wing (2006) defines Computational thinking (CT) as “solving problems, designing systems, and understanding human behavior by drawing on the concepts fundamental to computer science”. Later, he developed this definition and described CT as the process of formulating problems and solutions using information processing agents (Wing, 2011). Programming is a way of solving problems in the digital world and applying them by creating creative ideas (Nouri et al., 2019). It is thought that CT, which is a form of problem solving using information processing agents, can develop with programming learning (Garneli & Chorianopoulos, 2017). Wong and Cheung (2018) found that CT and programming learning are related, and suggested further studies on how CT can improve with programming learning. Traditional programming education is knowledge-oriented and focuses on developing abstract knowledge. This situation makes it difficult for students to develop programming skills even though they have programming knowledge. With authentic learning experiences, students can improve their programming skills by associating abstract knowledge with authentic contexts (Peng et al., 2017). The fact that programming teaching is mostly done away from the authentic context affects students' learning negatively (Resnick et al., 2009). In this context, the realization of programming teaching with authentic learning activities such as real life problems and collaborative studies can contribute to the development of students' programming skills and self-efficacy (Kong et al., 2018; Peng et al., 2017). The development of students' programming skills contributes positively to the development of their CT skills (Rodríguez-Martínez et al., 2019; Garneli & Chorianopoulos, 2017).

Purpose

The aim of this study is to investigate the effect of programming course designed with AL activities on students' computational thinking and programming self-efficacy. The problems of the study are listed below.

1. Do authentic programming activities significantly improve students' computational thinking?
2. Do authentic programming activities significantly improve students' programming self-efficacy?

Method

In this study, mixed method was used in which quantitative and qualitative data were collected together. The study was carried out with the explanatory sequential design. The CT scale developed by Korkmaz et al. (2017) was used to evaluate the development of students' CTs. Programming self-efficacy scale, developed by Ramalingam and Wiedenbeck (1998), adapted to Turkish by Altun and Mazman (2012), was used to examine the change in students' self-efficacy beliefs. At the end of the study, students were interviewed to evaluate their development in CT and programming self-efficacy. For this purpose, semi-structured interview form was prepared.

Findings

It was determined that students' creativity, algorithmic thinking and critical thinking developed significantly compared to the pre-study period. Although cooperativity and problem solving scores increased compared to the pre-study, there was no significant improvement. CT overall average score was significantly increased compared to the pre-study. In the interviews with the students, they mostly explained the development of creativity skills with "Giving sufficient time for coding", "Developing self-confidence as they solve problems" and "Working with problems". They often referred to the development of algorithmic thinking as "Sorting

tasks from easy to difficult”, “Learning to coding”. The development of critical thinking skills has been associated with “Giving you the opportunity to work on difficult problems”, “Providing the necessary environment for evaluating each course performance”, “Solving problems with my group mate”. The students stated the increase in the level of cooperation as “Having enjoyable working with my friends”, “Providing a comfortable environment”, “Facilitates working with difficult tasks”. They explained the development of problem solving with “The problems given in the course are interesting”, “Instructor and peer support”, “Cooperative work”. Students explained the general development of CT skills with “Carrying out interesting activities”, “Opportunity to get continuous support at home and at school”, “Working with daily life examples makes learning easier”, “Each course is linked to the topics covered in the previous week”, “Learning coding”. It was seen that some students' skill scores decreased or did not develop according to the pre-study. The students stated the situation in problem solving skills as “Sometimes, not being able to solve one problem negatively affects other solutions”. They explained the situation in cooperative skills as “Negative effect of poor performance of group mate”, “Willing to work individually”. While the simple programming tasks and general averages of students increased from low to high level, complex programming tasks increased from low to medium level.

Discussion & Conclusion

In this study, students were given the opportunity to work on authentic problems. Each group was given a different problem situation and the students were expected to find different solutions. Eddles-Hirsch et al. (2019) revealed that authentic activities such as research, finding solutions to problems, reflecting and evaluating can improve students' creativity skills. The instructor presented authentic programming tasks from simple to complex. Students were given the opportunity to work on different code samples each week. Malik et al. (2019) revealed that in the introduction to programming course, students' planning the solution steps of the program before developing the coding and working on authentic

problems developed algorithmic thinking more than traditional programming education. The students had completed their authentic programming tasks together with their group mates. Hwang et al. (2012) revealed that cooperative programming tasks improve students' performance positively. The students had studied authentic learning activities and complex programming tasks. Wang (2017) found that giving students the opportunity to work with projects and peer assessment significantly improved their programming skills and critical thinking compared to traditional programming teaching. The instructor gave the students interesting programming tasks that they might encounter in their daily life. Ge et al. (2006) revealed that students' working with open-ended and realistic problems contributed to their solving in the programming course. The students listed the authentic programming tasks from simple to complex. The instructor aims to increase students' confidence in themselves by completing simple programming tasks. Abdunabi et al. (2019) states that working with real-life projects may contribute positively to the programming self-efficacy.

Otantik programlama etkinlikleri: Öğrencilerin bilgi işlemsel düşünme ve programlama öz-yeterlilik inançlarına etkisi

Mücahit Öztürk, Aksaray Üniversitesi, Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü, ORCID ID: 0000-0003-4293-9086

Öz

Bu araştırmada otantik öğrenme etkinlikleriyle tasarlanan programlamaya giriş dersinin öğrencilerin bilgi işlemsel düşünme ve programlama öz-yeterlilik inançlarına etkisi incelenmiştir. Bu çalışma nicel ve nitel verilerin birlikte kullanıldığı karma yöntem araştırmadır. Araştırma grubu bir devlet üniversitesi, meslek yüksekokulu, Elektronik ve Otomasyon bölümünde öğrenim gören 17 öğrenciden oluşmaktadır. Araştırma öncesinde ve sonrasında öğrenciler bilgi işlemsel düşünme ölçeği ve programlama öz-yeterlilik inançları ölçeğini doldurmuştur. Araştırmanın sonunda öğrencilerin bilgi işlemsel düşünme ve programlama öz-yeterlilik gelişmelerini değerlendirmek için öğrencilerle görüşme yapılmıştır. Bulgular otantik programlama etkinliklerinin yaratıcılık, algoritmik düşünme, eleştirel düşünme ve bilgi işlemsel düşünme genel düzeylerini anlamlı şekilde geliştirebildiğini göstermiştir. Öğrencilerin bilgi işlemsel düşünmenin problem çözme ve işbirliklilik gibi bazı alt faktörlerinde anlamlı gelişmeler olmamıştır. Öğrenciler bilgi işlemsel düşünme becerilerinin gelişimini ilgi çekici etkinlikler yapılması, sürekli destek, günlük yaşam problemleri ile çalışma ve kodlama öğrenme ile ilişkilendirmiştir. Öğrencilerin programlama öz-yeterlilik inançlarının anlamlı şekilde gelişebildiği ortaya çıkmıştır. Gelecekteki araştırmalarda farklı eğitim seviyelerinde ve daha geniş örneklerle çalışılabilir. Öğrencilerin evde tekrar yapmalarını teşvik edecek farklı etkinlikler planlanabilir. Diğer taraftan bazı öğrencilerin grup içindeki performans farklılıklarından olumsuz etkilendiği ortaya çıkmıştır. Bu çerçevede öğrenciler her hafta gruplar arasında çaprazlanarak farklı öğrencilerle çalışma fırsatı bulabilir.

Anahtar Kelimeler: Otantik öğrenme, bilgi işlemsel düşünme, programlama öğrenme, öz-yeterlilik



İnönü Üniversitesi
Eğitim Fakültesi Dergisi
Cilt 22, Sayı 2, 2021
ss. 1611-1640
DOI:10.17679/inuefd.773764

Makale türü:
Araştırma makalesi

Gönderim Tarihi :
25.07.2020
Kabul Tarihi:
26.08.2021

Önerilen Atıf

Öztürk, M. (2021). Otantik programlama etkinlikleri: Öğrencilerin bilgi işlemsel düşünme ve programlama öz-yeterlilik inançlarına etkisi. *İnönü Üniversitesi Eğitim Fakültesi Dergisi*, 22(2), 1611-1640. DOI: 10.17679/inuefd.773764

GİRİŞ

Bilgi ve iletişim teknolojilerinin aktif kullanımı tüm meslek alanlarında öğrencilerden beklenen temel bir yeterlilik haline gelmiştir. Öğrencilerden 21. yy becerileri çerçevesinde eğitim yaşantılarında teknolojiyi kendi alanlarında karşılaşılabilecekleri problemlerin çözümünde kullanabilmesi beklenmektedir. Gelişmiş ve gelişmekte olan ülkeler temel eğitimden yükseköğretime kadar tüm eğitim kademelerinin müfredatlarına bilgi teknolojilerinin etkin kullanımı, programlama, yapay zeka ve robotik gibi konuları entegre etmektedir. Öğrencilerin problem çözme ve yaratıcı düşünme sürecinde bilgi teknolojilerinden yararlanmalarında bilgi işlemsel düşünme (BİD) becerilerine sahip olmalarının önemli olduğu düşünülmektedir (Wong ve Cheung, 2018). Literatür incelendiğinde BİD' in tanımına yönelik bir fikir birliği olmasa da en çok kullanılan tanımlamalardan biri olarak Wing (2006), BİD'i "temel bilgisayar bilimi kavramlarını kullanarak problem çözme, sistem tasarlama ve insan davranışlarını anlamak" şeklinde ifade etmiştir. Daha sonra bu tanımlama geliştirilerek BİD'i bilgi işlem ajanlarını kullanarak problemleri ve çözümleri formüle etme süreci olarak ifade edilmiştir (Wing, 2011). Ulusal Bilim Vakfı (NSF), Uluslararası Eğitimde Teknoloji Topluluğu (ISTE) ve Bilgisayar Bilimleri Öğretmenleri Derneği' ne (CSTA) göre BİD aşağıdaki becerileri içeren problem çözme sürecidir (ISTE, 2011):

- Formüle etmek: Bilgisayar vb. araçları, problemleri çözmek için formüle etmek.
- Organize ve analiz etmek: Verileri mantıksal olarak organize ve analiz etmek.
- Sunmak: Verileri modelleme ve simülasyonlar ile sunmak.
- Otomatikleştirmek: Algoritmik düşünme yoluyla çözümleri otomatikleştirmek.
- Tanımlamak, Analiz etmek, Uygulamak: Adımların ve kaynakların etkili ve verimli şekilde birleştirilmesi için olası çözümleri tanımlamak, analiz etmek ve uygulamak.
- Genelleştirme ve Transfer etmek: Problem çözme sürecinin genelleştirilmek ve çok çeşitli problemlere aktarmak.

Ayrıca İSTE, BİD’i yaratıcı düşünme, algoritmik düşünme, eleştirel düşünme, işbirlikli öğrenme ve iletişim becerilerini yansıtan bir süreç olarak açıklamaktadır (Korkmaz ve ark., 2017). BİD’in tanımı ve öğrenme ortamında uygulanabilirliğiyle ilgili bazı belirsizliklerin olması BİD’in öğrenme ortamındaki gelişimine ilişkin çalışmalar yapılmasını gerektirmektedir (Shute, Sun ve Asbell-Clarke, 2017). Román-González ve ark. (2017), öğrencilerin BİD gelişimi için programlama mantığını öğrenmeleri gerektiğini ifade etmektedir. Problem çözme ve yaratıcı fikirler üreterek uygulama yapma süreci olan programlamanın (Nouri ve ark., 2019) bilgi işlem ajanlarını kullanarak bir problem çözme şekli olan BİD’i geliştirebileceği düşünülmektedir (Garneli ve Chorianopoulos, 2018). Wong ve Cheung (2018) çalışmasında BİD ve programlama öğrenmenin ilişkili olduğunu ortaya çıkarmış ve BİD’in programlama öğrenimiyle nasıl geliştirebileceğine ilişkin daha fazla çalışma yapılması gerektiğini ifade etmiştir. Ching ve ark. (2018) göre programlama, geniş kapsamlı problem çözme süreci olan BİD’i geliştirmek için önemli bir rol oynamaktadır. Programlama dersinin genellikle gerçek yaşam bağlamından uzak şekilde işlenmesi öğrencilerin öğrenmesini olumsuz yönde etkilemektedir (Rodríguez-Martínez ve ark., 2019). Geleneksel programlama eğitimi bilgi odaklıdır ve soyut bilgiyi geliştirmeye çalışmaktadır. Bu durum öğrencilerin programlama bilgisine sahip olsa da programlama becerilerini geliştirmesini zorlaştırmaktadır. Otantik öğrenme deneyimleri ile öğrenciler soyut bilgiyi otantik bağlamlarla ilişkilendirerek programlama becerilerini geliştirebilir (Peng ve ark., 2017).

Otantik Öğrenme

Öğrencilerin otantik problemler üzerinde çalışmasının anlamlı öğrenmelerini geliştirebileceği düşünülmektedir. Otantik öğrenme öğrencilere gerçek dünya ile ilişkili bağlamlar sunar (Herrington ve ark., 2014). Otantik öğrenme ortamları iyi tanımlanmamış bir görevin belirli bir süre boyunca ve farklı açılardan incelenmesini gerektirir (Frydenberg, 2015). Otantik öğrenme etkinlikleri öğrencileri gerçek dünya problemlerini, önceki deneyimleri ile mevcut bilgilerini birleştirerek çözebilmelerini sağlar (Roach ve ark., 2018). Otantik öğrenme,

problem tabanlı öğrenme, proje tabanlı öğrenme, rol oynama ve işbirlikli öğrenme gibi farklı öğrenme etkinliklerini içermektedir. Otantik öğrenme ortamları bilginin gerçek hayatta nasıl kullanılacağına ilişkin bağlamlar sunarken öğrencilerin karmaşık görevler üzerinde çalışmasını sağlar. Ayrıca farklı uzmanlık seviyelerindeki öğrencilerin birlikte çalışmasına ve deneyimlerini paylaşmasına imkân tanımaktadır. Bu süreçte öğrencilerin öğrendiklerini yansıtmaları ve kendi öğrenme süreçlerini değerlendirmeleri beklenir (Shadiev ve ark., 2017). Herrington ve ark. (2014) göre otantik öğrenmenin öğeleri aşağıdaki gibi özetlenebilir:

Otantik bağlam: Bilginin gerçek hayatta nasıl kullanılacağını yansıtan özgün bağlamları içerir. Otantik bağlam süreç boyunca keşfedebilecek gerçek dünya durumlarını içeren karmaşık öğrenme ortamı sunmalıdır.

Otantik görevler: Öğrenme ortamı kötü tanımlanmış, alt kategorilere ayrılabilen ve birbirleriyle ilişkili problem durumları sunar. Öğrencileri araştırmaya yönlendirirken farklı çözüm yolları üretmeleri için teşvik eder.

Uzman performansına erişim ve modelleme: Öğrenciler problemleri çözerken bir uzman gibi düşünebilirken farklı uzmanlık seviyelerindeki öğrenci ya da sosyal çevreyle işbirliği yapabilmelidir. İnternet ve sosyal ağlar aracılığıyla sanal topluluklar oluşturularak öğrencilerin deneyimlerini paylaşabilmeleri sağlanabilir. Bu doğrultuda öğrenme sürecinin modellemesini yapabilir.

Çoklu roller ve perspektifler: Öğrenciler kendilerine verilen göreve farklı açılardan bakabilmelidir. Bir problem birden fazla yolla çözülebilmelidir.

Bilginin işbirlikli inşası: Öğrencilerin otantik problemler üzerinde çalışırken işbirliği yapmaları önemlidir. Verilen görevler bireysel olarak değil grupta birlikte tamamlanmalıdır.

Yansıtma: Öğrenciler otantik görevler üzerinde çalışarak kendi öğrenmelerini yansıtmalıdır. Öğrenme ortamında etkinlikler doğrusal bir şekilde yapılmaz. Öğrenci içeriğin herhangi bir kısmına dönebilmelidir. Bu süreçte performanslarını diğer öğrencilerle karşılaştırabilme fırsatı verilmelidir.

Ekleme: Öğrencilere güçlü ve zayıf yönleri hakkında akıl yürütmeleri ve düşüncelerini ifade etme fırsatı verilmelidir.

Koçluk ve destek: Otantik öğrenmede öğrenci tamamen kendi haline bırakılmamalıdır. Öğretmen bu süreçte öğrencilerine rehberlik yapmalı ve desteklemelidir. Bu süreçte daha yetenekli öğrenciler de grup arkadaşına gerekli desteği sağlamalıdır.

Otantik değerlendirme: Öğrenciler kendilerine verilen görevin başarıyla çözümüne ilişkin değerlendirme yapmalıdır. Öğrencilere performansını gösterme, izleme ve değerlendirme fırsatı verilmelidir. Değerlendirme faaliyeti öğrenme süreciyle bütünleştirilerek ve kriterler belirlenerek yapılmalıdır.

Bu öğeler otantik öğrenme ortamları tasarlamak için gerekli çerçeveyi sunmaktadır. Ayrıca son yıllarda otantik öğrenmeyi desteklemek için çevrimiçi ve mobil teknolojilerden yararlanılmaktadır (Chiu ve ark., 2018; Chin ve ark., 2017). Bu platformlar ile sanal topluluklar oluşturarak öğrencilerin bilgi ve deneyimlerini birbirleriyle paylaşmasına ve anlamlı öğrenmelerine katkı sağlanabilir.

Programlama öğretiminin de çoğunlukla gerçek yaşam bağlamından uzak şekilde yapılması öğrencilerin öğrenmesini olumsuz etkilemektedir (Resnick ve ark., 2009). Programlama dersinin gerçek yaşam problemleri ve işbirlikli çalışmalar gibi otantik öğrenme etkinlikleri ile yapılması öğrencilerin programlama becerilerinin ve öz-yeterliklerinin gelişmesine katkı sağlayabilir (Kong ve ark., 2018; Peng ve ark., 2017). Öğrencilerin programlama becerilerinin gelişimi onların BİD becerilerinin gelişimini olumlu yönde etkileyebilmektedir (Garneli ve Chorianopoulos, 2017; Rodríguez-Martínez ve ark., 2019).

İlgili Çalışmalar

Literatür incelendiğinde programlama öğretiminde BİD'in gelişimine ilişkin çeşitli çalışmalar bulunmaktadır. Rodríguez-Martínez ve ark. (2019) Scratch kullanımının öğrencilerin bilgi işlemsel düşünme becerilerine etkisini incelemiştir. 6.sınıf öğrencileriyle yaptığı çalışmada Scratch ile programlama öğretiminin öğrencilerin BİD becerilerini olumlu yönde

geliştirdiğini ortaya çıkarmıştır. Farklı örneklerde çalışmaların tekrarlanması önerilmiştir.

Wong ve Cheung (2018), programlama öğretiminin yaratıcı düşünme, eleştirel düşünme ve problem çözme üzerindeki etkisini incelemiştir. 358 ilkokul öğrencisiyle yapılan çalışmada anket ve görüşmelerle veriler toplanmıştır. Bulgular öğrencilerin programlama öğrenimiyle 21. yy becerilerinin gelişebileceğini düşündüklerini ortaya çıkarmıştır. Ayrıca BİD ile programlama öğreniminin ilişkili olabileceği ve gelecekteki araştırmalarda bu ilişkiyi ortaya koyabilecek çalışmalar yapılması gerektiği ifade edilmiştir.

Nouri ve ark. (2019), programlama öğretiminde öğrencilerin BİD becerilerinin gelişimine ilişkin öğretmenlerin görüşünü almıştır. Ortaöğretimde görev yapan 19 öğretmenle görüşmeler yapılarak tematik analiz yapılmıştır. Çalışmanın sonucuna öğretmenler programlama öğretimiyle öğrencilerin BİD becerilerinin geliştirilebileceğini ifade etmişlerdir. Lin ve ark. (2018), 158 öğrenci ile yaptıkları çalışmada görsel programlama etkinliklerinin öğrencilerin BİD becerilerini olumlu yönde geliştirdiğini ortaya çıkarmıştır. Perez-Marin ve ark. (2018), 132 ilkokul öğrencisiyle yaptığı çalışmasında Scratch ve metaforlar ile programlama eğitiminin öğrencilerin BİD becerilerini geliştirdiğini belirlemiştir.

Peng ve ark. (2017) ortaöğretimde öğrenim gören 29 öğrenciyle yaptığı çalışmasında otantik programlama etkinliklerinin öğrencilerin programlama öğrenimini kolaylaştırdığını ve dersteki motivasyonlarını artırdığını belirlemiştir. Castledine ve Chalmers (2011) otantik problem çözme etkinliklerinin öğrencilerin LEGO robotik öğrenmelerine etkisini incelemiştir. 6. sınıfta öğrenim gören 23 öğrenciyle yaptığı çalışmasında otantik problemler üzerinde çalışma ve tartışma yapmanın öğrencilerin programlama becerilerini geliştirebileceğini ortaya çıkarmıştır. Thomas ve ark. (2011) yüksekokulda öğrenim gören 12 öğrenciyle yaptığı çalışmasında programlama dersinde otantik tasarımların 21 yy. becerilerine etkisini değerlendirmiştir. Öğrenciler öğretim sürecinde gözlenmiş ve görüşmeler yapılmıştır. Çalışmanın sonunda otantik öğrenme etkinlikleriyle tasarlanan programlama dersinde öğrencilerin problem çözme ve yaratıcılık becerilerini geliştirebileceği ortaya çıkmıştır. Aşkar ve

Davenport (2009), 326 mühendislik öğrencisiyle yaptıkları çalışmalarında programlama öz-yeterliliklerinin bilgisayar becerileri ve deneyimleriyle anlamlı şekilde ilişkili olduğunu belirlemiştir.

Sonuç olarak öğrencilerin BİD becerilerinin gelişiminin programlama öğrenmeleriyle ilişkili olabileceği görülmüştür. Diğer taraftan otantik öğrenme etkinliklerinin öğrencilerin programlama öğrenmelerine olumlu etkileri olabileceği de söylenebilir. Bu araştırmada literatürdeki çalışmalardan farklı olarak otantik programlama etkinliklerinin öğrencilerin BİD becerilerinin gelişimi ile programlama öz-yeterliliklerine etkisi birlikte incelenmiştir. Bu bağlamda araştırmanın amacı otantik öğrenme etkinlikleriyle tasarlanan programlama dersinin öğrencilerin BİD ve programlama öz-yeterliliğine etkisini incelemektir. Bu amaç çerçevesinde aşağıdaki araştırma sorularına cevap aranacaktır:

1. Otantik programlama etkinlikleri öğrencilerin bilgi işlemsel düşünme becerilerini anlamlı şekilde geliştirir mi?
2. Otantik programlama etkinlikleri öğrencilerin programlama öz-yeterlilik inancını anlamlı şekilde geliştirir mi?

YÖNTEM

Araştırma Modeli

Bu çalışma nicel ve nitel verilerin birlikte kullanıldığı karma yöntem araştırmadır. Çalışma sıralı açıklayıcı tasarım ile yürütülmüştür. Sıralı açıklayıcı tasarımda önce nicel veriler analiz edilir sonra bu analizlere ilişkin nitel veriler toplanarak birlikte yorumlanır ve değerlendirilir (Creswell, 2012). Öğrencilerin bilgi işlemsel düşünme ve programlama öz-yeterliliğine ilişkin gelişimleri ölçeklerden elde edilen verilerle değerlendirilmiştir. Ayrıca öğrencilerin otantik programlama öğrenme sürecindeki gelişimlerini açıklamaları ve gerekçeler ortaya koymaları için görüşme yapılarak nitel veriler toplanmıştır. Bu çerçevede veri çeşitliliği

sağlanarak çalışma sonuçlarının daha güvenilir ve anlamlı şekilde yorumlanabilmesi amaçlanmıştır.

Araştırma Grubu

Bu çalışma Elektronik ve otomasyon bölümünde öğrenim gören 17 ön lisans öğrencisiyle yürütülmüştür. Öğrencilerin çoğunluğu Anadolu lisesi mezunu olup programlama deneyimi bulunmamaktadır (Anadolu Lisesi: 12, Meslek Lisesi: 4, İmam Hatip Lisesi:1). Meslek lisesinden gelen öğrenciler Elektronik (2) ve Bilgisayar (2) bölümlerinden mezun olmuştur. Ayrıca çalışma grubunda erkek (12) öğrencilerin sayısı, kız (5) öğrencilerin sayısından fazladır.

Veri Toplama Araçları

Bilgi işlemsel düşünme ölçeği: Öğrencilerin BİD becerilerindeki gelişimi değerlendirmek için Korkmaz ve ark. (2017) tarafından geliştirilen ölçekten yararlanılmıştır. Ölçek, 5 faktör ve 29 maddeden oluşan 5'li likert tipindedir. Ölçekten elde edilen puanlar 100'lük sisteme çevrilmiştir. Bu çevrimde Korkmaz ve ark. (2017)'nin geliştirdiği formülden yararlanılmıştır ($X_{\text{standard point}} = (X_{\text{ham point}} / \text{Ölçülen madde sayısı}) * 20$). Faktörlerden elde edilen puanlara karşılık gelen düzeyler 20-51: Düşük Düzey; 52-67: Orta Düzey; 68-100: Yüksek Düzey şeklinde yorumlanmıştır.

Programlama öz-yeterlik ölçeği: Öğrencilerin programlama öz-yeterlik inançlarındaki değişimi incelemek için Ramalingam ve Wiedenbeck (1998) tarafından geliştirilen Altun ve Mazman (2012) tarafından Türkçe'ye uyarlanan Programlamaya ilişkin öz yeterlilik algısı ölçeği kullanılmıştır. Basit programlama görevleri ve karmaşık programlama görevleri olmak üzere 2 faktörlüdür. Ölçek 7'li likert tipinde ve 9 maddeden oluşmaktadır.

Yarı yapılandırılmış görüşme formu: Çalışmanın sonunda öğrencilerin BİD ve programlama öz-yeterliklerindeki gelişimlerini değerlendirmek için tüm öğrencilerle görüşme yapılmıştır. Bu amaçla yarı yapılandırılmış görüşme formu hazırlanmıştır. Görüşmelerde öğrencilerin BİD ölçeği faktörleri ve programlama öz-yeterlilik ölçeği faktörlerindeki gelişme ya da gelişmeme durumlarına ilişkin olarak gerekçeler ifade etmesi beklenmiştir. Öğrencilerle ayrı ayrı

görülmüş ve görüşme sorularına verdikleri cevaplar görüşme formuna kaydedilmiştir. Yarı yapılandırılmış görüşme formu iki alan uzmanının görüşü alınarak hazırlanmıştır. Aday görüşme formu uzmanlara e-posta ile gönderilerek geribildirim alınmıştır. Geribildirimler doğrultusunda düzeltmeler yapılmış ve tekrar uzmanlara gönderilerek yarı yapılandırılmış görüşme formuna son hali verilmiştir.

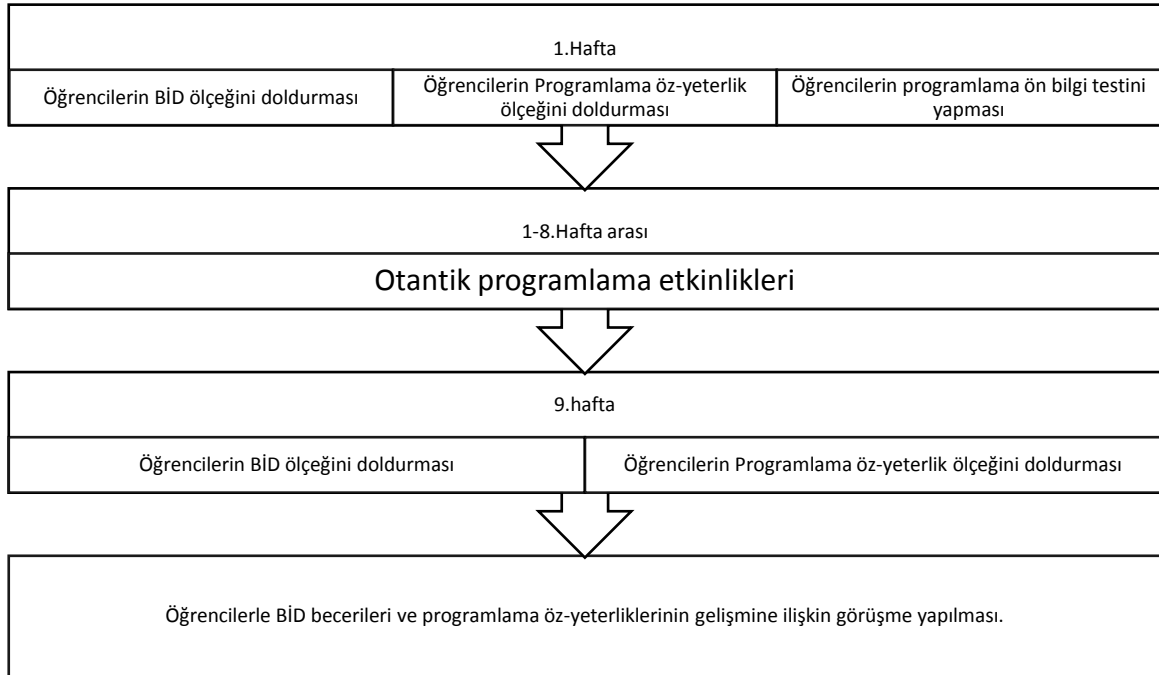
Veri Analizi

Öğrencilerin BİD becerileri ve programlama öz-yeterlik düzeylerine ilişkin toplanan veriler için SPSS 20 programı ile betimsel analizler yapılırken ve çalışma öncesi ve sonrasındaki değişime ilişkin olarak eşleştirilmiş t-testinden yararlanılmıştır. Ayrıca elde edilen veriler normallik testi (Shapiro-Wilk) yapılmış ve verilerin normal dağıldığı görülmüştür ($p=0.221$). Öğrencilerin tamamıyla BİD becerileri ve öz-yeterlik düzeylerindeki gelişim durumlarına ilişkin yapılan görüşmelerden elde edilen veriler içerik analizi yapılmıştır. İçerik analizinde araştırma sürecinde toplanan verilere ilişkin kavramlar ve ilişkiler ortaya çıkarılabilmektedir. Bu çerçevede birbiriyle ilişkili olan ya da benzeyen durumlar aynı tema altında gruplandırılarak yorumlanabilmektedir (Yıldırım & Şimşek, 2018). BİD ölçeği ve programlama öz-yeterlik ölçeğindeki faktörler içerik analizinin temalarını oluşturmaktadır. Öğrencilerin ilgili temadaki gelişim durumlarına ilişkin açıklama yapması ve gerekçeler üretmesi beklenmiştir. İçerik analizi için Nvivo 10 bilgisayar destekli nitel veri analizi programından yararlanılmıştır. Bu program ile temalara ilişkin öğrencilerin verdiği cevaplardan ortak kodlar ortaya çıkarılmıştır. Analiz sürecinde araştırmacının yanında bir uzmandan destek alınmıştır. Araştırmacı ve alan uzmanı temalara ilişkin kodları belirledikten sonra birbirleriyle karşılaştırmıştır. Çoğunlukla benzer kodların elde edildiği analiz sonuçlarında birbiriyle eş anlamlı olan farklı kodlar ile ortaya çıkarılan ek kodlar üzerinde tartışılarak kodlamalara son hali verilmiştir. Miles ve Huberman (1994)'ün güvenilirlik formülü ($\text{güvenilirlik} = \text{fikir birliği} / (\text{fikir birliği} + \text{anlaşmazlık})$) kullanılarak veri analizinin güvenilirliği % 83 olarak bulunmuştur.

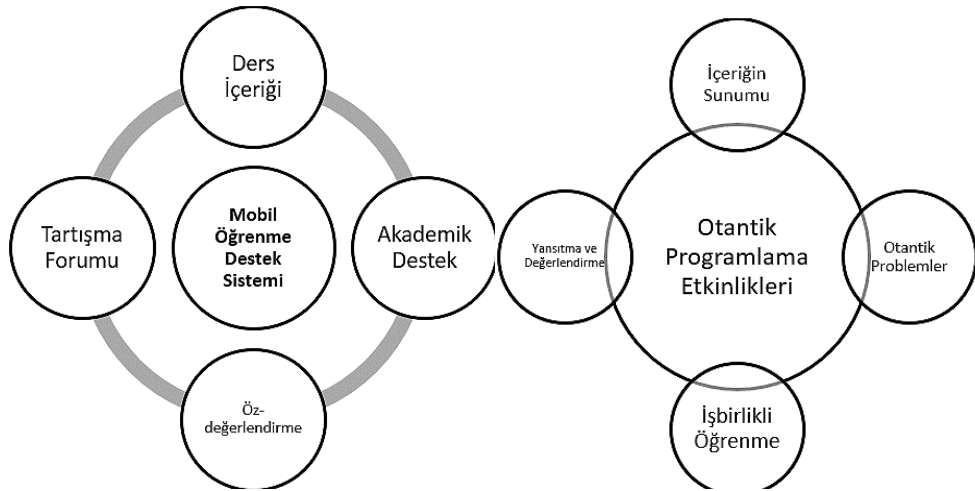
Süreç

Çalışma 8 hafta boyunca Programlamaya giriş dersinde bilgisayar laboratuvarında otantik öğrenme etkinlikleri ile yürütülmüştür. Programlamaya giriş dersi haftada 4 saat olarak işlenmektedir. C++ programlama dili ile çalışılmıştır. Öğrenciler çalışma öncesinde ve sonrasında BİD ölçeği ve programlama öz-yeterlik ölçeğini doldurmuştur. Ayrıca öğrencilere programlama bilgi testi yapılmıştır. Programlama bilgi testi iki öğretim elemanı tarafından hazırlanmıştır. Öğrencilerin çoğunluğunun programlama bilgi testine göre benzer programlama bilgisine sahip olduğu belirlenmiştir (Düşük düzey). Öğrencilerin birlikte çalışmaları ve yardımlaşmaları için 6 işbirlikli grup oluşturulmuştur. Çalışmanın sonunda öğrencilerle BİD ve programlama öz-yeterliliklerinin gelişimine ilişkin görüşmeler yapılmıştır. Şekil 1’de araştırma süreci gösterilmiştir.

Şekil 1. Çalışma Süreci



Programlamaya giriş dersi otantik öğrenme etkinlikleriyle yürütülmüştür. Öğrencilerin ders dışında yararlanmaları için mobil öğrenme destek sistemi tasarlanmıştır. Şekil 2’de yapılan ders tasarımının bileşenleri gösterilmiştir.



Şekil 2. Ders Tasarımı

Öğretim elemanı ve öğrencilerin ders dışında bilgi ve içerik paylaşımında bulunmaları için mobil öğrenme destek sistemi tasarlanmıştır. Öğretim elemanı her ders öncesinde işlenecek konuya ilişkin sunular, kavram haritaları ve kod örnekleri paylaşmıştır. Öğrencilerin bu içerikleri inceleyerek ve derse ön hazırlık yaparak gelebilmeleri sağlanmıştır. Öğretim elemanı öğrencilere basit programlama görevleri vererek çözüm üretmeleri, akademik destek almaları ve tartışma yapmalarını beklemiştir. Öğrencilerle daha hızlı iletişim için WhatsApp grubu oluşturulmuş ve mobil öğrenme destek sistemindeki güncellemeler ve derse ilişkin duyurular bu grup aracılığıyla yapılmıştır. Mobil öğrenme destek sistemi öğrencilerin derse hazırlanabilecekleri dersten sonra da tekrar yapabilecekleri bir platformdur. Öğretim elemanı mobil öğrenme destek sistemini haftalık güncelleyerek öğrencilerin öğrenmelerini pekiştirmelerini amaçlamıştır. Derste öğretim elemanı mobil öğrenme destek sistemi aracılığıyla öğrencilerle paylaştığı sunular üzerinden dersi anlatmış, teorik bilgiler vermiş ve derse gelmeden önce paylaştığı basit programlama görevlerine ilişkin çözümü göstermiştir. Bu süreçte öğretim elemanı öğrencilerden gelen sorulara cevap vermiş ve kavram yanlışlarını gidermeye çalışmıştır. Ardından öğrenciler kendilerine verilen otantik programlama görevlerini grup arkadaşlarıyla birlikte tamamlamıştır. Öğretim elemanı bu süreçte öğrencilerine destek olmuştur. Öğrencilere araştırma yapma ve internet aracılığıyla farklı bilgi kaynaklarına ulaşma fırsatı verilmiştir. Gruplar kendilerine verilen göreve ilişkin çözümlerini diğer gruplarla

paylaşmışlardır. Bu doğrultuda birbirlerinin deneyimlerinden yararlanmaları amaçlanmıştır.

Öğrenciler mobil öğrenme destek sistemi ile her dersin sonunda kendi öğrenme süreçlerini ve performanslarını değerlendirmiştir. Otantik programlama etkinliklerinin kapsamı Şekil 3'te gösterilmiştir.

<p>Otantik bağlam</p> <ul style="list-style-type: none"> •Öğrenciler her hafta gerçek yaşamda kullanabilecekleri program kodlamıştır. Örneğin programlamada karar yapılarına ilişkin olarak 1. grup, Boy kilo indeksi hesaplayan ve kullanıcıların zayıf ya da şişman olmalarına ilişkin bilgi veren program geliştirmiştir. 2.grup, kullanıcıdan vize ve final notlarını isteyip harf notuna ve ders geçme durumuna ilişkin bilgi veren program yazmıştır. 3.grup, küçük işletmeler için satışı yapılan ürünlerin KDV'sini hesaplayan program yazmıştır. Programda kullanıcıdan satışı yapılan ürünün türü, adedi ve alış fiyatı istenmiş bu üç kritere göre KDV oranı hesaplanmıştır. Diğer gruplara da gerçek yaşama yönelik benzer problemler verilmiştir.
<p>Otantik görevler</p> <ul style="list-style-type: none"> •Her gruba birbirinden farklı iyi tanımlanmamış problem durumu verilmiştir. (Örn. Öğrenci gruplarının birine Boy kilo indeksi hesabının nasıl yapıldığına ilişkin düşünceleri ve çözüm önermeleri beklenmiştir.)
<p>Uzman performansına erişim ve modelleme</p> <ul style="list-style-type: none"> •Öğrencilerin problemin çözümüne ilişkin internet ve diğer öğrencilerden araştırma yapmaları ve destek almaları teşvik edilmiştir. Ayrıca öğrencilere birbirlerinin deneyimlerinden yararlanma fırsatı verilmiştir. (Örn. Öğrencilerin Boy kilo indeksinin matematiksel olarak nasıl hesaplandığına ilişkin internetten ya da diğer öğrencilerden yardım almalarına izin verilmiştir.)
<p>Çoklu roller ve perspektifler</p> <ul style="list-style-type: none"> • Öğrenciler otantik programlama görevlerini birden fazla yolla çözebilirler. Öğrenciler okul dışında Whatsapp grubu üzerinden diğer öğrencilerle programlama üzerine tartışma ve bilgi paylaşmaları için teşvik edilmiştir.
<p>Bilginin işbirlikli inşası</p> <ul style="list-style-type: none"> •Öğrenciler otantik programlama görevlerini grup arkadaşlarıyla birlikte tamamlamıştır.
<p>Yansıtma</p> <ul style="list-style-type: none"> •Öğrenciler çözümlerini grup arkadaşlarıyla tartıştıktan sonra öğretim elemanına sunmuştur. Öğretim elemanı verilen görevi tamamlayan grupların çözümlerini diğer gruplara sunmalarını istemiştir.
<p>Ekleme</p> <ul style="list-style-type: none"> •Otantik görevler önceki hafta işlenen konuları kapsayacak şekilde basitten karmaşığa doğru kurgulanmıştır. Gruplar kendilerine verilen görevlere ilişkin çözümlerini sunarken diğer öğrencilerin değerlendirmeleri beklenmiştir. Bu doğrultuda öğrencilerin kendilerinin ve diğer öğrencilerin hatalarını görerek eksiklerini tamamlamaları amaçlanmıştır.
<p>Koçluk ve destek</p> <ul style="list-style-type: none"> •Öğretim elemanı öğrencilere otantik görevleri yaparken yaptıkları hatalara ilişkin sürekli geribildirim vermiştir. Öğrencileri öğrenme sürecinde destekleyerek rehberlik yapmıştır.
<p>Otantik değerlendirme</p> <ul style="list-style-type: none"> •Öğrenciler her dersin sonunda kendi performanslarını ve grup arkadaşının performansını değerlendirmiştir. Öğrencilerden varsa başarısızlıklarının gerekçelerini ve bu duruma ilişkin çözüm yollarını ifade etmeleri beklenmiştir.

Şekil 3. Otantik Programlama Etkinlikleri

BULGULAR

1. Otantik programlama etkinlikleri öğrencilerin bilgi işlemsel düşünme becerilerini anlamlı şekilde geliştirir mi?

Öğrencilerin BİD becerilerindeki gelişimi incelemek için çalışma öncesinde ve sonrasında BİD ölçeği kullanılmıştır. Öğrencilerin yaratıcılığı çalışma öncesinde yüksek düzeydeyken çalışmanın sonunda biraz daha gelişmiştir. Algoritmik düşünme düşük düzeyden orta düzeye çıkmıştır. İşbirliklilik ve eleştirel düşünme çalışma öncesinde ve sonrasında yüksek düzeydeyken çalışmanın sonunda biraz daha gelişmiştir. Problem çözmeye ilişkin sorular olumsuz olarak sunulduğundan düşük puan olumlu, yüksek puan olumsuz olarak değerlendirilmiştir. Bu çerçevede problem çözme becerisinin geliştiği ve yüksek düzeyde olduğu söylenebilir. BİD ölçeği genel ortalamalar incelendiğinde öğrencilerin orta düzeyde olduğu ve çalışma öncesine göre BİD beceri puanlarının arttığı ortaya çıkmıştır. Betimsel istatistik sonuçları Tablo 1’de gösterilmiştir.

Tablo 1.
BİD Ölçeği Betimsel Analizler

BİD Faktörleri	N	SS	X	Minimum	Maksimum
Yaratıcılık-önce	17	11.182	83.970	65	100
Yaratıcılık-sonra	17	7.606	93.529	72.5	100
Algoritmik düşünme-önce	17	21.139	49.019	20	96
Algoritmik düşünme-sonra	17	20.016	63.529	26	93
İşbirliklilik-önce	17	23.064	79.117	20	100
İşbirliklilik-sonra	17	20.615	85	30	100
Eleştirel düşünme-önce	17	22.186	58.117	20	92
Eleştirel düşünme-sonra	17	18.843	79.058	20	100
Problem çözme-önce	17	14.630	45.098	23.33	70
Problem çözme-sonra	17	15.181	37.843	20	70
Genel ortalama-önce	17	8.227	63.610	45.52	73.79
Genel ortalama-sonra	17	8.959	72.129	42.07	82.07

Öğrencilerin yaratıcılık, algoritmik düşünme ve eleştirel düşünme becerileri çalışma öncesine göre anlamlı şekilde gelişmiştir. İşbirliklilik ve problem çözme puanı çalışma öncesine göre artmasına rağmen anlamlı bir gelişme olmamıştır. BİD genel ortalama puanı çalışma

öncesine göre anlamlı şekilde artmıştır. Öğrencilerin BİD ölçeğine ilişkin eşleştirilmiş t testinden elde edilen sonuçlar Tablo 2’de gösterilmiştir.

Tablo 2.

BİD Ölçeği Eşleştirilmiş T Testi Sonuçlar

BİD Faktörleri	t	df	Sig. (2-tailed)
Yaratıcılık-önce - Yaratıcılık-sonra	-2,971	16	,009*
Algoritmik düşünme-önce - Algoritmik düşünme-sonra	-3,045	16	,008*
İşbirliklilik-önce - İşbirliklilik-sonra	-1,936	16	,071
Eleştirel düşünme-önce - Eleştirel düşünme-sonra	-4,074	16	,001*
Problem çözme-önce - Problem çözme-sonra	2,078	16	,054
Genel ortalama-önce - Genel ortalama-sonra	-4,080	16	,001*

*p<0.05

Öğrenciler, yapılan görüşmelerde yaratıcılık becerisinin gelişimini çoğunlukla “Kodlama yapmak için yeterli zamanın verilmesi”, “Problemleri çözdükçe kendine olan güvenin gelişmesi” ve “Gerçek hayat problemleri ile çalışmak” ile açıklamıştır. Örneğin Ö3 bu durumu şöyle ifade etmiştir:

“...Her ders farklı kodlama problemi ile çalıştım. Kod yazmak için yeterli zamanım olduğu için sorunları kolayca çözdüm...”

Ö7 ise şöyle ifade etmiştir:

“...Hocamızın verdiği kodlama sorularını yaptıkça programlamayı yapabileceğimi gördüm. Ayrıca sorulan soruların günlük hayatımıza yönelik olması daha kolay anlamamızı sağladı...”

Öğrenciler algoritmik düşünme becerisindeki gelişimi genellikle “Verilen görevlerin kolaydan zora doğru sıralanması”, “Kod yazmayı öğrenmek” olarak ifade etmiştir. Ö9’un değerlendirmesi şu şekildedir:

“...Derste önce kolay kodlarla çalıştık, sonra daha zor kodlamalar yaptık. Kodladıkça, programlama mantığını daha iyi anladım...”

Ö12 ise şöyle ifade etmiştir:

“...Dersten önce hocamız kolay sorular sorup derse gelince bu soruları çözüyordu. Bizlere uygulamalı olarak algoritması ile birlikte kod yazmayı öğretiyordu bu durum programlama mantığını anlamamı sağladı...”

Eleştirel düşünme becerisindeki gelişim “Zor problemler üzerinde çalışma fırsatı verilmesi”, “Her ders performansımızı değerlendirmek için gerekli ortamın sağlanması”, “Problemleri grup arkadaşım ile birlikte çözmek” ile ilişkilendirilmiştir. Ö10 bu durumu şöyle değerlendirmiştir:

“...Hocamız bazı derslerde çok zor sorular sordu. Bu soruları çözmekte zorluk çeksem de eksikliklerimi gördüm...”

Ö3’ün değerlendirilmesi şu şekildedir:

“...Dersler ilerledikçe daha zor kodlar yazmaya başladık. Derslerin sonunda kendimizi değerlendirdik bu durum beni daha fazla çalışmaya itti...”

Öğrenciler işbirliklilik düzeylerindeki artışı “Grup arkadaşım ile birlikte çalışmanın dersi eğlenceli hale getirmesi”, “Rahat bir ortamın sağlanması” ve “Zor görevlerle çalışmayı kolaylaştırması” şeklinde ifade etmiştir. Ö5’in açıklaması şu şekildedir:

“...Arkadaşımın yardımıyla kodlama yapmak çok faydalı oldu. Derste sıkılmadım. Zor soruları birlikte çözdük...”

Ö16 ise bu durumu şöyle değerlendirmiştir:

“...Kodlama yaparken arkadaşım ile birlikte çalıştık. Sınıfta rahat çalışma ortamı vardı yardımlaşarak çalışmak, hocamızdan destek almak çok iyiydi...”

Öğrenciler problem çözme becerisinin gelişimini “Derste verilen problemlerin ilgi çekici olması”, “Öğretim elemanı ve akran desteği”, “İşbirlikli çalışmalar” ile açıklamışlardır. Ö1 bu durumu şöyle ifade etmiştir:

“...Derste sorulan sorular günlük hayatımızla ilgiliydi. Bu soruları daha kolay anlamamı ve çözmemi sağladı...”

Bilgi işlemsel düşünme becerilerindeki genel gelişimi “İlgi çekici etkinlikler yapılması”, “Evde ve okulda sürekli destek alabilme fırsatı verilmesi”, “Gerçek hayattan örnekler üzerinde çalışmanın öğrenmeyi kolaylaştırması”, “Her dersin önceki hafta işlenen konularla bağlantılı olması” ve “Programlama yapmayı öğrenmek” ile açıklamışlardır. Ö2 bu durumu şöyle değerlendirmiştir:

“...Evde ve okulda hocama kolayca sorular sordum. Mobil site ile ders içeriğine erişebildim ve destek alabildim. Çözdüğümüz örnekler programlama öğrenmemi kolaylaştırdı...”

Ö10 genel değerlendirmeyi şöyle yapmıştır:

“...Hocamız bize sürekli destek oldu, dersten önce ve sonra ders notlarına ulaşmak derse hazırlanmamı ve tekrar yapabilmemi sağladı...”

Bazı öğrencilerin beceri puanlarında düşüş ya da çalışma öncesinde göre bir gelişmediği görülmüştür. Öğrenciler problem çözme becerisindeki bu durumu “Bazen bir problemi çözememenin diğer çözümleri de olumsuz etkilemesi” olarak ifade ederken işbirliklilik becerisindeki durumu “Grup arkadaşının performansının düşük olmasının olumsuz etkisi”, “Bireysel çalışma isteğinin olması” şeklinde açıklamışlardır. Ö17 bu durumu şu şekilde değerlendirmiştir:

“...Bazen grup arkadaşım dersle ilgilenmiyordu. Zor sorularda bütün işler üstümde kaldı. Bu durum beni olumsuz etkiledi...”

Tablo 3’te öğrencilerin BİD gelişimine ilişkin yaptıkları değerlendirmeler sonucunda ortaya çıkan kodlamalar gösterilmiştir.

Tablo 3.
Öğrencilerin BID Becerilerinin Gelişimine İlişkin Değerlendirmeleri

BID Faktörleri	BID Becerilerinin Gelişimi	BID Becerilerinin Gelişmemesi
Yaratıcılık	Yeterli zaman verilmesi Öz-güven gelişmesi Gerçek yaşam problemleri	
Algoritmik düşünme	Problemlerin kolaydan zora sıralanması Kodlama yapma	
Eleştirel düşünme	Zor problemlerle çalışma Kendini değerlendirme Akranla çalışma	
İşbirliklilik	Rahat öğrenme ortamı Kolay öğrenme	Bazı problemlerin çok zor olması Grup içindeki performans farklılıkları Bireysel çalışma isteği
Problem çözme	İlgi çekici problemler Öğretim elemanı desteği Akran desteği İşbirlikli öğrenme	
Genel ortalama	İlgi çekici etkinlikler Sürekli destek Günlük yaşam problemleri İlişkili konular Kodlama öğrenme	

2. Otantik programlama etkinlikleri öğrencilerin programlama öz-yeterlilik inancını anlamlı şekilde geliştirir mi?

Öğrencilerin programlama öz-yeterlilik inancındaki gelişimi incelemek için çalışma öncesinde ve sonrasında programlama öz-yeterlilik ölçeği kullanılmıştır. Toplanan verilerin analizinde betimleyici istatistik ve eşleştirilmiş t testinden yararlanılmıştır. Betimsel istatistik sonuçları Tablo 4’te gösterilmiştir.

Tablo 4.
Programlama Öz-Yeterlilik Ölçeği Betimsel Analizler

Programlama Öz-yeterlilik Faktörleri	N	SS	X	Minimum	Maksimum
Basit programlama görevleri-önce	17	1.803	3.274	1	7
Basit programlama görevleri-sonra		1.602	5.980	1	7
Karmaşık programlama görevleri-önce		1.696	3.215	1	7
Karmaşık programlama görevleri-sonra		1.710	4.705	1	7
Genel ortalama-önce		1.685	3.235	1	7
Genel ortalama-sonra		1.572	5.130	1	7

Öğrencilerin basit programlama görevleri ve genel ortalamaları düşük düzeyden yüksek düzeye çıkmıştır. Karmaşık programlama görevlerine ilişkin öz-yeterliliği ise düşük düzeyden orta düzeye yükselmiştir. Öğrencilerin programlama öz-yeterlilik inancı ölçeğine ilişkin eşleştirilmiş t testinden elde edilen sonuçlar Tablo 5’te gösterilmiştir.

Tablo 5.
Programlama Öz-Yeterlik Ölçeği Eşleştirilmiş T Testi Sonuçları

Programlama Öz-yeterlik Faktörleri	t	df	Sig. (2-tailed)
Basit programlama görevleri-önce - Basit programlama görevleri-sonra	-5,422	16	,000*
Karmaşık programlama görevleri-önce - Karmaşık programlama görevleri-sonra	-4,159	16	,001*
Genel ortalama-önce - Genel ortalama-sonra	-4,080	16	,001*

*p<0.05

Öğrencilerin basit programlama görevleri, karmaşık programlama görevleri ve genel ortalamalarının anlamlı şekilde geliştiği belirlenmiştir. Öğrenciler, “Her ders düzenli olarak kod yazma fırsatı verilmesi”, “Grup çalışması”, “Evde ve okulda sürekli olarak akran ve öğretmen desteği”, “Kodlama uygulamalarının bildiğimiz konularla ilişkilendirilmesi” ile açıklamıştır. Ö3 bu durumu şöyle açıklamıştır:

“...Doğrudan dersi dinlemek yerine kodlama yapmak daha kolay öğrenmemi sağladı. Kodladıkça kendime güvenim arttı. Grupla çalışmak ve ders notlarına evden ulaşmak başarılı olmamda etkili oldu...”

Ö14’ün açıklaması şu şekildedir:

“...Dersler çok eğlenceli geçiyordu, dersi kaçırsam bile hocamız içerikleri bizimle paylaşıyordu. Her zaman hocamdan ve arkadaşımından destek alabileceğimi bilmek kendime olan güvenimi artırdı...”

Birkaç öğrencide programlama öz-yeterliliğinin gelişmediği ya da sabit kaldığı görülmüştür. Öğrenciler bu durumu “Evde yeterince uygulama ve tekrar yapmamak”, “Verilen bazı görevi tamamlayamamanın olumsuz etkisi” ile açıklamıştır. Ö16’nın değerlendirmesi şöyledir:

“...Okulda öğrendiklerimi evde tekrar edemedim. Evde daha fazla kodlama yapmalıydım....”.

TARTIŞMA, SONUÇ VE ÖNERİLER

Bu çalışmada öğrencilere otantik problemler üzerinde çalışma fırsatı verilmiştir. Her gruba farklı bir problem durumu verilerek öğrencilerin farklı çözümler üretmeleri beklenmiştir. Öğrencilere kendilerine verilen görevleri tamamlamaları için yeterli süre verilmiştir. Öğrenciler grup içinde birbiriyle tartışarak çözüm önerilerini birbirleriyle karşılaştırmış ve nihai çözümlerini öğretim elemanına sunmuştur. Öğretim elemanı öğrencilere mevcut çözümlerine ilişkin geribildirim vermiş ve destek olmuştur. Bu çerçevede otantik öğrenme etkinliklerinin öğrencilere bağımsız ve esnek öğrenme ve farklı fikirler üretebilme imkânı sunarak yaratıcı düşünme becerilerini geliştirebildiği ortaya çıkmıştır. Eddles-Hirsch ve ark. (2019) çalışmasında araştırma yapma, problemlere çözümler üretme, yansıtma ve değerlendirme gibi otantik etkinliklerin öğrencilerin yaratıcılık becerilerini geliştirebileceğine ilişkin örnekler sunmuştur. Bu çalışmada öğrenciler yaratıcılık becerilerindeki gelişimi otantik problemler üzerinde çalışmak için gerekli ortamın sağlanmasıyla ilişkilendirmişlerdir. Çalışmanın sonunda öğrencilerin yaratıcılık becerilerinin çalışma öncesine göre anlamlı şekilde geliştiği görülmüştür.

Öğrenciler evde basit problemlerle çalışırken okulda daha karmaşık gerçek yaşam problemleriyle kodlama yapmıştır ve verilen problemin önce algoritmasını oluşturmuş ardından kodlamaya geçmişlerdir. Malik ve ark. (2019) programlamaya giriş dersinde öğrencilerin kodlamaya geçmeden önce programın çözüm adımlarını planlamanın ve gerçek yaşam problemleri üzerinde çalışmanın geleneksel programlama eğitime göre algoritmik düşünmeyi daha fazla geliştirebildiğini ortaya çıkarmıştır. Bu durum öğrencilerin her hafta otantik problemlerin belirli bir zorluk seviyesine göre sıralanması ve düzenli olarak kodlama yapma imkânı verilmesinin algoritmik düşüncelerine katkı sağladığını ifade etmeleri ile aynı çerçevede değerlendirilebilir. Bu çerçevede öğrencilerin algoritmik düşünmesinin çalışma öncesine göre anlamlı şekilde geliştiği ortaya çıkmıştır.

Öğrenciler otantik programlama görevlerini grup arkadaşıyla birlikte tamamlamıştır. Öğrenciler deneyimlerini paylaşarak birbirlerinden destek almışlardır. İşbirlikli grup çalışmasıyla

öğrencilerin birbirlerine problemlere ilişkin olarak çözümlerini rahatlıkla ifade etmeleri ve tartışmaları amaçlanmıştır. Bu çalışmada öğrenciler işbirlikli grup çalışması yapmaları için teşvik edilmiştir. Hwang ve ark. (2012) işbirlikli programlama görevlerinin öğrencilerin performanslarını olumlu yönde geliştirdiğini ortaya çıkarmıştır. Öğrencilerin işbirlikli çalışma puanı çalışma öncesine göre artsa da anlamlı şekilde gelişmemiştir. Bu durumun bazı öğrencilerin ifade ettiği gibi bireysel çalışmaya daha yatkın olunması ve grup içindeki performansı düşük öğrencilerin performansı yüksek öğrencileri olumsuz etkilemesiyle ilişkili olabileceği düşünülmektedir.

Öğrenciler otantik öğrenme etkinlikleri ile karmaşık programlama görevleri üzerinde çalışmıştır. Ayrıca onlar diğer gruplara verilen farklı karmaşık problemlerin çözümünü de incelemiş ve değerlendirmiştir. Öğrencilere verilen otantik programlama görevlerinin zorluğuna göre farklı süreler verilerek üzerinde yeterince düşünme fırsatı verilmiştir. Ayrıca her ders performanslarına ilişkin değerlendirme yapmaları sağlanarak eksiklerini fark etmeleri sağlanmıştır. Wang (2017) öğrencilere akranları değerlendirme ve projeler üzerinde çalışma fırsatı verilmesinin geleneksel programlama öğretimine göre programlama becerilerini ve eleştirel düşüncelerini anlamlı şekilde geliştirdiğini ortaya çıkarmıştır. Bu çalışmada öğrencilere her ders kendilerini ve akranlarını değerlendirme ve karmaşık problemler üzerinde uygulama yapma fırsatı verilmesinin öğrencilerin eleştirel düşüncelerine katkı sağlayabildiği düşünülmektedir.

Öğretim elemanı öğrencilere günlük hayatlarında karşılaşılabilecekleri ilgi çekici programlama görevleri vermiştir. Öğrencilerin karmaşık problemleri bildikleri konularla ilişkilendirerek çözmeleri amaçlanmıştır. Ayrıca işbirlikli grup çalışmasıyla öğrenciler zor problemler ile çalışırken birbirlerinin deneyimlerinden faydalanmıştır. Ge ve ark. (2006), öğrencilere programlama dersinde işbirlikli grup çalışması, açık uçlu ve otantik problemlerle çalışmanın problem çözmelerine katkı sağladığını ortaya çıkarmıştır. Bu çalışmada benzer etkinlikler yapılmıştır. Öğrencilerin problem çözme beceri puanlarında artış olsa da anlamlı

gelişme olmamıştır. Bazı öğrencilerin verilen problemleri çözmekte zorlanmasının diğer problemler üzerindeki çalışmalarını olumsuz etkilediği düşünülmektedir. Ayrıca bazı gruplarda öğrencilerin performansındaki farklılıkların ve bireysel çalışmanın tercih edilmesinin karmaşık problemler üzerinde çalışmayı zorlaştırdığı söylenebilir.

Programlamaya giriş dersinde öğrencilerin gerçek hayatla ilişkili problemlerle işbirlikli çalışarak farklı çözüm önerileri getirmişler ve birbirlerinin deneyimlerinden yararlanmışlardır. Öğrenciler verilen problemlere ilişkin çözümlerini diğer gruplarla paylaşırken kendilerinin ve birbirlerinin performanslarını değerlendirme imkanı bulmuşlardır. Öğretim elemanı okul içinde ve dışında öğrencilerini desteklemiş ve geribildirim vermiştir. Bu doğrultuda öğrencilerin bilgi işlemsel düşünmesinin çalışma öncesine göre anlamlı şekilde geliştiği ortaya çıkmıştır.

Öğrencilere verilen otantik programlama görevleri basitten karmaşığa doğru sıralanmıştır. Öğretim elemanı öğrencilerin basit programlama görevlerini tamamlaması ile kendilerine olan güvenlerinin artmasını amaçlamıştır. Öğrenciler işbirlikli grup çalışması ile özellikle zor görevler ile çalışırken birbirlerine destek olmuşlardır. Öğrenciler her ders öğrendikleri komutları farklı programlama görevleri üzerinde uygulamıştır. Öğretim elemanı bu süreçte öğrencilerin çözümlerine geribildirim vererek hatalarını düzeltmelerini sağlamıştır. Bu doğrultuda öğrencilerin programlama öz-yeterlilik inançlarını anlamlı şekilde geliştirebildiği ortaya çıkmıştır. Abdunabi ve ark. (2019) çalışmasında öğrencilerin gerçek yaşam projeleriyle çalışmasının programlama öz-yeterliliğine olumlu katkısı olabileceğini ifade etmektedir. Bazı öğrencilerin öz-yeterliliklerinin ise değişmediği görülmüştür. Öğrencilerin evde yeterince tekrar yapmamaları ve bazı problemleri çözmekte zorlanmasının kendilerine olan güvenlerini olumsuz etkilemiş olabilir.

Bu çalışmada yükseköğretimde programlama dersinde otantik öğrenme etkinliklerinin öğrencilerin bilgi işlemsel düşüncelerine ve programlama öz-yeterlilik inançlarına olumlu yönde katkı sağlayabileceği ortaya çıkmıştır. Gelecekteki araştırmalarda farklı eğitim seviyelerinde ve daha geniş örneklerle çalışılabilir. Öğrencilerin bilgi işlemsel düşünmenin

problem çözüme ve işbirliklilik gibi bazı alt faktörlerinde anlamlı gelişmeler olmamıştır.

Öğrencilerin evde tekrar yapmalarını teşvik edecek farklı etkinlikler planlanabilir. Diğer taraftan

bazı öğrencilerin grup içindeki performans farklılıklarından olumsuz etkilendiği ortaya çıkmıştır.

Bu çerçevede öğrenciler her hafta gruplar arasında çaprazlanarak farklı öğrencilerle çalışma fırsatı bulabilir.

Çıkar Çatışması Bildirimi

Yazar, bu makalenin araştırılması, yazarlığı ve / veya yayınlanmasına ilişkin herhangi bir potansiyel çıkar çatışması beyan etmemiştir.

Destek/Finansman Bilgileri

Yazar(lar), bu makalenin araştırılması, yazarlığı ve / veya yayınlanması için herhangi bir finansal destek almamıştır.

Etik Kurul Kararı

Bu araştırma için Aksaray Üniversitesi kurumundan (24.04.2020-2020/03-09) etik izin alınmıştır.

KAYNAKÇA/REFERENCES

- Abdunabi, R., Hbaci, I., & Ku, H-Y. (2019). Towards enhancing programming self-efficacy perceptions among undergraduate information systems students. *Journal of Information Technology Education: Research*, 18, 185- 206. <https://doi.org/10.28945/4308>
- Askar, P., & Davenport, D. (2009). An investigation of factors related to self-efficacy for Java Programming among engineering students. *Online Submission*, 8(1).
- Castledine, A. R., & Chalmers, C. (2011). LEGO Robotics: An authentic problem solving tool?. *Design and Technology Education: An International Journal*, 16(3).
- Chin, K. Y., Lee, K. F., & Chen, Y. L. (2018). Using an interactive ubiquitous learning system to enhance authentic learning experiences in a cultural heritage course. *Interactive Learning Environments*, 26(4), 444-459. <https://doi.org/10.1080/10494820.2017.1341939>
- Ching, Y. H., Hsu, Y. C., & Baldwin, S. (2018). Developing computational thinking with educational technologies for young learners. *TechTrends*, 62(6), 563-573. <https://doi.org/10.1007/s11528-018-0292-7>
- Chiu, P. S., Pu, Y. H., Kao, C. C., Wu, T. T., & Huang, Y. M. (2018). An authentic learning based evaluation method for mobile learning in Higher Education. *Innovations in Education and Teaching International*, 55(3), 336-347. <https://doi.org/10.1080/14703297.2017.1417147>
- Creswell John W. (2012). *Planning, Conducting, and Evaluating Quantitative and Qualitative Research*. Lincoln: University of Nebraska Daulat Purnama
- Eddles-Hirsch, K., Kennedy-Clark, S., & Francis, T. (2019). Developing creativity through authentic programming in the inclusive classroom. *Education* 3(13), 1-10. <https://doi.org/10.1080/03004279.2019.1670714>
- Frydenberg, M. (2015). Achieving digital literacy through game development: an authentic learning experience. *Interactive Technology and Smart Education*, 12(4), 256-269. <https://doi.org/10.1108/ITSE-08-2015-0022>
- Garneli, V., & Chorianopoulos, K. (2018). Programming video games and simulations in science education: exploring computational thinking through code analysis. *Interactive Learning Environments*, 26(3), 386-401. <https://doi.org/10.1080/10494820.2017.1337036>
- Ge, X., Thomas, M. K., & Greene, B. A. (2006). Technology-rich ethnography for examining the transition to authentic problem-solving in a high school computer programming class. *Journal of Educational Computing Research*, 34(4), 319-352. <https://doi.org/10.2190/924L-4002-J5P4-1077>
- Herrington, J., Reeves, T. C., & Oliver, R. (2014). Authentic learning environments. In *Handbook of research on educational communications and technology*. Springer, New York, NY.
- Kong, S. C., Chiu, M. M., & Lai, M. (2018). A study of primary school students' interest, collaboration attitude, and programming empowerment in computational thinking education. *Computers & Education*, 127, 178-189. <https://doi.org/10.1016/j.compedu.2018.08.026>
- Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). A validity and reliability study of the Computational Thinking Scales (CTS). *Computers in Human Behavior*, 72, 558-569. <https://doi.org/10.1016/j.chb.2017.01.005>
- Lin, C. C., Chao, P. Y., Lin, E. T., & Tzeng, H. L. (2018). Exploring the Role of Visual Programming Activities in Computational Thinking. In *2018 1st International Cognitive Cities Conference (IC3)*. IEEE.

- Malik, S. I., Shakir, M., Eldow, A., & Ashfaque, M. W. (2019). Promoting Algorithmic Thinking in an Introductory Programming Course. *International Journal of Emerging Technologies in Learning*, 14(1).
- Miles, M. B., & Huberman, A. M. (1994). *An expanded sourcebook qualitative data analysis*. Sage Publications, London.
- Nouri, J., Zhang, L., Mannila, L., & Norén, E. (2019). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry*. <https://doi.org/10.1080/20004508.2019.1627844>
- Peng, J., Wang, M., & Sampson, D. (2017). Visualizing the complex process for deep learning with an authentic programming project. *Journal of Educational Technology & Society*, 20(4), 275-287.
- Pérez-Marín, D., Hijón-Neira, R., Bacelo, A., & Pizarro, C. (2018). Can computational thinking be improved by using a methodology based on metaphors and scratch to teach computer programming to children?. *Computers in Human Behavior*, 105. <https://doi.org/10.1016/j.chb.2018.12.027>
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Kafai, Y. B. (2009). Scratch: Programming for all. *Commun. Acm*, 52(11), 60-67. <https://doi.org/10.1145/1592761.1592779>
- Roach, K., Tilley, E., & Mitchell, J. (2018). How authentic does authentic learning have to be?. *Higher Education Pedagogies*, 3(1), 495-509. <https://doi.org/10.1080/23752696.2018.1462099>
- Rodríguez-Martínez, J. A., González-Calero, J. A., & Sáez-López, J. M. (2019). Computational thinking and mathematics using Scratch: an experiment with sixth-grade students. *Interactive Learning Environments*. <https://doi.org/10.1080/10494820.2019.1612448>
- Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the computational thinking test. *Computers in Human Behavior*, 72, 678–691. <https://doi.org/10.1016/j.chb.2016.08.047>
- Shadiev, R., Hwang, W. Y., & Huang, Y. M. (2017). Review of research on mobile language learning in authentic environments. *Computer Assisted Language Learning*, 30(3-4), 284-303. <https://doi.org/10.1080/09588221.2017.1308383>
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142–158. <https://doi.org/10.1016/j.edurev.2017.09.003>
- Thomas, M. K., Ge, X., & Greene, B. A. (2011). Fostering 21st century skill development by engaging students in authentic game design projects in a high school computer programming class. *Journal of Educational Computing Research*, 44(4), 391-408. <https://doi.org/10.2190/EC.44.4.b>
- Wang, X.-M., Hwang, G.-J., Liang, Z.-Y., & Wang, H.-Y. (2017). Enhancing Students' Computer Programming Performances, Critical Thinking Awareness and Attitudes towards Programming: An Online Peer-Assessment Attempt. *Educational Technology & Society*, 20 (4), 58–68.
- Wing, J. (2011). Research notebook: Computational thinking—What and why. *The Link Magazine*, 20-23.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. <https://doi.org/10.1145/1118178.1118215>

Wong, G. K. W., & Cheung, H. Y. (2018). Exploring children's perceptions of developing twenty-first century skills through computational thinking and programming. *Interactive Learning Environments*. <https://doi.org/10.1080/10494820.2018.1534245>

Yıldırım, A., & Şimşek, H. (2008). *Sosyal Bilimlerde Nitel Araştırma Yöntemleri (11. Baskı)*. Ankara: Seçkin Yayıncılık.

İletişim/Correspondence

Dr. Öğr. Üyesi Mücahit ÖZTÜRK
mucahitozturk@aksaray.edu.tr