



**T.C.
AKSARAY ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

HARİTA MÜHENDİSLİĞİ ANABİLİM DALI

**ÇOK ÖLÇÜTLÜ KARAR VERME ANALİZLERİ İÇİN AÇIK
KAYNAK KODLU EKLENTİ GELİŞTİRİLMESİ**

YÜKSEK LİSANS TEZİ

Emre YILMAZ

DANIŞMAN

Dr. Öğr. Üyesi Süleyman Sefa BİLGİLİOĞLU

AKSARAY, 2022



**T.C.
AKSARAY ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

HARİTA MÜHENDİSLİĞİ ANABİLİM DALI

**ÇOK ÖLÇÜTLÜ KARAR VERME ANALİZLERİ İÇİN AÇIK
KAYNAK KODLU EKLENTİ GELİŞTİRİLMESİ**

YÜKSEK LİSANS TEZİ

Emre YILMAZ

DANIŞMAN

Dr. Öğr. Üyesi Süleyman Sefa BİLGİLİOĞLU

AKSARAY, 2022

Aksaray Üniversitesi Fen Bilimleri Enstitüsü'nün 202306019 numaralı Yüksek Lisans öğrencisi Emre YILMAZ tarafından hazırlanan “ÇOK ÖLÇÜTLÜ KARAR VERME ANALİZLERİ İÇİN AÇIK KAYNAK KODLU EKLENTİ GELİŞTİRİLMESİ” adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Harita Mühendisliği Anabilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Dr. Öğr. Üyesi Süleyman Sefa BİLGİLİOĞLU

Aksaray Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum/onaylamıyorum

Üye: Prof. Dr. Hacı Murat YILMAZ

Aksaray Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum/onaylamıyorum

Üye: Dr. Öğr. Üyesi Muzaffer Can İban

Mersin Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum/onaylamıyorum

Tez Savunma Tarihi:/...../.....

Jüri tarafından kabul edilen bu tezin Doktora Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

.....

Doç. Dr.

Fen Bilimleri Enstitüsü Müdürü

DOĞRULUK BEYANI

Yüksek lisans tezi olarak sunduğum bu çalışmayı, akademik kurallara ve bilimsel etik, ahlak ve geleneklere aykırı düşecek bir yol ve yardıma başvurmaksızın yazdığımı, yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu, çalışmamda kullandığım verilerin orijinalliğini ve her türlü intihalden uzak olduğunu beyan ederim.

Enstitü tarafından belli bir zamana bağlı olmaksızın, tezimle ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçlara katlanacağımı bildiririm.

İmza

Emre YILMAZ

TEŞEKKÜR

Tüm hayatım boyunca attığım her adımda yanımda olan, hiçbir fedakârlığı esirgemeyen ve çalışmalarım esnasında manevi destekleri için her zaman minnettar olduğum annem Nuran YILMAZ'a, babam Zülküf YILMAZ'a ve kardeşim Samet YILMAZ'a çok teşekkür ederim.

Bu çalışmamı tamamlamam ve gelecekte yeni çalışmalar yapmam için beni sürekli teşvik eden, çalışmalarına ayıracağım zamanı mutluluğumuza ayrılmış zaman, elde edeceğim mutluluğu ise kendi mutluluğu olarak gören hayat arkadaşım Öğr.Gör. Şüheda Yılmaz'a çok teşekkür ederim.

Bu çalışmanın yürütülmesi sırasında mesleki bilgi birikimiyle desteğini esirgemeyen, değerli tecrübelerini ve zamanını benimle paylaşan danışman hocam Dr. Öğr. Üyesi Süleyman Sefa BİLGİLİOĞLU'na çok teşekkür ediyorum. Ayrıca çalışmanın gelişimine değerli fikirleriyle katkı sunan hocalarım Prof. Dr. Hacı Murat YILMAZ ve Dr. Öğr. Üyesi Muzaffer Can İBAN'a teşekkür ediyorum.

Emre YILMAZ

AKSARAY, 2022

İÇİNDEKİLER

TEŞEKKÜR	i
İÇİNDEKİLER	ii
ÖZET.....	iii
ABSTRACT	iv
ŞEKİLLER DİZİNİ	v
ÇİZELGELER DİZİNİ	vi
SİMGELER VE KISALTMALAR.....	vii
1. GİRİŞ	1
2. TEMEL TANIM VE KAVRAMLAR	4
2.1 Coğrafi Bilgi Sistemleri.....	4
2.2 CBS Bileşenleri	4
2.3 Yazılım	7
2.3.1 Kapalı kaynak kodlu yazılım	8
2.3.2 Açık kaynak kodlu yazılım	9
2.4 Açık Kaynak Kodlu CBS Yazılımlarına Eklenti Geliştirilmesi	11
2.4.1 Python	13
2.4.2 QGIS	15
2.4.3 Qt Designer	16
2.4.5 PyQt	16
2.4.6 PyQGIS	17
2.4.7. GDAL ve OGR	18
2.5 Çok Ölçütlü Karar Verme Yöntemi	18
2.5.1 Analitik Hiyerarşi Yöntemi (AHY)	20
2.6 ÇÖKV Analizleri için Eklenti Geliştirilmesi	24
3. YAPILAN ÇALIŞMALAR	29
3.1. ÇÖKV Analizleri için Açık Kaynak Kodlu Eklenti Geliştirilmesi	30
3.1.1. Uygulama yöntemlerinin tespiti.....	30
3.1.2. Kullanılacak yazılımın tespiti	31
3.2. Yeni Eklenti Üretme.....	32
3.2.1. Plugin Builder	33
3.2.2. Yazılımın derlenmesi.....	34
3.3. Arayüz Tasarımı	35
3.3.1. Qt Designer	36
3.4. Eklenti Kodlama.....	43
3.4.1. Pyqgis.....	44
3.4.2. Fonksiyonların kodlanması.....	45
3.4.3. Signal-Slot kullanımı	49
4. SONUÇ VE ÖNERİLER.....	51
KAYNAKLAR	54
ÖZGEÇMİŞ.....	60

YÜKSEK LİSANS TEZİ

ÇOK ÖLÇÜTLÜ KARAR VERME ANALİZLERİ İÇİN AÇIK KAYNAK KODLU EKLENTİ GELİŞTİRİLMESİ

Emre YILMAZ

Aksaray Üniversitesi
Fen Bilimleri Enstitüsü
Harita Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Süleyman Sefa BİLGİLİOĞLU

ÖZET

Son yıllarda dünyada ve ülkemizde açık kaynak kodlu yazılımların kullanımı giderek artmaktadır. Açık kaynak kodlu yazılımlar, kullanıcılar tarafından daha güvenilir bulunmasının yanı sıra ücretsiz olması sayesinde popülerliğini her geçen gün arttırmakta, kurulan gönüllü topluluklar (community) sayesinde tüm dünya kullanıcılarını aynı hedefte buluşturmaktadır. Özgür yazılım dünyada Free Software olarak bilinmektedir. Free kelimesi ise ücretsiz anlamını değil yazılımdaki özgürlüğü ifade etmek anlamını taşır. Özgür yazılımın gelişimi için 1985 yılında ilk organizasyonel girişim olan Free Software Foundation (Özgür Yazılım Vakfı) kurulduktan sonra yıllar içerisinde birçok topluluk ve vakıf kurulmuştur. 2006 yılında kurulan Açık Kaynak Coğrafya Vakfı (OSGeo) açık kaynaklı bir CBS yazılımı olan QGIS'i geliştirmiştir. QGIS yazılımı beraberinde sunduğu açık kaynaklı python yazılımı desteği ile kullanıcıların kendi eklenti araçlarını geliştirmesini sağlarken, eklenti geliştiricilerin grafiksel kullanıcı ara yüzleri (GUI) tasarlayabilmesini sağlayan Qt Designer yazılımı ile farklı tasarım mimarilerinin gelişimine olanak verir. Bu çalışmada QGIS yazılımı yardımıyla çok ölçütlü karar verme (ÇÖKV) yöntemleri kullanılarak özgür bir CBS eklentisi tasarımı yapmak amaçlanmıştır. Bu kapsamda Qt Designer programı ile arayüz tasarımı yapılmış, eklentide yer alan coğrafi araçlar python programlama dilinden ve QGIS API'sini kullanarak eklenti geliştirmede kullanılan PyQGIS bağdaştırıcısından yararlanılarak kodlanmıştır. Coğrafi araçlar ile arayüz arasındaki signal-slot bağlantıları kodlanarak tasarım tamamlanmıştır. Tasarlanan eklentinin tüm dünya kullanıcılarına ücretsiz olarak paylaşılmasıyla özgür yazılımın gelişimine fayda sağlamak hedeflenmiştir.

Anahtar Kelimeler: Coğrafi Bilgi Sistemleri (CBS), Açık Kaynaklı Yazılım, Python, QGIS, Eklenti.

Haziran, 2022; 60 sayfa

M.Sc. THESIS

DEVELOPMENT OF OPEN SOURCE PLUGIN FOR MULTI-CRITERIA DECISION MAKING ANALYSIS

Emre YILMAZ

**Aksaray University
Graduate School of Natural and Applied Sciences
Department of Geomatic Engineering**

Supervisor: Asst. Prof. Dr. Süleyman Sefa BİLGİLİOĞLU

ABSTRACT

In recent years, the use of open source software has been increasing worldwide and in our country. Open source software, in addition to being more reliable by users, increases its popularity day by day thanks to its free nature and brings together all users of the world with the same purpose thanks to the voluntary communities established. The word Free in this software does not mean free but means freedom in the world. After the Free Software Foundation, the first corporate initiative to develop free software, in 1985, many associations and foundations have been established over the years. Founded in 2006, the Open Source Geography Foundation (OSGeo) developed QGIS, an open-source GIS software. QGIS software allows users to build their own plug-in tools with open source python software support. In contrast, Qt Designer software, which allows plug-in developers to design graphical user interfaces (GUI), allows the development of different design architectures. This study aims to create a free GIS plug-in using multi-criteria decision-making (MCDM) methods with the help of QGIS software. In this context, the interface design was made with the Qt Designer program and the geographic tools included in the plug-in were coded using the QGIS API using the python programming language and the PyQGIS adapter used in plug-in development. The communication between the geographical tools and the interface was coded with signal-slot connections and the design was completed. It is aimed to contribute to the development of free software by sharing the designed plug-in with all users of the world free of charge.

Keywords: Geographic Information System (GIS), Open Source Software, Python, QGIS, Plugin.

June, 2022; 60 pages

ŞEKİLLER DİZİNİ

Şekil 3.1. Plugin Builder 3 eklentisinin yüklenmesi.....	33
Şekil 3.2. Yeni eklenti üretimi.....	33
Şekil 3.3. Yazılımın derlenmesi.....	34
Şekil 3.4. Eklenti menüleri.....	35
Şekil 3.5. Eklenti menü akış diyagramı.....	36
Şekil 3.6. Plugin Builder3 ile üretilen .ui dosyasının Qt Designer'daki ham hali.	36
Şekil 3.7. Clip sekmesi.....	38
Şekil 3.8. Dem sekmesi.....	39
Şekil 3.9. Grid veriye dönüşüm menüsü sekmeleri; a) aspect, b) slope.....	39
Şekil 3.10. Grid veriye dönüşüm menüsü sekmeleri; a) rasterize (attribute), b) rasterize (pixelvalue).....	40
Şekil 3.11. Proximity sekmesi.....	40
Şekil 3.12. Matris hesaplayıcı sekmesi.....	41
Şekil 3.13. Reclassify sekmesi.....	42
Şekil 3.14. Raster calculator sekmesi.....	42
Şekil 3.15. Üretilen .py dosyasında hazır gelen modüller ve sınıflar.....	43
Şekil 3.16. Eklenti için kullanılan modüller ve sınıflar.....	44
Şekil 3.17. Dosya_ac() fonksiyonu kodları.....	46
Şekil 3.18. Vektor_sec() ve raster_sec() fonksiyonları kodları.....	47
Şekil 3.19. Slope() fonksiyonu kodları.....	48
Şekil 3.20. Slope() fonksiyonu signal-slot bağlantıları.....	49

ÇİZELGELER DİZİNİ

Çizelge 2.1. İkili karşılaştırma matrisi için kullanılan ölçek	20
Çizelge 2.2. Rastgele indeks	23



SİMGELER VE KISALTMALAR

AHY	Analitik Hiyerarşi Yöntemi
AKK	Açık Kaynak Kod
CBS	Coğrafi Bilgi Sistemi
CR	ConsistencyRatio
CSS	Closed Source Software
ÇÖKV	Çok Ölçütlü Karar Verme
DBMS	Data Base ManagmentSystem
DEM	DigitalElevation Model
GDAL	Geospatial Data Abstraction Library
GUI	Graphical User Interface
İHA	İnsansız Hava Aracı
KKY	Kapalı Kaynak Yazılım
MTA	Maden Teknik Arama
OSGeo	Open Source Geospatial Foundation
ÖAKK	Özgür Açık Kaynak Kod
SAM	Sayısal Arazi Modeli
SQL	Structure Query Language
TBMM	Türkiye Büyük Millet Meclisi
Tİ	Tutarlılık İndeksi
TO	Tutarlılık Oranı
VTYS	Veri Tabanı Yönetim Sistemi
WMS	Web Map Server

1. GİRİŞ

Dünyada ve ülkemizde teknolojinin gelişimi Coğrafi Bilgi Sistemleri (CBS)'nde kullanılan yazılımların çeşitliliğini buna bağlı olarak da yazılım tercihindeki önemini giderek daha da arttırmıştır. Geçmiş yıllarda CBS projelerinde kullanılan yazılımlar ticari amaçlı, kısıtlı sayıda araca sahip, mekansal analiz kabiliyetleri düşük buna karşın satın alma maliyetleri çok yüksek yazılımlardır (Yılmaz, 2006). Günümüzde kullanıcıların ücretli lisanslı yazılımlardaki kısıtlamalara olan tepkisi, yazılımın açık kaynak kod ve kapalı kaynak kodlu yazılımlar olarak ikiye ayrılması sonucunu ortaya çıkarmıştır. Bu durum CBS yazılımlarını da etkilemiş Açık Kaynak Yazılım (AKY) ve Kapalı Kaynak Yazılım (KKY) olarak iki şekilde gelişimine sebep olmuştur.

KKY olarak bilinen yazılımlar, başka kişilerin yazılımı kullanmak, değiştirmek veya değişiklikleri paylaşmak için yayıncısı tarafından bazı lisans haklarını saklı tuttuğu ve kullanıcısının ücret ödediği özgür olmayan yazılımlardır. Kapalı kaynak kod yazılımların ticari amaçlarla tasarlanmaları ve güvenilirliklerinin tartışılır olması tüm dünya üzerindeki özgür yazılım geliştiricilerini alternatif yazılımlar geliştirmeye teşvik etmiştir (Yılmaz ve Tarhan, 2019). Özgür yazılım olarak Türkçe ye çevrilmiş ifade, dünyada free software olarak bilinmektedir. Free kelimesi özgürlük (freedom) kelimesinin taşıdığı özgür anlamında kullanılmaktadır. Bu açıdan açık kaynak yazılımları özgür yazılımlar, kullanıcıları ise özgür yazılım kullanıcıları olarak nitelendirilmektedir (Hanbay-Tiryaki ve Balaman, 2021). Açık kaynak kod yazılımlarının ücretsiz ve güvenilir olmaları kullanımlarını giderek arttırmıştır.

Dünyada açık kaynak kod tanımı ilk defa Richard Stallman'ın Kapalı kaynak kod ile tasarlanmış Unix işletim sistemine karşı 1983 yılında GNU projesi ile başlamıştır. Özgür yazılımın gelişimi için organizasyonel ilk girişim ise 1985 yılında yine Richard Stallman tarafından Özgür Yazılım Vakfı (Free Software Foundation) kurulması olmuştur. 1989 yılında Stallman tarafından GNU için ücretsiz yazılım lisansı Genel Kamu Lisansı (General Public License) tasarlanması açık kaynak kodlu yazılımların popülerliğini daha da arttırmıştır (Stallman, 2003).

2000 yılından sonra kurulan birçok vakıf kar amacı gütmeyen açık kaynak yazılımın gelişimine katkı sağlamıştır. Bunlara sürdürülebilir ekosistemler oluşturmayı hedefleyen Linux vakfı, bulut bilişimin gelişimini amaçlayan Cloud Foundry, Eclipse,

Open Stack gibi vakıflar örnek olarak gösterilebilir (Ölmez ve Dai, 2016). Özgür yazılımın gelişmesi için 2006 yılında kurulan bir diğer vakıf Açık Kaynak Coğrafya Vakfı (OSGeo)'dır. OSGeo vakfı açık kaynak CBS yazılımları geliştirmek amacıyla kurulmuştur. Bu vakıf tarafından geliştirilen QGIS yazılımı, GNU Genel Kamu Lisansı ile lisanslandırılmış açık kaynak CBS yazılımıdır. QGIS, OSGeo'nun resmi projesidir. Linux, Unix, Mac OSX, Windows ve Android üzerinde çalışabilmekte ve çok sayıda vektör, raster ve veritabanı biçimlerini ve işlevlerini desteklemektedir (URL-1). Tasarımcıları QGIS yazılımını geliştirirken esnek bir tasarım mimarisine sahip olacak biçimde, istenildiğinde kullanıcılar tarafından istekleri doğrultusunda özgürce kendi eklentilerini geliştirebilecekleri şekilde tasarlamışlardır. Kullanıcıların geliştirerek diğer kullanıcıların kullanımına sunduğu eklentilerin sayısı giderek artmaktadır (Congedo, 2013; Jung, 2013; Vitalis vd., 2020; Ellsäßer vd., 2020; Nielsen vd., 2021). QGIS yazılımı ile geliştirilen eklentilerin diğer kullanıcılarla paylaşılabılır olması tüm dünya üzerindeki açık kaynak kodlu coğrafi yazılım kullanıcılarını ortak hedeflerde buluşturmaktadır.

Dünya üzerindeki geliştiriciler birçok farklı amaç için eklentiler geliştirmiş olmalarına rağmen karar vericilerin mekan ile ilgili alınacak kararlarına yardımcı olacak Çok Ölçütlü Karar Verme (ÇÖKV) alanında kısıtlı sayıda eklenti çalışması vardır. Günümüzde kamu ve özel kuruluşlarda ÇÖKV yöntemleri kullanılarak mekansal analizler yapmak yaygın hale gelmiş ancak kullanılan CBS yazılımlarında bu işlemler hala karmaşık bir dizi işlemten geçtikten sonra yapılabilmesi kullanıcılar açısından problem olarak devam etmektedir. Ülkemizde son yıllarda ÇÖKV yöntemleri kullanılarak yapılan tez, makale ve yayınların sayısı oldukça artmış olmasına karşın literatürdeki çalışmalar incelendiğinde çalışmayı yapanlar açısından iki temel problem tespit edilmiştir. İlk olarak kullanılan CBS yazılımlarının KKY olmasından kaynaklı maliyet problemi, diğer problem ise kullanılan açık kaynak veya kapalı kaynak CBS yazılımı ile ÇÖKV yöntemleri kullanımında birçok araç ve işlem kalabalığının kullanıcı için karmaşaya ve zaman kaybına sebep olmasıdır. Bu kapsamda kullanıcıların bir CBS programının kullanımını öğrenmeden ÇÖKV yöntemlerini kullanarak analizlerini yapabileceği bir eklenti (plugin) tasarımı geliştirmek amaçlanmıştır. Bu çalışmada QGIS yazılımı içinde ÇÖKV'nin doğrudan kullanılabildiği, diğer kullanıcılarla rahatlıkla paylaşılacakları ve açık kaynak kodlu olması sebebiyle diğer tasarımcılar tarafından da daha fazla geliştirilebilecek

yeni bir eklenti tasarlanmıştır. Ara yüz tasarımı kullanıcıların grafiksel kullanıcı ara yüzleri tasarlayabilmesine olanak sağlayan Qt Designer programı ile yapılmıştır. Eklentide yer alan coğrafi araçlar python programlama dili kullanılarak kodlanmıştır. Coğrafi araçların kodlanmasında python programlama diline ek olarak, python ile QGIS API'sini kullanarak yeni bir CBS yazılımı geliştirmede kullanılan PyQGIS bağdaştırıcısından yararlanılmıştır. Ara yüz tasarımı ve coğrafi araçlar arasındaki ilişkiler, Qt kütüphanesi ile python programlama dilini bir araya getiren PyQt modülü sayesinde kurulmuştur. Son olarak signal-slot bağlantıları ile tasarım tamamlanmıştır. Eklenti tasarımı tamamlandığında QGIS eklenti deposuna eklenmesi planlanmaktadır. Böylece kullanıcı QGIS yazılımına eklentiye içe aktararak kolaylıkla kullanımı amaçlanmıştır.



2. TEMEL TANIM VE KAVRAMLAR

2.1 Coğrafi Bilgi Sistemleri

CBS; öznitelik, raster ve vektör veriler gibi konuma dayalı ve konuma dayalı olmayan her türlü dijital veri türlerinin toplanması, depolanması, analiz edilmesi, işlenmesi ve sunulması görevlerini yerine getiren gelişmiş bir sistem olarak tanımlanmaktadır (Yomralıoğlu, 2000). CBS tanımı, sistemi kullanan kurumun ihtiyaçlarına göre mekansal verinin toplanması, depolanması, işlenmesi ve gösterimini yapan, karar destekleme işlevi olan, sayısal bir bilgi sistemi biçiminde yapılabilir (Uluğtekin ve Doğru, 2005).

CBS, dünyada meydana gelen olayları haritalamak ve analiz etmek için bilgisayar tabanlı bir araçtır. CBS teknolojisi, sorgu ve istatistiksel analiz gibi yaygın veri tabanı işlemlerini haritaların sunduğu görselleştirme ve coğrafi analiz avantajlarıyla bütünleştirir. Bu yetenekler, CBS'yi diğer bilgi sistemlerinden ayırır ve olayları açıklamak, sonuçları tahmin etmek ve stratejiler planlamak için çeşitli kamu ve özel kuruluşlar için değerli kılar. Harita yapımı ve coğrafi analiz yeni değildir, ancak CBS bu görevleri eski manuel yöntemlerden daha iyi ve daha hızlı gerçekleştirir. Günlük ihtiyaçlardan bilimsel araştırmalara kadar pek çok alanda gerçekleştirilen faaliyetlerin büyük ölçüde mekansal verilerin kullanımıyla ilişkili olması, CBS teknolojisinin büyük bir hızla gelişmesine olanak sağlamaktadır (Çabuk, 2015).

2.2 CBS Bileşenleri

Vankova vd. (2022) değerlendirmelerine göre halihazırda CBS'nin uygulanabileceği tüm alanlara uygun, genel kabul görmüş bir tanım bulunmamaktadır. Yine de her tanım veri, donanım, yazılım ve organizasyon yapısı gibi temel bileşenleri içermekte ve temel fonksiyonların ise verilerin girişi, analizi, yönetimi ve sunumu ile temsil edildiğini belirtmektedir. Güncel tanımlar incelendiğinde CBS'nin tanımı kişinin çalışma alanı ve bakış açısına göre farklı yapılırken, tanımının gelişmelere paralel olarak zamanla değiştiği ve CBS'nin temel işlevlerini yerine getirebilmesi için beş temel bileşenin bir araya getirilmesi gerektiği görülmüştür. Bunlar donanım, insan, veri, yazılım ve yöntem olarak beş ana başlık altında toplanmıştır (Kapluhan, 2014).

CBS'nin fonksiyonlarını yerine getirmesini sağlayan bilgisayar ve buna bağlı yan ürünlerin tamamı donanım olarak adlandırılmaktadır. Günümüzde CBS yazılımları, merkezi bilgisayar sunucuları ile bağımsız veya ağ bağlantısı yardımıyla kullanılan masaüstü bilgisayarlar gibi çok çeşitli donanım türleri şeklinde çalışmaktadır. Oluşturulması planlanan bilgi sistemlerinde ana donanımın bilgisayar olmasına rağmen bilgisayara gelen verinin toplanması, işlenmesi ve çıktı haline getirilmesi bilgisayara bağlı birtakım donanımların olmasını zorunlu hale getirmektedir. Teknolojinin gelişimiyle beraber bilgisayarların bulunduğumuz yüzyılda artan kullanımı, bilgisayara bağlı teknolojilerinde gelişimini arttırmış ve CBS de kullanımı da hızla yaygınlaşmıştır. CBS teknolojisi içerisindeki yazılım ve donanımlar, bilgisayar teknolojisindeki gelişmeleri takip edebilecek ve işlevselliğini geliştirecek şekilde seçilmelidir. Bu sebeple kurulması düşünülen bilgi sistemlerinde yer alacak yazılım ve donanımlar zaman içerisinde yenilenebilir özellikte olmalıdır (Güney ve Çelik, 2004). Donanım gereksinimleri, kurumsal gereksinimlere göre büyük ölçüde değişebilir. Güvenli tesisler, merkezi sunucular ve dizüstü bilgisayarlar veya masaüstü bilgisayarlar, tarayıcılar, yazıcılar vb. arasındaki intranet bağlantılarını kullanabilir. Diğer kuruluşlar, merkez ve uzak konumlar arasında iletişimi ve iş birliğini kolaylaştıran yüksek hızlı internet bağlantılı cihazlar aracılığıyla CBS kullanabilir. Akıllı telefonlar, tabletler ve diğer mobil bilgi işlem cihazları da CBS teknolojilerini giderek daha fazla kullanılmaktadır. CBS özellikle bilişim teknolojilerinde yaşanan gelişmeler ile web ve mobil platformlarda yaygın olarak kullanılmaktadır (Taşyürek, 2021). CBS'nin bu kadar hızlı gelişmesinin ve giderek popüler olmasının ardındaki sebep; bilgisayar ve çevre birimlerinin maliyetlerinin hızla düşmesinin yanı sıra donanım endüstrisinin hızlı gelişimidir (Akça, 2000).

CBS teknolojisi, sistemi yöneten ve onu gerçek dünya problemlerine uygulamak için planlar geliştiren kişiler olmadan sınırlı bir değere sahiptir. CBS kullanıcıları, sistemi tasarlayan ve bakımını yapan teknik uzmanlardan, günlük işlerini yapmalarına yardımcı olmak için kullananlara kadar çeşitlilik gösterir.

1980'lerden itibaren CBS teknolojisi özellikle mekansal verilerin elde edilmesinde, yönetilmesinde ve analiz edilmesinde yeni bir bilgi işleme teknolojisi olarak belirmiştir (Erden ve Coşkun, 2010). CBS'de veri; türlerine göre geometrik ve öznitelik verisi olmak üzere iki ana başlıkta tanımlanır. Geometrik veri detayın belirli

bir referans sistemine göre yerini ve biçimini iki veya üç boyutlu gösteren koordinat veya piksel değerlerdir. Öznitelik veri ise detayın konum verisine ait tanımsal bilgisi olarak tarif edilir (Yomralıoğlu, 2000). Coğrafi verilerin üretimi, CBS bileşenleri arasında en çok zaman ve maliyet gerektiren işlemi oluşturmaktadır (Çetinkaya ve Toz, 2011). CBS’de dijital ve dijital olmayan biçimlerde elde edilen veriler çeşitli işlemlerden geçirilerek CBS programlarının kullanımına hazır hale getirilir. Dijital veriler CBS programının kullanıma uygun şekilde programa aktarılması gerekmektedir. Kağıt üzerinde yer alan çeşitli haritalar, paftalar, dökümanlar, evraklar vb. veriler ise öncelikle dijital hale getirilmeli daha sonra CBS programına aktarılmalıdır. CBS çalışmasının süre ve önem olarak yarıdan fazlasını oluşturan bu sürecin sonunda veriler analize uygun hale getirilir. Sonraki adım olan veri analizi, amaca yönelik esas çalışmaların yapıldığı bölümdür (Kapluhan, 2014).

CBS birçok bileşenin aynı anda sekronize çalışabilmesi ve yönetilebilmesi ile başarılı olabilmektedir. Proje öncesi, proje evresi ve proje sonrasında tüm süreçlerin başarılı yönetilmesi izlenecek yöntemin doğru belirlenmesi ile mümkün olabilmektedir. Bir CBS seçimi, basit bir donanım-yazılım satın alma sorunu değildir. CBS yaklaşımı bilgi hakkında yeni bir düşünme yöntemi gerektirir (Akça, 2000). Belirlenen yöntemin başarılı seçimi projenin başarılı olmasını sağlarken aynı zamanda kendinden sonra geliştirilecek bilgi sistemlerinin başarıya ulaşmasında da katkı sağlayacaktır. Bilgi sisteminin tasarımında amaç; daha önce zor ve karmaşık yöntemlerle sistematik bir şekilde analizi mümkün olmayan olay ya da olgulara bir sistem mimarisi kazandırarak karmaşıklığın giderilmesi iken yöntem ise; zaman, para, kaynaklardan tasarruf ederek, verinin en etkin şekilde kullanımının yanında belli kalite ve standartlarda ürünün ortaya konulmasıdır. Maliyet tahmini CBS projesinin başarılı veya başarısız olmasını belirleyen çok önemli bir aşamadır. CBS projelerinin sağladığı büyük kolaylık ve yararların değerini anlamak amacı ile bir çalışma yapılmış ve ortaya pek çok örnek hakkında CBS kullanımının getirileri ve sağladığı tasarruf miktarlarının hesaplandığı kullanışlı bir çalışma çıkarılmıştır. Bu çalışma the Joint Nordic Project Report- isimli bir rapor ile sunulmuştur ve konuya iyi bir bakış sağlamaktadır (Güney ve Çelik, 2004).

2.3 Yazılım

Bilgisayar yazılımı, CBS'nin bir diğer önemli bileşenidir. CBS yazılım programları veya uygulamaları ile CBS verilerini depolama, analiz etme ve görselleştirme işlemleri gerçekleştirilmektedir. Temel CBS yazılım uygulamaları, Veri Tabanı Yönetim Sistemlerini (VTYS) ve kullanıcıların CBS verilerini sayısallaştırmasına, depolamasına, yönetmesine ve sorgulamasına, karmaşık analizler yapmasına ve raporlar, çizelgeler, haritalar üretmesine olanak tanıyan menü seçeneklerine sahip bir grafik kullanıcı arabirimi veya gösterge tablosunu içerir. Yazılımlar, kullanım kolaylıkları ve geliştirilmiş arayüzleri ile kullanıcılarının, yüksek bilgisayar bilgisine gerek duymadan, bilgisayardan istenen verimi alabilmelerini sağlamıştır (Akça, 2000).

CBS teknolojilerinin gelişimi sırasında kullanıcılar ve sistem tasarımcıları temin edilmesi maliyetli olmasına rağmen yaygın ve güvenli olması gibi sebeplerle lisanslı, kapalı kaynak kodlarla tasarlanmış ticari paket yazılımlar kullanmaktaydılar. Ancak teknolojiye hızlı gelişim CBS nin diğer bileşenleri olan veri ve donanımda hızlı bir gelişmeye sebep olurken buna ek olarak yazılım bileşenini de hızla etkilemiş ve her geçen gün artan bir ivme ile yeni ve bir diğerine göre daha etkili olabilecek yazılımları beraberinde getirmiştir. CBS nin içinde bulunduğu yazılım pazarına devamlı, yeni teknolojiler ve bilgiler kullanan CBS yazılımları sürülmüştür (Köktürk, 2003).

Yeni teknolojilerin yıllar içinde gelişimi sonucunda modern CBS teknolojisinde kullanılan bilgisayar yazılımları, sabit haritalarda çeşitli harita türlerinin görselleştirilmesiyle sınırlı olmaktan çıkmıştır. Günümüzde çeşitli varlık gruplarının; kartografik metodolojiyi ve en son bilgi ve iletişim teknolojileri çözümlerini kullanan CBS yazılım üreticileri, mekansal veri sağlayıcıları, CBS teknolojisinin sosyal, ekonomik ve politik yaşamın seçilmiş alanlarında uygulanmasının organizatörlerinin işbirliği yaptığı belirli bir ekonomi dalı olarak görülmektedir (Turek ve Stępnia, 2021). CBS de kullanılan paket yazılımlar CBS yazılımları ve veri tabanı yazılımları olarak iki ana başlıkta incelenebilir. Veri tabanı yazılımları bu tez çalışmasının konusu olmadığından irdelenmemiş, CBS yazılımları konusu detaylı olarak incelenmiştir.

CBS uygulamalarında kullanılan paket yazılımlar yaygın olarak; veri işleme yöntemlerine göre, vektör verileri işleyen sistemler, raster verileri işleyen sistemler ve vektör-raster verilerini işleyen karma sistemler olarak üçe ayrılmaktadır (Alkış, 1996).

Karataş ve Kırbaş (2015) yaptıkları çalışmada CBS yazılımlarının; haritalama işlemlerinde, yönetim planlamalarında, afet yönetimi ve suç analizi gibi birçok farklı alanda kullanıldığını belirtmiştir. Stępniaak ve Turek (2020) yaptıkları çalışmada ise modern CBS araçlarının sadece coğrafi değil, aynı zamanda soyut alanları da haritalayabildiğini belirtmiştir. Soyut haritalama kavramını; aynı CBS aracına aynı anda birçok kullanıcı tarafından çevrimiçi olarak erişebilmesi olarak tanımlamışlardır. Geçmişte CBS, öncelikle dünya yüzeyi ve doğası ile ilgili verilerin analizi, elde edilmesi ve görselleştirilmesi için kullanılıyordu. Bununla birlikte, CBS son yıllarda veri yönetiminde uygulama kazanmıştır (Vankova vd., 2022).

Ülkemizde ve dünyada en çok kullanılan CBS paket yazılımlarına örnek olarak; ArcGIS Pro, ArcGIS Desktop, QGIS, Hexagon Geomedia, MapInfo Professional, Feature Manipulation Engine, Global Mapper, gvSIG, GRASS GIS, WhiteBoxGAT, Cadcorp, GE Smallworld, Manifold GIS, TatukGIS, Maptitude, AutoCADMap 3D, Golden Software MapViewer, ILWIS, SAGA GIS, GeoDa, BentleyMap, IDRISI TerrSet, TNT GIS, MapWindow, uDig, Jump GIS, FalconView, OrbisGIS, Diva GIS yazılımları örnek gösterilebilir.

İlerleyen kısımlarda yazılımlar açık ve kapalı kaynak kod kullanımlarına göre de ayrıca incelenmiştir.

2.3.1 Kapalı kaynak kodlu yazılım

KKY, halka dağıtılmayan özel mülkiyetli yazılımlardır. Bu yazılımlar şifrelidir. Bu nedenle, kaynak kodunu yasal olarak kopyalama, değiştirme, güncelleme ve düzenlemeye ilişkin münhasır haklara yalnızca kodu oluşturan ilk yazarlar sahiptir. Geliştirici genellikle satın alma sonrasında kullanıcılara destek sağlar ve yazılımın beklendiği gibi çalışmasını sağlar (URL-2). KKY'da kullanıcılar geliştiricinin yazılım üzerindeki verdiği yetkiler ile özellikleri kullanmakta sınırlandırılabilir. Kullanıcı yazılımda farklı bir özellik isterse veya var olan özellik de değişiklik isterse geliştirici ile iletişime geçip bir ücret karşılığı bu işlemleri yaptırması gerebilir. KKY bu yönden kullanıcıları geliştiricilere bağımlı kılar (Gündüz, 2019).

2.3.2 Açık kaynak kodlu yazılım

AKY, kaynak kodları özel bir telif hakkı lisansı ile isteyen herkesin incelemesine, kullanımına ve dağıtımına sunulan yazılımlardır. Bu sayede kullanıcıya yazılımı değiştirme ve değiştirdiği haliyle tekrar dağıtım özgürlüğü de sunduğu için çeşitli platformlarda dünyanın her tarafından bir araya gelen gönüllü yazılımcılar, imece yöntemi ile endüstri standartlarında geliştirilmeye devam etmektedir (Yılmaz, 2017).

Açık kaynak yöntemi, açık kaynak topluluğu içindeki bir düşünme ve iş birliği biçimidir. Bu düşünme ve iş birliği hareketi, küresel bir topluluğun hem bireysel hedeflerinde hem de grup hedeflerinde kaynak kod aracılığıyla iş birliği yapması, bunları paylaşması ve desteklemesi için bir yöntem oluşturmuştur (URL-2). Gönüllü kullanıcıların açık kaynak yazılımların gelişmesine katkıda bulunmaları hem yazılım güvenliği açısından kullanıcılarının yazılımdan emin olmalarını hem de yazılımın yeteneklerini, niteliğini arttırmaktadır. Ayrıca AKY'lara kullanıcılarının sağladığı katkılar, kullanıcıların birbirleri ile etkileşimlerini ve iletişimlerini de arttırmaktadır (Hanbay-Tiryaki ve Balaman, 2021).

Açık kaynak kod tanımı ilk defa Richard Stallman'ın KKY ile tasarlanmış Unix işletim sistemine karşı 1983 yılında geliştirmeye başladığı ve 1984 yılında tamamladığı GNU projesi ile başlamıştır. GNU'nun açılımı "GNU is not Unix" yani "GNU Unix değildir" şeklinde tasarlanmıştır. Tasarım Unix ile birçok benzerlik göstermesine rağmen tasarımı aşamasında kullanılan çekirdek, sistem araçları, kütüphaneleri ve son kullanıcı yazılımlarının her biri GNU projesi kapsamında geliştirildiği için tamamen özgür bir yazılımdır ve Unix'e ait hiçbir kod bulunmamaktadır (Canbaz ve Erdemir, 2021). Genel Kamu Lisansı (GPL) 1989 Yılında Stallman tarafından GNU için tasarlanmıştır. GPL birçok yerde kullanılan ücretsiz yazılım lisansıdır ve özgün hali Richard Stallman tarafından GNU projesi için yazılmıştır (Şen vd., 2010).

Özgür açık kaynak kodlu (ÖAKK) yazılımlardaki değişim kapalı kaynak yazılıma olan ilgi ve güven giderek azalmış, ÖAKK yazılımlar dünyanın her tarafındaki gönüllü yazılım uzmanlarının paylaşımlarıyla gelişerek birçok alanda kullanılmaya başlanmıştır. Peru, İsrail, İspanya ve Almanya gibi ülkeler; çeşitli organizasyonlar, ticari firmalar ve hatta devlet politikalarıyla kabul edilen AKK'lı yazılımları

desteklemektedir. KKY’lardan lisans maliyetleri ve güvenlik tehditi nedeni ile kaçınılmaktadırlar (Şahinsoy, 2011).

Son yıllarda açık kaynak kodlu yazılımların hızlı gelişimi ve buna bağlı artan kullanımı yazılım maliyetini ciddi oranda düşürdüğü için her türlü ihtiyaca yönelik ekonomik çözümlerin bulunabilmesini kolay ve pratik hale getirirken, daha çok ihtiyacın çözümü için küçük bütçeye sahip birçok tasarımcıya da ilham olmaktadır. Küçük bütçeye sahip birçok tasarımcıya ek olarak dünya üzerindeki devletlerinde kamusal ölçekte açık kaynak kodlu yazılımlara yöneldiği bilinmektedir. Ülkemizde de TBMM’nin 2013 tarih ve 1041 Karar no ile onaylanan Onuncu Kalkınma Planının (2014-2018) 412. Maddesinde “Açık kaynak kodlu yazılımlar, büyük veri, bulut bilişim, yeşil bilişim, mobil platform, nesnelerin interneti gibi ürün, hizmet ve yönelimler değerlendirilerek kamu için uygun olabilecek çözümler hayata geçirilecektir.” denilerek açık kaynak kodlu yazılımların kullanımına başlayacağı vurgulanmıştır (Kalkınma Bakanlığı, 2013). Buna örnek olarak, MTA Uzaktan Algılama Servisi tarafından hazırlanan portalda bulunan haritalar ve veriler belli bir kurala bağlı olarak AKK’lı QGIS yazılımı ile birlikte kullanılmaktadır. MTA’nın Web Map Server (WMS) adresi bilindiğinde QGIS’in bu veriye erişim özelliği kullanılarak ulaşılabilmektedir (Ayday vd., 2016).

Son yıllarda AKY’lar ticari yazılımlara nazaran daha kaliteli ürünler üreterek karşımıza çıkmaktadır. Bunun en önemli sebeplerinden bir tanesi AKY’ların dünyanın dört bir tarafından yazılım geliştiricilerin dikkatli incelemesinden geçmesi ve hatalarından arındırılmış olmasıdır (Yılmaz, 2017). Kapalı kaynak kod yazılımların maliyeti, ulaşılabilirliğine karşın, açık kaynak kodlu yazılımlar her geçen gün önemi artmaktadır (Onyıl ve Ulukavak, 2021).

Özgür yazılımlar, tüm kullanıcılara, yazılımı istediği gibi çalıştırma, dağıtma, değiştirme ve değişmiş halini tekrar dağıtma özgürlüğü sağlamaktadır (Kepoğlu ve Düzgün, 2008). Özgür yazılım dünyada Free Software olarak bilinmektedir. Buradaki Free kelimesi türkçe deki bedava anlamıyla değil, freedom (özgürlük) kelimesindeki özgür anlamında kullanılmaktadır. Özgür yazılımın gelişimi için ilk defa 1985 yılında Richard Stallman tarafından Özgür Yazılım Vakfı kurulmuştur. Vakıf herhangi bir kar amacı taşımaksızın bilgisayar yazılımları oluşturmak, düzenlemek ve dağıtımını sağlamak için kurulmuştur. Özgür Yazılım Vakfı’nın kurucusu Richard Stallman’a göre bir yazılımın özgür yazılım olabilmesi için kullanıcıların yazılımı kullanma,

kopyalama, dağıtma, geliştirme, yazılım üzerinde çalışma ve değişiklik yapma özgürlüğünün bulunması gerekir. Buna göre yazılım kullanıcılarının dört temel özgürlüğe sahip olması gerekmektedir:

- Amaçları doğrultusunda yazılımı geliştirme özgürlüğü (Özgürlük 0).
- Yazılımın kaynak koduna erişim ile yazılımın nasıl çalıştığını inceleme ve kendi gereksinimleri doğrultusunda değiştirme özgürlüğü (Özgürlük 1). Kaynak koduna erişim bu yaklaşım için bir ön şarttır.
- Yeniden dağıtma ve topluluklarla paylaşabilme özgürlüğü (Özgürlük 2).
- Yazılımı geliştirme ve yazılımın gelişmiş hali ile topluma dağıtma özgürlüğü (Özgürlük 3) (Şahinsoy, 2011).

Lisanslı yazılımlar ticari amaçlı üretilmeleri sebebiyle bir bilgi sistemi tasarlamak için yazılıma ait birçok araç ya da eklentiye de yine yazılımın beraberinde satın almak gerekmektedir. Programcılar kapalı kaynak kodlu yazılımlara yeni özellikler ekleseler dahi, geliştirdikleri bu eklentileri yasal olarak yeniden dağıtmaya hakları yoktur. Buna ek olarak kaynak kodlara yetkisiz erişim kanunlarla da yasaklanmıştır. Bu durum lisanslı yazılıma bağımlılığı arttırırken, özgürce değişiklik yapamama problemlerini doğurmaktadır. Bu problemlerin yarattığı olumsuzluklara alternatif olarak açık kaynak kodlu yazılımların hızlı gelişimi tasarımcıları bu maliyetsiz yöntemle teşvik etmiş, yazılım maliyetindeki bu düşüş ise tasarımcılara kendi özgür yazılımlarını geliştirebilme imkanı sunduğu için problemleri daha geniş bir perspektiften inceleyebilme kabiliyeti kazandırmıştır.

2.4 Açık Kaynak Kodlu CBS Yazılımlarına Eklenti Geliştirilmesi

Kapalı kaynak CBS yazılımları, kullanıcılarına yazılımın sunduğu ön tanımlı araçlar ile analizlerini gerçekleştirmesine müsaade etmektedir. Kapalı kaynak kodlu yazılımlar ticari amaçlı yazılımlar olduğu için KKY geliştiricileri kullanıcıların analizlerini gerçekleştirmesi için bazı araçları da satın almaya zorlamaktadır. Ücretli ve kapalı kaynak CBS yazılımları ile kıyaslandığında, ÖAKK CBS yazılımlarının her geçen yıl oldukça büyük bir gelişme içerisinde olduğu, hatta gelişmiş pek çok ücretli ve kapalı kaynak CBS yazılımından çok daha fazla işlev ve özellik barındırdığı gözlenmektedir (Beyhan vd., 2010). Özgür yazılım dünyasındaki çeşitlilik kullanıcılarına belirli bir ücret ödemek yerine ücretsiz olan benzer yazılımları tercih

edebilme avantajı sağlamaktadır. Yazılım karşılığında ücret talep edilmesi bir süre sonra projenin yaygınlaşmasını engelleyerek ve yazılımın geliştirilmesini yavaşlatarak olumsuz etkilemektedir (Kepoğlu ve Düzgün, 2008).

Geçtiğimiz yıllarda yalnızca satın alınarak ve kullanımı yine belirli ücretler karşılığında üretici firmalar tarafından öğretilen kapalı kaynak kodlu yazımlar küçük bir tasarımcı kitlesi tarafından kullanılabilmekteydi. Günümüzde tasarımcı profili hızla değişmekte, her yaştan ve meslekten yeni tasarımcılar belirli topluluklar kurarak bir araya gelmekte ve ihtiyaç duyulan alanları keşfetmeye çalışmaktadır. ÖAKK CBS topluluklarının genellikle gönüllülük esasına bağlı çalışmaları, özgür yazılımların ömürlerini belirli bir şirkete ya da bireye bağlı kalmaktan kurtarmakta bu ise çok daha uzun süreler boyunca geliştirici toplulukların kullanıcılarına destekler verilebilmesini sağlamaktadır (Beyhan vd., 2010). Kullanıcı profilindeki bu değişim ve sayılarındaki artış buna ek olarak ta internet üzerinden kurulan topluluklar, eskiden lokal olarak çözümü aranan birçok ihtiyaca yönelik çözüm önerilerini de geliştirdikleri açık kaynak kodlu yazılımlar ile tüm dünya kullanıcılarına yaymaktadırlar. Ayrıca kullanılan farklı CBS mimarileri sayesinde gelişmiş performans, kolay kullanım, veri yönetimi ve ölçeklenebilirlik sağlanmaktadır. Cep telefonundan, mobil ve masaüstü kullanıcılarına kadar değişik platformlardaki kullanıcılar CBS fonksiyonlarından kolaylıkla faydalanabilmektedir (Kapluhan, 2014).

Gönüllü yazılımcılar, ÖAKK CBS yazılımlarının hemen hemen tamamının gelişmesine değişik biçimlerde ve ölçülerde katkı sağlamaktadırlar. Özellikle bir takım uzmanlaşmış CBS özelliklerinin ve fonksiyonlarının eklenti biçiminde tasarlanmasında gönüllü yazılımcıların katkıları büyüktür (Beyhan vd., 2010). Özgür yazılım kullanıcıları Richard Stallman'ın özgür yazılım tanımında belirttiği gibi yazılımı kendi gereksinimleri doğrultusunda değiştirme özgürlüğüne sahip olmaları sayesinde analizlerini yapabileceği araçları kendileri geliştirebilir ve tüm dünya kullanıcılarına ücretsiz olarak yeniden dağıtma ve paylaşma özgürlüğüne sahiptir. Bu sayede AKK CBS yazılımları kullanıcılara göre şekillenebilmekte kullanıma göre eklentiler geliştirilebilmekte, diğer kullanıcılar tarafından geliştirilen eklentileri kullanabilmekte ve internet ortamında geliştiricilerin paylaştığı birçok bilgidен rahatlıkla faydalanıp uygulayabilmektedirler (Önder ve Kavzoğlu, 2019).

Özgür CBS yazılımları beraberinde açık kaynak programlama dili ile kullanıcıya doğrudan işlemlerini yerine getirebileceği bir konsol ekranı da sunmaktadır. Bu konsol yardımıyla kullanıcı küçük kod parçacıkları halinde işlemlerini yapabileceği gibi isteğe bağlı olarak beraberinde sunulan designer araçları ile karmaşık analizlerin yapılabileceği grafiksel kullanıcı arayüzlerine sahip eklentilerde geliştirebilmektedir. Bu geliştirme yeni bir CBS yazılımı geliştirmek olabileceği gibi kullanılan programın ara yüzü içinde yeni bir eklenti yazmak da mümkün olabilmektedir (Kepoğlu ve Düzgün, 2008). Açık kaynak CBS yazılımlarında en yaygın kullanılan programlama dili olan python'ın ana yapısı ile eklenti geliştirilirken kullanılan modül ve kütüphaneleri aşağıda detaylı olarak incelenmiştir. Ayrıca açık kaynak bir CBS yazılımı olan QGIS beraberinde sunulan grafiksel kullanıcı arayüzleri tasarımları geliştirmekte kullanılan Qt Designer yazılımı detaylı olarak incelenmiştir.

2.4.1 Python

Python programlama dili etkin, çoğu işletim sisteminde rahatlıkla kullanılabilen, kod yazılımı sade, okunabilirliği kolay, yüksek seviyeli, nesne tabanlı bir dildir. Standart kütüphaneleri ve modülleri ile birlikte özellikle bilimsel alanda en yaygın olarak kullanılan diller arasına girmiştir (Sarahoğlu vd., 2014). Python, dinamik semantik ile yorumlanmış, nesne yönelimli, üst düzey bir programlama dilidir (Yılmaz, 2019). Geçmişten günümüze incelendiğinde Python, ilk defa GuidoVan Rossum tarafından geliştirilmiş ve 1991 yılında, kaynak kodları 0.9.0 versiyonu olarak piyasaya sürülmüş açık kaynak kodlu bir programlama dilidir (Göçgün ve Aytuğ, 2021). 1994 yılında yayımlanan 1.0 sürümü eski sürümü olan 0.9.0. sürümünden farklı olarak reduce, filter, lambda, map gibi fonksiyonlara sahiptir. 1.0 sürümünden sonra 1.4 sürümünde karmaşık sayılar için dahili destek sağlamıştır. 1997 de 1.5 sürümü, 2000 yılında ise 1.6 ve 2.0 sürümleri yayımlanmıştır. Yayımlanan sürümlerle python'ın resmi sitesinden erişilebilmektedir. Python'daki hızlı gelişim neticesinde 2007 yılında QGIS 0.9'da python desteği ilk olarak tanıtılmıştır. Yayımlanan sürümleri ile birlikte kullanıcılarına mekansal analizler yapabilme kabiliyeti kazandırmıştır (Kepoğlu ve Düzgün, 2008). 14 Ocak 2022 yılında ise python son sürümü olan 3.10.2 yayımlanmıştır (URL-3).

Python, ilk kez geliştirildiğinden günümüze kadar, dünya genelinde kabul gören ve birçok alanda üzerinde uygulama geliştirilen bir dil olmuştur. Diğer programlama

dillerine göre kolay sözdizimine sahip olduğu ve kullanıcılar tarafından C/C++/ Java gibi dillerle kıyaslandığında kodların daha az satırda ve daha okunaklı yazıldığı gözlemlenmektedir (Ayvaz vd., 2016).

Python ile yazılan programların en büyük özelliklerinden birisi, C ve C++ gibi dillerin aksine, derlenmeye gerek duymadan çalıştırılabilirlerdir. Python’da derleme işlemi bulunmadığı için oldukça hızlı şekilde programlar geliştirilebilmektedir. (Fırat, 2013). Ayrıca, Java’ya göre daha küçük bir kod boyutuna sahiptir ve bu sebeple çok daha az bellek tüketir ve çok daha hızlıdır (Khoirom vd., 2020). Python ve R gibi dillerde kolaylıkla konumsal analizlerin gerçekleştirilebilmesi, açık kaynak teknolojilerin gerek eğitimde gerekse araştırmalarda yaygın olarak kullanımını desteklemektedir (Anbaroğlu ve Serhan, 2019).

Python programlama dilinin popülerliğindeki hızlı artışın yayımlanan sürümleri ile birlikte mekansal analizler yapma konusunda yüksek kabiliyetler kazanması özgür yazılımcıların python kullanımını arttırdığı gözlemlenmektedir. Mevcut CBS yazılımlarında uygulama veya bağımsız CBS araçları geliştirmek için kullanılan diğer dillerle karşılaştırıldığında python’ın, zengin kütüphaneleri, mekansal analizlere uygunluğu ve kolay anlaşılabilirliği nedeniyle tercih edildiği görülmektedir (Akın vd., 2020).

Programlama dilleri için popüler derecelendirme indekslerinden TIOBE (The Importance of Being Earnest) Index’in 2022 yılı ilk çeyreğindeki programlama dillerini popülerliklerine göre karşılaştırması incelendiğinde ilk 5 programlama dilinin sırasıyla Python, C, Java, C++, C# olduğu gözlemlenmiştir. Python %15,33 oy almış, ikinci sıradaki C ise %14,08 oy almıştır. PYPL (Popularity of Programming Language) Index’in 2022 yılı ilk çeyreğindeki programlama dillerini popülerliklerine göre karşılaştırması incelendiğinde ilk 5 programlama dilinin sırasıyla Python, Java, JavaScript, C/C++, C# olduğu gözlemlenmiştir. Python %28,52 oy almış, ikinci sıradaki Java ise %18,12 oy almıştır. Popüler kullanıcı platformu Stackoverflow ‘ın programlama dillerini her yıl karşılaştırdığı anket incelendiğinde Python; 2019 ve 2020 yıllarında SQL ‘in ardından 4. Sırada yer alırken 2021 yılında HTML/CSS’in ardından 3. Sırada yer almaktadır.

2.4.2 QGIS

Ücretsiz ve açık kaynak CBS yazılımlarının kullanıcılarının belirli gereksinimlerini karşılayarak çeşitli mekansal uygulamalar oluşturma ve özelleştirme imkanı vermesi popülerliğini arttırmıştır. QGIS, bu yazılımlar arasında en esnek olanlardan biridir. Kullanıcıların GIS yazılım ortamında geliştirmeler yapabildiği python kullanımı sayesinde mekansal uygulamalar geliştirmeye olanak tanır (Oxoli vd., 2016).

QGIS, GNU genel kamu lisansı altında dağıtılan kullanıcı dostu bir açık kaynak CBS yazılımıdır. QGIS OSGeo'nun resmi projesidir. Linux, Unix, Mac OSX, Windows ve Android üzerinde çalışır ve çok sayıda vektör, raster ve veritabanı biçimlerini ve işlevlerini destekler (Kuka ve Bushati, 2014). QGIS veri görselleştirme ve analiz araçları; esas olarak C++ koduyla yazılmıştır, ancak aynı zamanda python'da yazılmış API ve eklentiler aracılığıyla işlevlerini yerine getirir. QGIS, veri analizini geliştiren geniş bir eklenti deposuna sahiptir ve ayrıca diğer açık kaynaklı programlarda bulunan çeşitli görüntü işleme araçlarına erişim sağlar (Congedo, 2013). QGIS, tescilli muadilleriyle benzer işlevler ve özellikler içerir ve kullanıcıların mekansal verileri görüntülemesine, işlemesine ve oluşturmaya olanak sağlamaktadır. Çeşitli mekansal veri dosyası uzantıları olan; ESRI ShapeFiles, KML, GML gibi coğrafi dosya formatlarını, PostgreSQL/PostGIS, ODBC, ESRI Personal GeoDatabase, SQLite gibi coğrafi veritabanlarını ve OPeNDAP, GeoJSON gibi ağ protokollerini desteklemektedir. Ayrıca OGR vektör kütüphanesi ve GDAL raster kütüphanesini destekler. GDAL, 2014 yılından itibaren raster kütüphanesi tarafından desteklenen IMG, TIF, TFW, JPEG, JPEG 2000, JPGW, GIF, PNG, BMP dahil 133 raster veri formatı ve DXF, SVG, GML, GPX ve KML dahil 79 vektör veri formatı ile çalışabilmektedir. Open geospatial consortium görselleştirme ve veri erişim hizmetleri web haritası ve web özellik hizmetleri WMS ve WFS sağlamaktadır. QGIS, GPS entegrasyonu desteği, Open Street Map veri sunucuları ile etkileşim ve veri dönüştürme araçları dahil olmak üzere mevcut eklentilerle uygulamanın temel işlevselliğine yönelik uzantıların geliştirilebileceği ve kullanılabileceği bir "eklenti" mimarisi de içerir (Khan ve Mohiuddin, 2018).

QGIS3 resmi sitesinden ücretsiz olarak indirilerek Windows, macOS, Linux ve BSD işletim sistemlerine ek olarak mobil ve tablet için olan kurulum dosyaları indirilebilmektedir. QGIS3 paket yazılımı "OSGeo4W Shell" , "QGIS Desktop with

GRASS”, “QGIS Desktop” , “Qt Designer with QGIS customwidgets” , “SAGA GIS” beraberinde sunmaktadır.

2.4.3 Qt Designer

1990 yılında Eirik Chambe-Eng ve Haavard Nord isimli geliştiriciler tarafından ultrason görüntüleri için bir C++ veritabanı uygulaması tasarlanmak istenmiş ve sistemin Unix, Macintosh ve Windows üzerinde bir GUI ile çalışabilmesi; çapraz platform desteğine sahip olması amaçlanmıştır. Çapraz platform desteği bir uygulamanın birden fazla işletim sisteminde çalışabilmesi anlamına gelmektedir (Kucher, 2016). GUI ifadesi ise bilgisayar ve elektronik cihazlarda işlem yapılan komutlar ve bu komutların çıktıları yerine simgeler, ikonlar ve diğer görsel grafikler yardımıyla kullanması ile kullanıcılara yardımcı olması amacıyla geliştirilmiştir. Bilgisayar biliminde GUI terimi monitörün ekranda sunduğu görsel araçları ifade etmektedir. GUI, tasarlanmadan önce kullanıcılar bilgisayar üzerindeki tüm işlemlerini klavye üzerinden komutlar yazarak gerçekleştirmektedir. GUI, bilgisayarın ekranında sunulan pencereler, ikonlar, düğmeler vb. yardımıyla kullanıcıların komut satırına komutlar yazmasına gerek kalmadan bilgisayarları kullanabilmesini sağlar (Jansen, 1998).

Çalışmanın yapıldığı yıllarda C ve C++ dilleri ile resim görüntüleme işlemlerini yerine getirebilecek bir uygulama yazımı ve yazılan uygulamanın Unix, Macintosh ve Windows işletim sistemlerinde çalışmasının zorlukları sebebiyle Qt adı verdikleri bir kütüphane geliştirmeye başlamışlardır (URL-4).

Qt Designer, Qt kütüphanesinin sahip olduğu widget'ler sayesinde GUI tasarlamak için kullanılan Qt programıdır. Qt Designer uzun kodlar yazmaya gerek kalmadan kısa sürede tasarımlar yapabilmek için zaman içinde daha da geliştirilmiştir (Rischpater, 2013). Qt Designer açıldığında önceden hazırlanmış bir tasarımı import edebilen veya yeni tasarım yapılmasını sağlayan bir pencere açılmaktadır. Açılan pencerede tasarım için dialog, main window ve widget seçenekleri sunulmaktadır.

2.4.5 PyQt

PyQt, İngiliz şirket Riverbank Computing Limited tarafından geliştirilen ve çapraz platform uygulamaları geliştirmek için C++ ile yazılmış Qt kütüphanesinin Python

bağlamasıdır. PyQt, C++ ile yazılmış Qt kütüphanesini ve çapraz platform tarafından yorumlanan Python programlama dilini bir araya getiren GUI modülüdür. PyQt geliştiricileri PyQt4, PyQt5 ve son sürüm olarak Ocak 2021 de PyQt6 sürümlerini yayımlamışlardır. PyQt; QtCore, QtGui, QtNetwork, QtXml, QtSvg, QtOpenGL, QSql modüllerine sahiptir.

1. QtCore modülü, Qt tarafından desteklenen, eventloop ve Qt'nin signal-slot bağlantısı dahil olmak üzere, GUI olmayan çekirdek sınıfları içeren ve grafik arayüze ihtiyaç duymayan modüllerden biridir.
2. QtGui modülü, pencereleme sistemi entegrasyonu, OpenGL ve OpenGL ES entegrasyonu, 2D grafikler, temel görüntüleme, yazı tipleri ve metin için sınıflar sağlar. Bu sınıflar, Qt'nin kullanıcı arabirimi teknolojileri tarafından dahili olarak kullanılır ve örneğin düşük seviyeli OpenGL ES grafik API'lerini kullanarak uygulamaları yazmak için doğrudan kullanılabilir (URL-5).
3. QtNetwork modülü, TCP/IP kullanan programlama uygulamaları için bir dizi API sağlar. İstekler, tanımlama bilgileri ve HTTP üzerinden veri gönderme gibi işlemler çeşitli C++ sınıfları tarafından gerçekleştirilir (URL-5).
4. QtXml modülü, XML dosyalarıyla çalışmayı sağlar.
5. QtSvg modülü, SVG dosyalarını görüntüler.
6. QtOpenGL modülü, OpenGL kitaplığı aracılığıyla 2 ve 3 boyutlu grafikleri işlemeyi sağlar.
7. QSql modülü, veri tabanlarıyla çalışmak için araçlar sunar.

2.4.6 PyQGIS

PyQGIS, python dili ile QGIS API'sini kullanarak yeni bir CBS yazılımı geliştirmek veya QGIS içinde yeni bir eklenti yapmak için QGIS topluluğu tarafından geliştirmiş bir bağdaştırıcıdır (Kepoğlu ve Düzgün, 2008). Python programlama dili ile eklenti geliştirmek C++ ile yazılmış klasik eklentilere göre yazması, anlaması, bakımı, dağıtımı Python dilinin dinamik yapısı sebebiyle daha kolaydır. Python eklentileri QGIS eklenti yöneticisinde C++ eklentileri ile bir arada listelenmektedir (URL-6). Özgür yazılımlardan olan SIP ve SWIG yazılımları, C veya C++ kütüphanelerini Python ile birlikte kullanabilmek için yazılımcının işini kolaylaştırır. Genişletebilme özelliği için SWIG en sık kullanılan yazılımlardan birdir. SIP ise Riverbank Computing Limited firması tarafından Qt yazılım geliştirme ortamını Python'un

erişimi sağlaması amacıyla geliştirilmiş özgür bir yazılımdır (Kepoğlu ve Düzgün, 2008). Qt'nin PyQt'ye ve QGIS'in Pyqgis'e dönüşerek Qt ve QGIS kütüphanelerinin Python programlama diliyle kullanılmasını SIP aracı sağlamaktadır. SIP aracı ile PyQt veya PyQGIS sınıfları kullanıldığında C++ sınıflarındaki kodlar python kullanılarak çalışmaktadır.

2.4.7. GDAL ve OGR

GDAL ve OGR, Frank Warmerdam tarafından C++ programlama dili kullanılarak geliştirilmiştir. GDAL (Geospatial Data Abstraction Library), OSGeo (Open Source Geospatial Foundation) tarafından izin verilen X/MIT tarzı özgür yazılım lisansı altında yayınlanan, raster ve vektör coğrafi veri formatlarını okumak ve yazmak için kullanılan bir bilgisayar yazılım kitaplığıdır. Projeksiyonlar ve dönüşümler PROJ kitaplığı tarafından desteklenmektedir. Bir kitaplık olarak, desteklenen tüm formatlar için çağıran uygulamaya tek bir raster soyut veri modeli ve tek vektör soyut veri modeli sunar. Ayrıca, veri çevirme ve işleme için çeşitli kullanışlı komut satırı yardımcı programları ile birlikte gelmektedir. OGR, GDAL'den ayrılan OpenGIS Simple Features'den ilham alan ayrı bir vektör IO kütüphanesiyken GDAL 2.0 sürümü ile GDAL ve OGR bileşenleri birbirine entegre edilmiştir. Aralık 2021 tarihinde GDAL/OGR 3.4.1 sürümü yayınlanmıştır (URL-7). Pek çok ÖAKK CBS yazılımı, özellikle veri biçim değişimi ve koordinat dönüşümleri için OGR, GDAL ve Proj4 (Cartographic Projections Library) gibi ortak bazı kütüphaneleri kullanmaktadırlar (Beyhan ve Belge, 2008). QGIS, GDAL kütüphanesi desteği sayesinde coğrafi raster formatları (GeoTIFF, ErdasImagineImages, Arc/Info ASCII Grid, vb.), OGR kütüphanesi ile de vektör formatlarını (ESRI Shapefile, MapInfo, GML, vb.) desteklemektedir (Kepoğlu ve Düzgün, 2008).

2.5 Çok Ölçütlü Karar Verme Yöntemi

Çok Ölçütlü Karar Verme'nin genel amacı birçok seçim ölçüt ve farklı ölçüt önceliklerinin olduğu durumlarda gerçekleşmesi mümkün birçok seçenek arasından “en iyi” seçeneğin seçiminde karar vericiye yardım sağlamaktır (Erden ve Coşkun, 2011a). Karmaşık problemlerin çözümünde ölçüt ağırlıklarının belirlenebilmesi için ÇÖKV yöntemlerinden faydalanılmaktadır. ÇÖKV objektif ve her aşaması kolay kontrol edilebilir bir yöntemdir. Seçim ölçütlerinin ve karar vericilerinin birden fazla

olduğu durumlarda ortak bir çalışma ortamı ve iletişim kolaylığı sağladığı için oldukça kullanışlıdır. Ortaya çıkan alternatiflerin her birini ölçütlere uygunluk açısından değerlendirir, genel ve öznel görüşlerin ortak bir platformda değerlendirilmesine olanak sağlar (Ünaldık, 2019).

Güler ve Yomralıoğlu (2019) yer seçimi araştırmalarının birçoğunda CBS tabanlı ÇÖKV metotlarının kullanıldığını vurgulamış, Bilgilioğlu (2022) yaptığı çalışmada ise ÇÖKV modellerinin, literatürde birçok farklı bölgede kentsel ve bölgesel planlama, arazi uygunluk, afet duyarlılık/tehlike/risk bölgeleme ve yer seçimi gibi farklı birçok çalışmada uygulandığını ve başarılı sonuçların elde edildiğini belirtmiştir.

ÇÖKV yöntemlerinden bazıları, 90'lı yıllarda ticari ve açık kaynaklı olarak gevşek entegrasyon modu ile GIS yazılımları için geliştirilmeye başlanmıştır. İlk olarak Visual Basic, C, C ++, Python gibi programlama dilleriyle uygulama programlarının yazılmasına imkan sağlayan makrolar şeklinde geliştirilmiştir. Bu sayede platformdaki eklentilerin hazırlanma, derlenme ve çalıştırılmasını sağlayan araçlarda geliştirilmiş oldu (Zeydan vd., 2020).

Bir yer seçiminde karar verme süreci aşağıdaki işlem basamakları takip edilerek yapılmaktadır.

1. Yer seçimi alternatiflerinin değerlendirilmesinde kullanılacak ölçütlerin belirlenmesi;
2. Önemli ve etkin ölçütleri tanımlama ve ağırlıklarının belirlenmesi;
3. Yer seçimi alternatiflerinin geliştirilmesi;
4. Alternatiflerin değerlendirilmesinin ardından yer seçimi kararının verilmesidir (Erden ve Coşkun, 2011b).

Literatürde yer alan ÇÖKV yöntemlerinden bazıları; Ağırlıklı Çarpım Yöntemi, Analitik Hiyerarşi Yöntemi, Basit Ağırlıklı Toplam Yöntemi, Bulanık Ağırlıklı Toplam Yöntemi, Bulanık Mantık İşlemi, Değer/Fayda Fonksiyonu Yaklaşımı, İdeal Nokta Yöntemi, Sıralı Ağırlıklı Ortalama ve Uyum Yöntemidir (Bilgilioğlu, 2014).

2.5.1 Analitik Hiyerarşi Yöntemi (AHY)

1977 yılında Saaty tarafından geliştirilen Analitik Hiyerarşi Yöntemi (AHY), karmaşık problemler için çoklu ölçütlere göre ikili karşılaştırmalar yardımıyla sıralanmasında alternatifleri seçmeye yarayan, hem nitel hem de nicel yönlerin dikkate alındığı çok ölçütlü karar verme yöntemlerinden biridir. AHY ile karar verme adımları aşağıda belirtilen sırayla yapılmaktadır;

- i. Problem tanımlanır
- ii. Ölçütler ve alt ölçütler belirlenir
- iii. Alternatifler belirlenir
- iv. Problem için bir karar hiyerarşisi oluşturularak sorun bir karar ögeleri hiyerarşisine ayrılır. Satty bir hiyerarşinin her bir grubundaki ögelerin bağımsız olması gerektiğini varsaymıştır.
- v. İkili karşılaştırma matrisleri oluşturulur, öncelik vektörleri hesaplanır
- vi. Tutarlılık analizi yapılır
- vii. Alternatifler sıralanır (Zare vd., 2022).

AHY’de ölçütler ve alternatifler karşılaştırılırken karar vericilerin farklı iki kriterin birbirine göre ne kadar önemli olduğu durumları için ikili karşılaştırma değerleri oluşturmada, Çizelge 2.1’de gösterilen 1’den 9’a kadar puanlandırılmış Saaty ölçeği olarak da bilinen tercih ölçeğinden faydalanmaktadır. Bu ölçeğin etkinliği farklı alanlardaki uygulamalar ve başka ölçekler ile yapılan teorik karşılaştırmalar sonucunda saptanmıştır (Kuruüzüm ve Atsan, 2001).

Çizelge 2.1. İkili karşılaştırma matrisi için kullanılan ölçek (Saaty, 2000).

Mutlak Ölçekte Önem Derecesi	Tanım
1	Eşit Derecede Önemli
3	Diğerine Göre Orta Derecede Önemli
5	Temel veya Güçlü Derecede Önemli
7	Çok Güçlü Derecede Önemli
9	Aşırı Derecede Önemli
2,4,6,8	İki Bitişik Yargı Arasındaki Ara Değerler

Problemi tanımlamak için öncelikle durumu etkileyen ölçütler belirlenir. Her bir ölçüt ikili olarak karşılaştırılır ve sayısal değerler atanır. Karşılaştırma çiftlerinin sayısı, n tane kriteri gösterecek şekilde hesaplanır (Denklem 2.1).

$$n(n-1)/2 \quad (2.1)$$

Ölçütleri ağırlıklandırmak için çapraz kriterler (nxn) boyutlu ikili karşılaştırma matrisi (A) oluşturulur (Denklem 2.2).

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \quad (2.2)$$

a_{ij} değeri, i kriterinin j kriterinden daha önemli olma derecesini temsil eder. Karşılaştırma matrisinin köşegenindeki bileşenler (yani $i=j$ olduğunda), ilgili kriter kendisiyle karşılaştırılırken 1 değerini alır. Kriterlerin karşılaştırılması, önem derecelerine göre ve Tablo 2'deki değerlere göre bire bir ve ikili olarak yapılır. Köşegenin altındaki değerler için Denklem 2.3 kullanılır.

$$a_{ji} = \frac{1}{a_{ij}} \quad (2.3)$$

A matrisinin j sütunundaki her bir değer, j sütunundaki değerlerin toplamına bölünerek her sütundaki değerlerin toplamının 1 olacağı yeni bir A_w matrisi oluşturulur. Bu şekilde, normalleştirilmiş bir ikili karşılaştırma matrisi belirlenir (Denklem 2.4).

$$A_w = \begin{bmatrix} \frac{a_{11}}{\sum a_{i1}} & \frac{a_{12}}{\sum a_{i2}} & \dots & \frac{a_{1n}}{\sum a_{in}} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \frac{a_{n1}}{\sum a_{i1}} & \frac{a_{n2}}{\sum a_{i2}} & \dots & \frac{a_{nn}}{\sum a_{in}} \end{bmatrix} \quad (2.4)$$

c_i değerleri, A_w matrisinin i sütunundaki değerlerin ortalaması olarak hesaplanır ve C sütun vektörü (öncelik vektörü) elde edilir (Denklem 2.5).

$$C = \begin{bmatrix} c_1 \\ c_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ c_n \end{bmatrix} = \begin{bmatrix} \frac{a_{11}}{\sum a_{i1}} + \frac{a_{12}}{\sum a_{i2}} + \dots + \frac{a_{1n}}{\sum a_{in}} \\ \frac{a_{21}}{\sum a_{i1}} + \frac{a_{22}}{\sum a_{i2}} + \dots + \frac{a_{2n}}{\sum a_{in}} \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ \frac{a_{n1}}{\sum a_{i1}} + \frac{a_{n2}}{\sum a_{i2}} + \dots + \frac{a_{nn}}{\sum a_{in}} \end{bmatrix} \quad (2.5)$$

AHY kendi içinde tutarlı bir sisteme sahiptir, ancak sonuçların doğruluğu, karar vericilerin ölçütler arasında yaptıkları birebir karşılaştırmadaki tutarlılığa bağlı olacaktır. Bu bağlamda AHY, bu karşılaştırmalardaki tutarlılığı ölçmek için bir süreç sunmaktadır. Tutarlılığı belirlemek için önce tutarlılık vektörü olarak bilinen $A \times C$ matrisi hesaplanır (Denklem 2.6).

$$A \times C = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \times \begin{bmatrix} c_1 \\ c_2 \\ \cdot \\ \cdot \\ \cdot \\ c_n \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} \quad (2.6)$$

İkinci olarak, $\lambda_{\max}$ değeri hesaplanır (Denklem 2.7).

$$\lambda_{\max} = \frac{1}{n} \sum_{i=1}^n \frac{x_i}{c_i} \quad (2.7)$$

Burada $\lambda_{\max}$, karşılaştırma çiftleri matrisinin özdeğeridir. Daha sonra Tutarlılık İndeksi (Tİ) hesaplanır (Denklem 2.8).

$$TI = \frac{\lambda_{\max} - n}{n - 1} \quad (2.8)$$

Hesaplanan Tİ, test edilecek öncelik vektörünün tutarlılığını sağlar. AHY'de Tİ hesaplaması, kriter sayısı ile $\lambda_{\max}$ katsayısının karşılaştırılmasına dayanır.

Son olarak, n değerlerinin yeterli düzeyde tutarlılığa sahip olup olmadığını belirlemek için Rastgele İndeks (Rİ) ve Tİ derecelendirilir ve Tutarlılık Oranı (TO) belirlenir (Denklem 2.9). Rİ, Saaty tarafından belirlenen standart bir değere sahiptir. Çizelge 2.2, ölçüt sayısına göre değişen bu değerleri göstermektedir.

$$TO = TI / RI \quad (2.9.)$$

$TO \leq 0.10$ ise yapılan karşılaştırmalar tutarlı olarak kabul edilir. $TO > 0.10$ ise sonuç tutarsızdır. Bu durumda AHY anlamlı sonuçlar vermeyebilir ve ikili karşılaştırma matrislerinin yeniden değerlendirilmesi gerekir.

Çizelge 2.2. Rastgele indeks (Saaty, 2000).

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Rİ	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49	1.51	1.48	1.56	1.57	1.59

ÇÖKV uygulamalarında farklı ölçeğe sahip ölçütler bir araya getirilerek standartlaştırılmalı veya ortak bir uygunluk değerine dönüştürülmelidir. Analitik bir yöntem olarak ağırlıklı lineer kombinasyon, birçok araştırmacı tarafından birden fazla ölçütlerin dikkate alınması gereken yer seçimi gibi problemlerin çözümünde kullanılmaktadır. ÇÖKV uygulamalarında dikkate alınacak her bir özelliğe ölçüt adı verilir ve her bir ölçüte önemini gösteren bir ağırlık değeri atanır. Her bir ölçüt, haritalar veya katmanlar tarafından mekansal olarak temsil edilir ve CBS ortamındaki tüm ölçütlerin birleşimi için grid veri dosya formatına dönüştürülür (Malczewski, 1999).

2.6 ÇÖKV Analizleri için Eklenti Geliştirilmesi

Veri çeşitliliği ve hacminin fazla olduğu problemlerin çözümünde etkin bir araç olarak kullanılan CBS, bu özelliği sayesinde neredeyse her meslek dalını ilgilendirmeyi ve günlük hayatın bir parçası olmayı başarmıştır. Gelişen bilgi işlem teknolojileri ile birlikte, farklı alanlara ve sahip olduğundan daha fazla kullanıcıya da hitap edecek şekilde geliştirilmesi, kullanıcılar için daha önce zor ve karmaşık yöntemlerle analizinin mümkün olmadığı veya kısmen mümkün olduğu olay ya da olayları tespit etmesi ve buna yönelik çözümler sunması gerekmektedir. (Kaplukan, 2014). Bu gereklilikten dolayı ticari ve açık kodlu CBS yazılımı sayısı ve bu yazılımların içerdikleri fonksiyonlar giderek artmaktadır. Ancak birçok CBS yazılımına rağmen kullanıcıların projelerini yapmak için kullanmak istediği fonksiyonlar yazılımın yalnızca küçük bir parçasını oluşturabilir veya ihtiyaç duydukları bazı fonksiyonlar o yazılımda hiç bulunmayabilir. Bu nedenle kullanıcılar bir işlemi gerçekleştirmek için birden fazla yazılıma ihtiyaç duyabilmekte ya da çok basit bir işlem için yazılım edinmek zorunda kalabilmektedir. Tüm bu sebepler ise analizlerin yapılabilmesinde kısıtlamalara ve zorluklara neden olmaktadır (Yılmaz, 2012). Bu gereklilik neticesinde CBS yazılımları, üzerinde eklenti geliştirebilecek araçlara sahip olarak üretilmekte bu ise daha çok sayıda problemin çözümüne esneklik kazandırmaktadır. Kullanıcılar kendi geliştirdikleri veya başka kullanıcılar tarafından geliştirilen eklentilerin varlığı sayesinde ihtiyaç duyulan alanlardaki problemlere çözümler aramakta bu sayede CBS nin gelişimine katkı sunmaktadırlar. Bu kapsamda problemlerine ÇÖKV yöntemleriyle çözüm arayan kullanıcıların, ihtiyaçları doğrultusunda AKK'lı yazılımları doğrudan ve CBS eklentisi geliştirilerek kullandığı literatürde yer alan çalışmaların bazıları aşağıda incelenmiştir.

Güler ve Yomralıoğlu (2019) yapmış oldukları çalışmada günümüz teknolojisi haline gelen elektrikli araçlar için şarj istasyonu yer seçimin de açık kaynak kodlu yazılım olan QGIS ile çözüm üretmeye çalışmışlardır. Açık kaynak kodlu CBS yazılımı kullanarak elektrikli şarj istasyonu yer seçimi için Bulanık Analitik Hiyerarşi Yöntemi (BAHY) kullanılarak mekansal verilerin analiz edilmesi ile çıktılar üretilmesine yönelik çözüm önerisi sunmuşlardır. Çalışmanın farklı araştırmacılar tarafından değişik çalışma bölgelerinde kullanımının sağlanması ve çalışmada elde edilen sonuçların güvenilirliğinin artırılması amacıyla AKK'lı CBS yazılımı olan QGIS

yazılımından faydalanmışlardır. QGIS yazılımında yapılan çalışmaya yönelik mekansal analizleri içeren model oluşturmuş ve kriter ağırlıklarının kullanımıyla elektrikli araç şarj istasyonu uygunluk indeksini elde edilmesiyle çalışmalarını tamamlamışlardır. Çalışma açık kaynak kodlu QGIS yazılımının ÇÖKV yöntemlerinden BAHY'nin uygulanmasında doğrudan kullanımına örnek olarak incelenmiştir.

Congedo (2013) yaptığı çalışmada QGIS 1.8 sürümü üzerinde Semi-Automatic Classification Plugin isimli bir eklenti geliştirmiştir. Geliştirilen eklenti ile Dar as Salaam şehrinde bulunan kıyılardaki iklim değişikliğine uyum sağlamak, kıyı bölgelerinde yaşayan sakinlerin geçim kaynakları üzerinde iklim değişikliğinin araştırılması, belediyelerin iklim değişikliğini anlama kapasitelerini arttırmak, kentsel faaliyetler için strateji ve planlara entegre etmek için metodolojiler geliştirmek amacıyla uzaktan algılama görüntülerini sınıflandırılmada kullanılacak ROI (Region and Interest) ve Classification araçlarından oluşan bir eklenti tasarlamıştır. ROI aracı sekmesi, yarı otomatik sınıflandırmalarda kullanılan eğitim alanı toplamaya yönelik iken, Classification aracı sekmesi ise SAGA tarafından sağlanan denetimli sınıflandırma algoritmaları (Sextante uygulamasını) kullanarak uzaktan algılama görüntüsünün yarı otomatik olarak sınıflandırılmasını sağlamaktadır.

Jung (2013) yaptığı çalışmada QGIS için LecoS (Landscape ecology Statistics) Arazi Ekolojisi İstatistikleri adlı bir eklenti geliştirilmiştir. Python yazılımına ait SciPy ve Numpy kitaplıklarından yararlanılarak hazırlanan eklenti temel ve gelişmiş arazi metriklerini hesaplamak ve yürütmek için çeşitli işlevler sağlayarak araziye dair analizler gerçekleştirmektedir. Bu amaçla bir bölgeye ait vektör formatındaki arazi verilerini analiz eden eklenti sonuçta o bölgede yer alan bitki örtüsünü kategorilere ayırarak, yüzdelerini kullanıcılar ile paylaşmaktadır.

Saralıoğlu vd. (2014) yapmış oldukları çalışmada python programlama dili ile bazı görüntü işleme algoritmalarını KKK'lı ticari bir yazılım olan ArcGIS yazılımı içinde geliştirmişlerdir. Böylece literatürde yaygın olarak kullanılan görüntü kaynaştırma, sınıflandırma algoritmaları ve kalite ölçme metriklerini, bir python görsel ara yüzü ile ArcGIS eklentisi olarak oluşturarak CBS ve Uzaktan Algılama konularında çalışan araştırmacıların kullanımına sunmuşlardır.

Yalçın ve Sabah (2017) yapmış oldukları çalışmada, QGIS yazılımı ve AHY kullanılarak Edirne ilinde yer alan sanayi kuruluşlarının deprem tehlike analizini gerçekleştirmiştir. Çalışma sonucunda deprem açısından tehlikeli ilçeleri tespit ederek tehlike derecelerine göre sıralamışlardır. Edirne ili için sanayi/deprem riski ilişkisini belirlemeye çalışmışlardır. Çalışmalarında deprem tehlike analizinde kullanılması düşünülen mekansal verilerin birbirleriyle oranının hesaplandığı “kriterler sekmesi”, bu parametrelerin ilçeler için değer girişlerinin yapıldığı ve bu değerlerin her ilçe için oranlarının hesaplanabildiği “alternatifler sekmesi” ve bu sekmelerden elde edilen değerlerin analiz sonuçlarının gösterildiği “sonuç sekmesi” bir yazılım oluşturmuşlardır.

Duarte vd. (2018) yaptıkları çalışmada İnsansız Hava Aracı (İHA) görüntülerinden elde edilen görüntüler aracılığıyla görüntü bölümleme ve görüntü sınıflandırma ve fraktal analiz teknikleri yardımıyla ağaç tacı üzerindeki çeşitli parametreleri ve metrikleri tahmin eden bir QGIS eklentisi geliştirmiştir. Metrikler; alan, çevre, ağaç sayısı, ağaçlar arasındaki mesafe ve kayıp bir ağaç olarak belirlenmiştir. Bu metodoloji zeytin, okaliptüs ve bağ olmak üzere üç farklı plantasyonda test edilmiştir. Uygulama geliştirilirken QGIS 2.18 sürümü, uzaktan algılama işlemleri için AKK’lu Orfeo Toolbox 5.0.0 yazılımı ve GRASS GIS 7.2.2 sürümlerinden sağlanan çeşitli araçlara ek olarak Python 2.7 sürümü ve PyQt4 API'sinden yararlanılmıştır. Eklenti verileri işlemek ve dışa aktarmak için 4 adet lineedit ayrıca 2 adet basmalı butondan oluşan bir arayüz tasarımına sahiptir.

Mileu ve Queirós (2018) yaptıkları çalışmada AKK ile asimetric haritalama tekniği kullanarak nüfus dağılımını daha iyi modelleyebilmek ve nüfus dağılım haritalarını iyileştirmek için mevcut olan nüfus sayımı verileri kullanılarak nüfus yoğunluğunu arazi kullanımına göre haritalandırmasını sağlayan bir QGIS eklentisi geliştirmiştir. Çalışmada Portekiz Guimarães belediyesi 2011 nüfus sayımı verilerini 25 m'lik bir alana yeniden dağıtmak için Corine arazi örtüsü veri kümesi kullanılmıştır. Uygulama geliştirilirken GRASS GIS'den sağlanan çeşitli araçlara ek olarak python 2.7 sürümü ve PyQt4 API'sinden yararlanılmıştır. Geliştirilen eklenti ile nüfus sayımlarından elde edilen nüfus sayımı verilerinin, su kütleleri veya boş araziler gibi nüfussuz alanların nüfus sayımı alanlarından çıkarılmasıyla dağılımının otomatik olarak daha doğru bir şekilde yapılması amaçlanmıştır.

Kalkan (2019) yapmış olduđu çalışmada; Leaflet kütüphanesinden yararlanarak geliştirdiği web uygulamasında; daha önce CBS tecrübesi olmayan gayrimenkul yöneticilerinin verileri hızlı ve kolay bir şekilde görüntüleme ve konumsal analizler yapabilmelerini amaçlamıştır. Geliştirdiği araçlar sayesinde çeşitli sorgulamaların yapılması ve dış unsurlara ulaşılması sayesinde faktörlerin belirlenmesi ile portföylerin verimli yönetilmesini sağlamıştır. Bu sayede daha önce CBS tecrübesi olmayan kullanıcılara yönelik bir tasarım yapıldığı görülmektedir.

Vitalis vd. (2020) yapmış oldukları çalışmada QGIS 3.0 programına 2018 yılından itibaren üç boyutlu görselleştirme desteğinin eklenmesi ve CityJSON, CityGML 2.0 veri modelinin üç boyutlu şehir modelleri için kullanılmaya başlaması sayesinde QGIS'de üç boyutlu şehir modelleri için CityJSON veri setlerinin yüklenmesini sağlayan bir eklenti geliştirmişlerdir. Eklenti, bir CityJSON dosyasını ayrıştırmakta ve tüm şehir nesnelerini tanımlamak için yapısını analiz etmektedir. Daha sonra analiz edilen her şehir nesnesinin geometrisi ve nitelikleri QGIS özelliklerine dönüştürülerek kullanıcı tercihlerine göre katmanlara ayrılmaktadır.

Çalışkan ve Anbaroğlu (2020) yapmış oldukları çalışmada QGIS 3.0 programında Kruskal'ın En Kısa Yol Ağacı algoritmasından yararlanarak Geo-MST eklentisi geliştirmişlerdir. Yol ağacı, bir graf üzerinde tüm düğümleri kapsayan ağaç şeklinde bir yolu ifade etmektedir. Ağaç şeklinde olduğu için döngü içermemekte ve aynı noktaya gelmemektedir. En az maliyetli olan en kısa yol ağacı olarak adlandırılır ve bir graf üzerinde birden çok yol ağacı olabilir. Algoritma, nokta çiftleri arasından komşuluklara en yakın düğümü kendisine katarak graf üzerindeki en kısa mesafeyi bulmaktadır. Versiyon 1.0 olarak tasarlanan eklenti daha sonra versiyon 2.0 olarak güncellenmiştir. Eklenti Esri ShapeFile, GeoJSON ve GML vektör formatları ile TIFF raster veri formatlarını desteklemektedir. Eklenti geliştirilirken kodlar python ve PyQGIS yardımıyla yazılmış, arayüz tasarımı ise PyQt kütüphanesi yardımıyla tasarlanmıştır.

Zeydan vd. (2020) yapmış oldukları çalışmada NetCAD firması işbirliğiyle ÇÖKV yöntemlerinin doğrudan kullanıldığı bir MKDS (Mekansal Karar Destek Sistemi) geliştirmişlerdir. Bu amaçla, yer seçim ve değerlendirme problemleri için, belirlenen kriterlere göre değişebilen parametreler ile tematik haritalar yapabilen bir model geliştirmişlerdir. Model geliştirilirken bulanık ÇÖKV metotları ve CMA-ES

algoritmalarından yararlanmışlardır. Geliştirilen program Netcad Mimar'da Analiz modülü altında kullanıcılara sunulmuştur.

Titti vd. (2022) yapmış oldukları çalışmada, makine öğrenmesi algoritmalarından yararlanarak heyelan duyarlılık haritalaması için python yazılımı ile dört adet menüden oluşan (Classify SI, Data Preparation, SI ve SI k-fold) bir Qgis eklentisi geliştirmiştir.

Yukarıda bahsi geçen çalışmalara ek olarak literatürdeki AKK'lı yazılımlar yardımıyla Mekansal Karar Analizine yönelik bazı çalışmalar incelendiğinde, QGIS vb. programların kullanıcısı tarafından lisanslı programlar yerine maliyetsiz ve güvenilir olduğu gibi sebeplerden tercih edildiği, yazılım geliştiriciliği noktasında ise genel olarak sonuç çıktı ve görsellerinin oluşturulması için kod parçacıkları halinde birtakım uygulamaların yapıldığı gözlenmektedir. Ayrıca ticari bir yazılım olan Netcad yazılımına C sharp yazılımı kullanarak ÇÖKV yönteminde kullanılmak üzere bir program geliştirilmiştir. Buna ek olarak kapalı kaynak kodlu ticari bir yazılım olan ArcGIS yazılımına açık kaynak kodlu python yardımıyla bir takım eklentiler yazıldığı da görülmektedir. Son yıllarda makine öğrenmesi algoritmalarından yararlanan bazı uygulamalarda mevcuttur.

3. YAPILAN ÇALIŞMALAR

Günümüzde teknolojinin gelişimi ve modern dünyanın değişen şartları ihtiyaç tanımını da değiştirmiştir. CBS tabanlı sistemlerin mobil cihazlarda da kullanımı her geçen yıl farklı bir boyut kazanırken, yapay zekadaki olağanüstü gelişim ile birlikte otonom sistemlerin kullanılmaya başlanması yapay zeka ile CBS'nin birleştirilmesi sonucunu doğurmuştur. Tüm bu gelişimler ışığında iki temel ihtiyaçtan; ilk olarak CBS ile problemlere yönelik çözüm önerileri ve analizler sunmak, ikinci olarak da hali hazırdaki CBS yazılımlarının geliştirilmesi özellikle de maliyetsiz olması ve esnek yapısı sayesinde açık kaynak kodlu özgür CBS yazılımlarının geliştirilmesi ihtiyacı tespit edilmiştir. Araç ve eklentilerinin karmaşıklığı kullanıcıyı zorladığı için bu alanda, Richard Stallman'nin tanımladığı özgür yazılım ölçütlerine uygun bir yazılım kullanıcılara sade ve pratik bir arayüz tasarımı ile sunarken, aynı zamanda kullanıcıların sadece yapılması istenilen analizlerin kolay ve hızlı yapabileceği bir tasarım ortaya konulması amaçlanmıştır. Bu kapsamda tasarlanması planlanan bir CBS yazılımı ile kullanılan CBS yazılımlarının araç ve eklentilerinin karmaşıklığı kullanıcı için en aza indirilirken, geliştirilecek yazılım ile ÇÖKV yöntemlerinin açık kaynak kodlu özgür CBS uygulamasında doğrudan kullanımı, basite indirgenmiş bir tasarım mimarisi ile ücretsiz olarak kullanıcılara sunulması bu tez çalışması ile birlikte çalışılmıştır.

CBS projeleri geliştirilirken kullanılan CBS yazılımlarının çeşitliliği, araç ve eklentilerinin karmaşıklığı, program yapılarının tüm analizleri yapabilecek araçları barındırması ve beraberinde kullanılmak istenilen araçların hangisinin olduğunun belirlenmesindeki güçlükler özgür yazılım geliştiricileri tarafından tespit edilen problemlerin başında gelmektedir. Her ne kadar CBS yazılım geliştiricileri arayüz mimarisi açısından öğrenimi ve kullanımı kolay bir tasarım yapmak isteseler de gelişen teknoloji çok farklı veri tipleri ve analiz araçları ortaya çıkarmış, bu ise aynı yazılımın her gelişen sürümünde biraz daha karmaşık bir görünüme sahip olmasını sağlamıştır.

Maliyetsiz oluşunun yanı sıra sahip olduğu topluluklar sayesinde çok hızlı gelişen, internetin olduğu her yerden rahatlıkla indirilebilir Linux, Mac ve Windows gibi işletim sistemleri kullanan tüm kullanıcılar için sorunsuz erişim ve kullanım kolaylığı

sağladığı için açık kaynak kodlu yazılımların kullanımı artmıştır. CBS projelerinde problemlere yönelik çözüm arayan kullanıcıların, ÇÖKV yöntemlerinden herhangi birini kullanarak analizler yapabilmesi bir CBS programı kullanma gerekliliğini beraberinde getirmektedir. Ücretsiz ve yalnızca ÇÖKV işlemlerini yerine getirmek için tasarlandığı için karmaşık menü tasarımına sahip olmayan ayrıca açık kaynak kodları sayesinde ilerleyen zamanda isteyen kullanıcılar tarafından daha da geliştirilebilir bir yazılımın kullanıcılara sunulması düşünülmüştür.

3.1. ÇÖKV Analizleri için Açık Kaynak Kodlu Eklenti Geliştirilmesi

CBS eklentisi geliştirme işlemine ilk olarak ihtiyacın tespiti ile başlanmıştır. Bu çalışma kapsamında geliştirilmesi planlanan eklenti ile ÇÖKV yöntemleri yardımıyla açık kaynak kodlu özgür bir CBS uygulamasında doğrudan kullanımı amaçlanmaktadır. Yapılan literatür incelemesinde ÇÖKV yöntemlerinin kullanıldığı eklentilerde bu yöntemlerin etkili kullanılmayarak yalnızca kısıtlı işlemleri yerine getirebilen eklentilerin varlığı tespit edilmiş, buna karşın ÇÖKV yöntemlerinin kapsamlı olarak kullanıldığı eklentilerin sayısının ise çok az olduğu tespit edilmiştir. Bu alanda çalışmalar yapan kullanıcıların ÇÖKV yöntemlerinin doğrudan kullanıldığı bir CBS eklentisi ile analizlerini kolaylıkla yerine getirmesi beklenmektedir. QGIS programında geliştirilen eklenti bu sayede dünya üzerindeki tüm kullanıcıların kullanımına sunulması hedeflenmiştir. ÇÖKV analizleri için açık kaynak kodlu eklenti geliştirme adımları aşağıda detaylı olarak açıklanmıştır.

3.1.1. Uygulama yöntemlerinin tespiti

Çalışma kapsamında açık kaynak kodlu özgür CBS yazılımı geliştirebilmek için yerli ve yabancı literatür de daha önce yapılmış çalışmaların incelenmesi yapıp ön bilgi edinilmiştir. Proje geliştirilirken çalıştırılabilir (Executable) bir program dosyası haline getirmek ya da halihazırda kullanılan bir CBS programına eklenti yazmak olarak iki yöntem belirlenmiştir.

İlk olarak projeyi çalıştırılabilir bir program dosyası haline getirmek düşünülmüştür. Ancak gerek Linux, Mac, Unix ve Windows gibi işletim sistemi kullanıcılarının farklı ortamlarda executable bir dosyayı çalıştırmaktan kaynaklanacak güçlükler, gerekse de Gdal başta olmak üzere bazı kütüphanelerin dışardan import edilmesinin

kullanıcıya getirebileceği güçlükler tespit edilmiştir. Executable bir programda bazı kütüphaneler dışarıdan import edilse bile Gdal gibi kütüphanelerin kullanılan yazılıma uygun sürümünün bilgisayara indirilerek yüklenmesi gerekmekte, bunun ise kullanıcı için ayrıca birtakım soruları ve sorunları da beraberinde getireceği düşünülerek vazgeçilmiştir.

İkinci olarak bir CBS programına eklenti geliştirilmesi düşünülmüştür. Eklenti geliştirmenin GDAL kütüphanesi başta olmak üzere, üzerinde eklenti geliştirilecek CBS programında zaten yüklü olması, kullanıcı için kurulum ve kullanım kolaylığı sağlaması avantajları dışında, tasarımcı içinde yayımlanması ve paylaşılmasının kolaylığı düşünülerek bu yönetime karar verilmiştir.

3.1.2. Kullanılacak yazılımın tespiti

Açık kaynak kodlu CBS eklentisi geliştirilmeden önce QGIS3, gvSIG, GRASS GIS, ILWIS, SAGA GIS, MapWindow, GeoDa, Whitebox GAT, uDig, OpenJump, FalconView, OrbisGIS, Diva GIS gibi dünya üzerindeki popüler açık kaynak kodlu CBS programları incelenmiştir.

İncelenen SAGA GIS, ILWIS, GRASS GIS ve TerraView yazılımlarının daha çok tecrübeli CBS kullanıcıları için özellikle ILWIS ve SAGA yazılımlarının araştırma yapan bilim insanları için daha kullanışlı olduğu tespit edilmiştir. MapWindow, QGIS, Thuban ve Java tabanlı CBS yazılımların tüm kullanıcı düzeyleri için kullanılabilir olduğu ancak daha önce incelenen GRASS, SAGA ve ILWIS yazılımlarının daha üst düzey bir takım CBS çözümlemelerini yapabilmesi sebebiyle araştırmacılar için daha fazla tercih edileceği tespit edilmiştir. Daha çok CBS için görüntüleme sistemi olarak geliştirilmiş TerraView, Thuban ve OpenEV yazılımlarından kullanımı kolay olan CBS yazılımının Thuban olduğu, TerraView yazılımının ise sahip olduğu özellikler ve kullanıcı kitlesi bakımından üst düzey kullanıcılara hitap ettiği tespitleri yapılmıştır. QGIS ve MapWindow yazılımlarının da eklentileri yardımıyla çeşitli karmaşık bilimsel ve konumsal analizleri yerine getirebildiği gibi tespitler yapılmış ve değerlendirme aşamasına geçilmiştir.

Yapılan incelemeler değerlendirildiğinde QGIS3 programının açık kaynak kodlu python programlama dili ile eklenti geliştirmeyi mümkün kılması, GRASS GIS

algoritmalarının QGIS araç kutusuna dahil edilmesinin sağladığı avantajlar, Qt Designer ile doğrudan etkileşimli çalışabilmesi, internet üzerinden kolay erişimi, Windows, macOS ve Linux işletim sistemleri için kolay kurulum imkanları sağlaması, sürekli güncellenmesi ve eklenti geliştirmeyi kolay hale getiren bir tasarıma sahip olması sebebiyle QGIS3 programı ile eklentinin geliştirilmesine karar verilmiştir. Eklenti geliştirilirken QGIS3.12.0 sürümü kullanılmıştır. QGIS3.12 paket yazılımı ile birlikte “OSGeo4W Shell”, “QGIS Desktop 3.12.0 with GRASS 7.8.2”, “QGIS Desktop 3.12.0”, “Qt Designer with QGIS 3.12.0 customwidgets”, “SAGA GIS (2.3.2)” kurulumları da tamamlanmıştır. Python 2.7 sürümünün sözdizimi farklılıkları ve QGIS3 ile Python3 sürümünün kullanılmaya başlanması sebebiyle eklenti geliştirilirken Python3 sürümünden yararlanılmıştır.

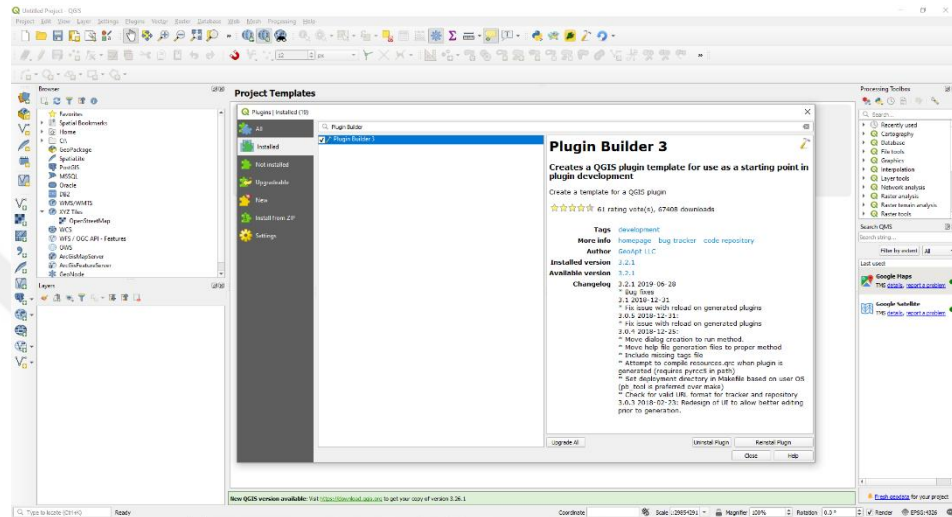
3.2. Yeni Eklenti Üretme

Tasarlanması planlanan bir eklentide `_init_.py`, `mainPlugin.py`, `resources.qrc`, `resources.py`, `form.ui`, `form.py` ve `metadata.txt` dosyalarının üretilmesi gerekmektedir. Üretilen dosyalar QGIS yazılımının kurulumuyla birlikte gelen (QGIS3\profiles\default\python\pluginsklasör yolunda bulunan) `plugins` klasörünün içerisinde yer almalıdır. Bu dosyalar aşağıda tanımlanan işlemleri yerine getirmektedir.

1. `_init_.py` = eklentinin başlangıç noktasıdır. `classFactory ()` metodu ve diğer gerekli kodların başlatma kodlarını içermelidir. `_init_.py` dosyası python bileşen çağırma sistemi için gereklidir. Ayrıca, eklentinin QGIS içerisinde çalışabilmesi için `classFactory()` fonksiyonunun olması gerekir. Fonksiyon `QgisInterface`` sınıfını çağırmalı ve `mainplugin.py` dosyasındaki eklentinin ana sınıfını geri döndürmelidir.
2. `mainPlugin.py` =Eklentinin ana kodlarını içeren dosyadır.
3. `resources.qrc` = QT Designer tarafından oluşturulan bir .xml dosyasıdır. Görsel arayüzlerin sahip olduğu kaynakların dosya yollarını içerir.
4. `resources.py` = .qrc dosyasının Python tarafından tanımlanmış çevirisidir.
5. `form.ui` = Qt Designer yazılımının oluşturduğu kullanıcı arayüzüdür.
6. `form.py` = Python tarafından tanımlanmış `form.ui` dosyasıdır.
7. `metadata.txt` = Genel bilgileri, versiyon numarası, adı ve eklentinin web sitesi, eklenti yapısı gibi bilgileri içerir (URL-6).

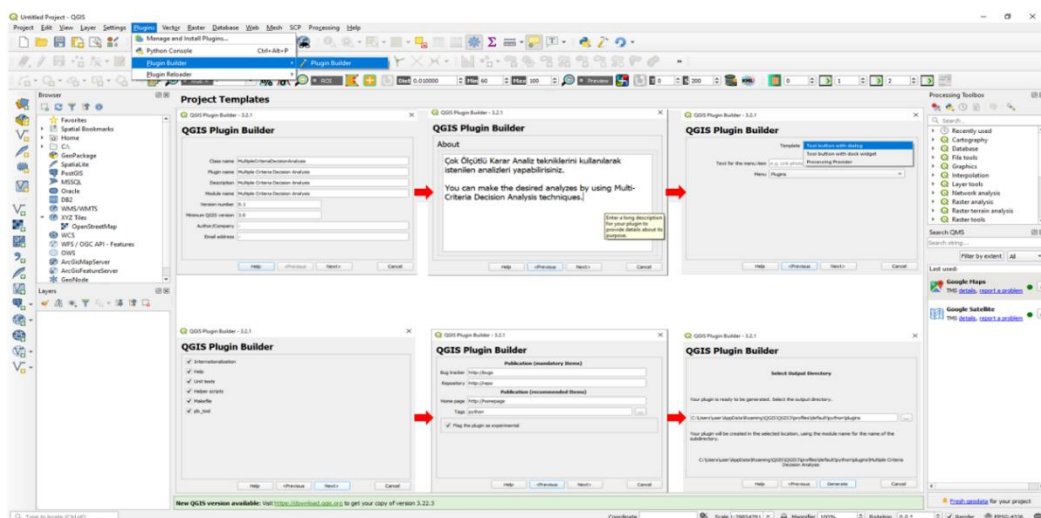
3.2.1. Plugin Builder

Tasarımı planlanan eklentide gerekli tüm dosyaların oluşturmak için daha önce yine özgür yazılımcılar tarafından geliştirilen “Plugin Builder 3” eklentisinden yararlanılmıştır. Plugin Builder 3 eklentisi kurulumu QGIS 3 yazılımının plugins menüsünde yer alan “Manage and Install Plugins” sekmesinden gerçekleştirilmiştir (Şekil 3.1).



Şekil 3.1. Plugin Builder 3 eklentisinin yüklenmesi.

Plugins menüsünün altında yer alan Plugin Builder eklentisi çalıştırılarak eklenti üretme işlemine başlanabilir. “MultipleCriteriaDecisionAnalysis” isimli yeni eklenti üretme adımları sırasıyla Şekil 3.2’de gösterilmiştir.



Sekil 3.2. Yeni eklenti üretimi.

Plugin Builder ile eklenti üretimi tamamlandıktan sonra (QGIS3\profiles\default\python\plugins klasör yolunda bulunan) plugins klasörü içerisinde MultipleCriteriaDecisionAnalysis isimli yeni bir klasör oluşur. Oluşan klasör içerisinde help, i18n, scripts, test isimli klasörler _init_.py, MultipleCriteriaDecisionAnalysis.py, MultipleCriteriaDecisionAnalysis_dialog.py, plugin_upload.py isimli python dosyaları icon isimli png dosyası, MultipleCriteriaDecisionAnalysis_dialog_base.ui isimli ui dosyası, pb_tool.cfg dosyası, resources.qrc dosyası, Makefile, pylintrc isimli dosyalar ve metadata.txt, README.txt isimli metin dosyaları yer almaktadır.

3.2.2. Yazılımın derlenmesi

Plugin Builder3 eklentisi, ürettiği MultipleCriteriaDecisionAnalysis eklentisi için plugins klasörü içerisine yeni bir multiplecriteriadecision analysis klasörü oluşturmuştur. Ancak eklenti oluşturulurken “There source compiler pygcc5 was not found in yourpath. You'll have to manually compile there sources.qrc file with pygcc5 before installing your plugin” uyarı mesajı ile karşılaşılmaktadır. Problemin kaynağı yeni üretilen plugin dosyasının içerisinde resources.qrc dosyasının resources.py formatında bir .py dosyasına ihtiyaç duymasıdır. Bunun için komut satırında pluginin oluşturulduğu dizine gidilerek;

>> pygcc5 resources.qrc -o resources.py kodu çalıştırılmalıdır. Kod çalıştıktan sonra resources.py isimli yeni bir dosya oluşur. Oluşan dosya sayesinde plugin sorunsuz çalışabilecektir. Ayrıca ön tanımlı olarak Plugin Builder3 eklentisinin ürettiği icon.png resim dosyası değiştirilmek istendiğinde yeni bir png uzantılı resim dosyası icon ismiyle kaydedilerek;

>> pygcc5 resources.qrc -o resources.py kodu tekrar çalıştırılmalıdır (Şekil 3.3).



```
Command Prompt
Microsoft Windows [Version 10.0.19043.1526]
(c) Microsoft Corporation. All rights reserved.

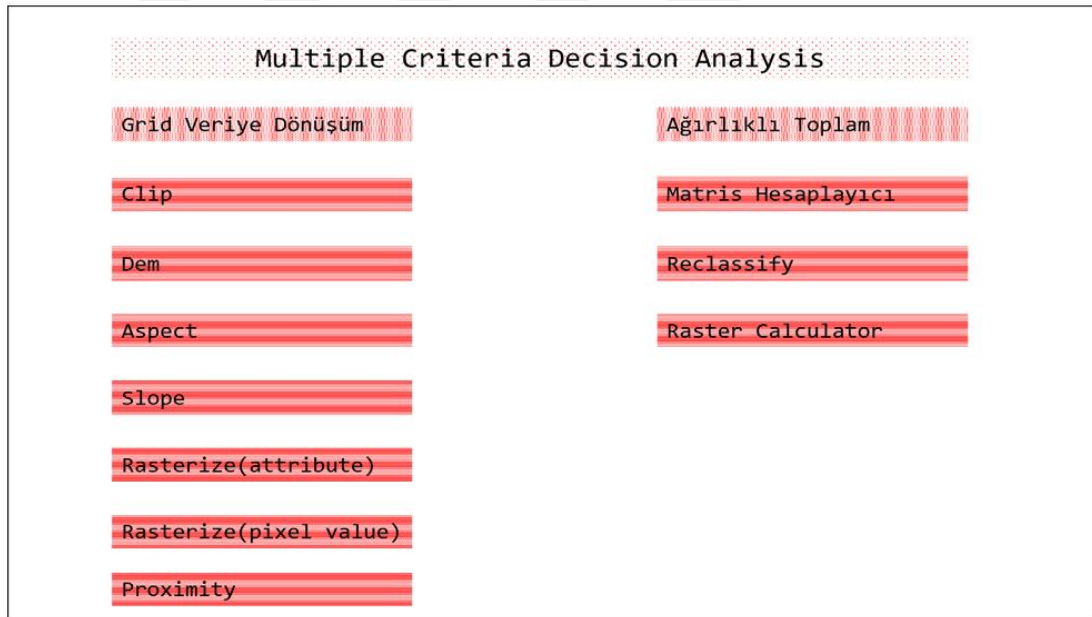
C:\Users\user>cd C:\Users\user\AppData\Roaming\QGIS\QGIS3\profiles\default\python\plugins\multiple_criteria_decision_analysis
C:\Users\user\AppData\Roaming\QGIS\QGIS3\profiles\default\python\plugins\multiple_criteria_decision_analysis>pygcc5 resources.qrc -o resources.py
C:\Users\user\AppData\Roaming\QGIS\QGIS3\profiles\default\python\plugins\multiple_criteria_decision_analysis>
```

Şekil 3.3. Yazılımın derlenmesi.

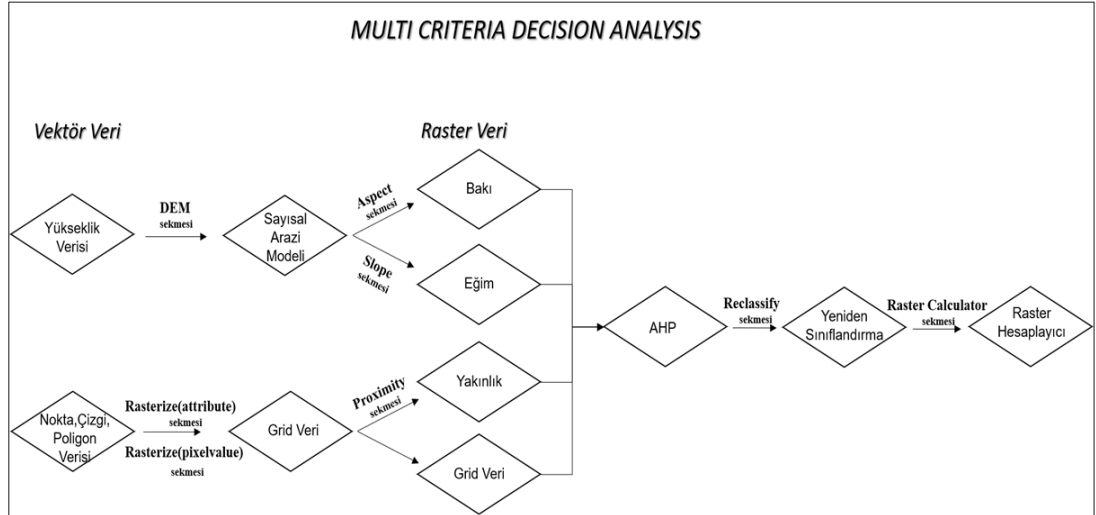
3.3. Arayüz Tasarımı

QGIS3.12 programı ile birlikte kurulumu yapılan “Qt Designer with QGIS 3.12.0 customwidgets” arayüz tasarımları için geliştirilmiş bir tasarım aracıdır. Standart Qt Designer programına ek olarak Widget Box’ında “QGIS Custom Widget” aracına sahiptir. “QGIS CustomWidget” aracı, eklenti tasarımında yeni coğrafi araçlar tasarlanırken tasarımcıya QGIS 3 programında hali hazırda bulunan ön tanımlı bazı araçları sunmaktadır.

Geliştirilmesi planlanan MultipleCriteriaDecisionAnalysis eklentisinin arayüz tasarımında hangi araçlara sahip olması gerektiği ve menü yerleşimi bir diyagram ile belirlenmiştir. Diyagram oluşturulurken QGIS 3 yazılımında ön tanımlı sunulan araç isimleri tercih edilerek QGIS 3 yazılımını daha önce kullanan kullanıcılar için kolay anlaşılabilir olması düşünülmüştür (Şekil 3.4).



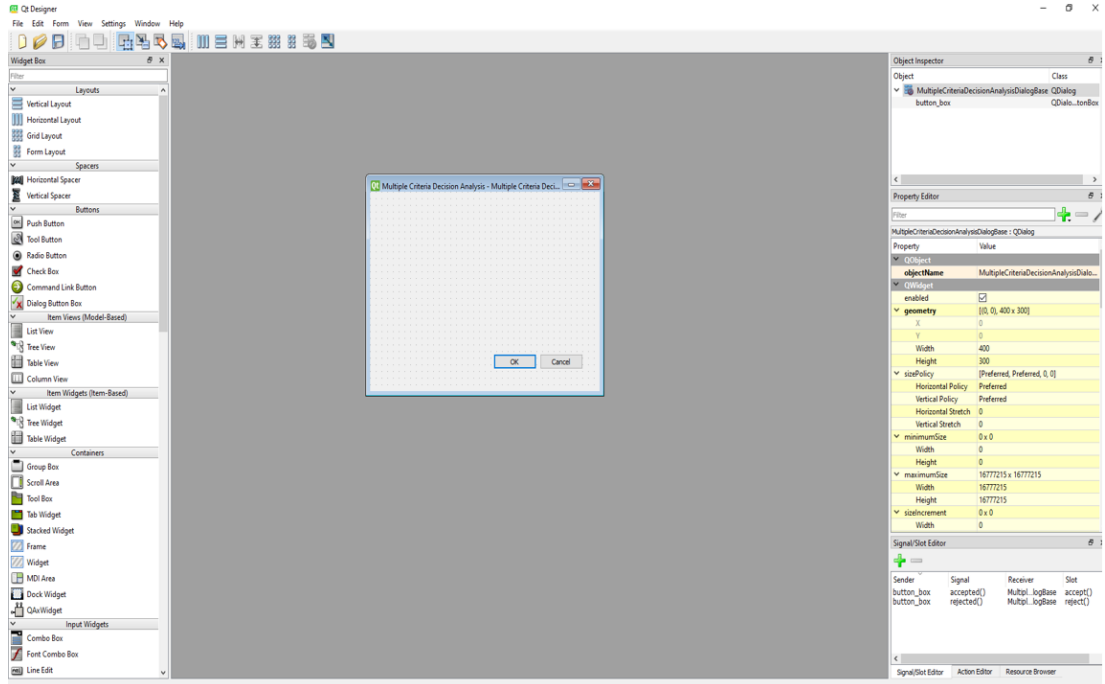
Şekil 3.4. Eklenti menüleri.



Şekil 3.5. Eklenti menü akış diyagramı.

3.3.1. Qt Designer

Plugin Builder3 eklentisinin tasarım yapmak için sunduğu pencerede Tool button with dialog, Tool button with dock widget ve Processing Provider seçenekleri bulunmaktadır. MultipleCriteriaDecisionAnalysis eklenti tasarımında tool button with dialog seçeneği tercih edilmiştir. Dialog kullanıcıdan bir seçenek seçmesini isteyen ya da kullanıcıyı bilgilendirmede kullanılan pencere tipidir (Şekil 3.6).



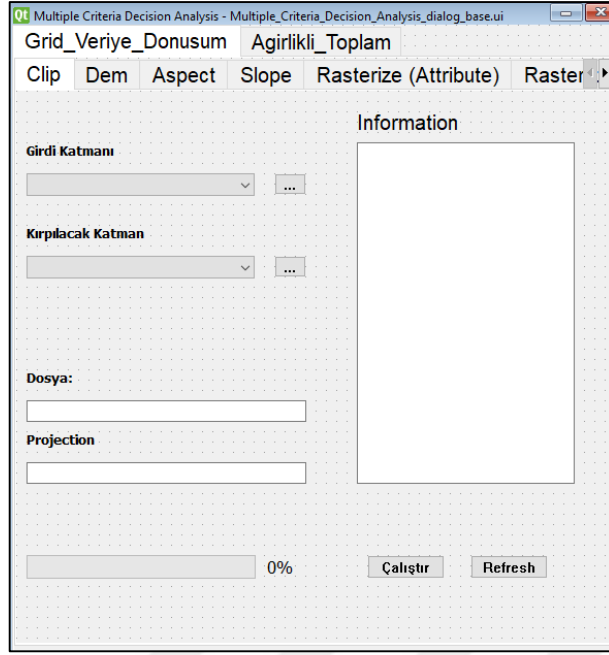
Şekil 3.6. Plugin Builder3 ile üretilen .ui dosyasının Qt Designer'daki ham hali.

MultipleCriteriaDecisionAnalysis eklentisinin menü tasarımı daha önce akış diyagramında planlandığı şekliyle Qt Designer programında tasarlanmıştır. Tasarım için gerekli Widget'ler Qt Designer programında yer alan WidgetBox'tan sağlanmıştır. Menü tasarımı için QTabWidget sınıfından TabWidget'den yararlanılmıştır. Oluşturulan tüm sınıflara ait bilgilere Qt Designer programında yer alan Object Inspector'den erişilebilmektedir. “Grid Veriye Dönüşüm” ve “Ağırlıklı Toplam” isimli 2 adet menü oluşturulmuştur.

“Grid Veriye Dönüşüm” menüsü ilk olarak vektör haldeki yükseklik verisinin raster veri tipindeki DEM (sayısal yükseklik modeli) verisine dönüştürülmesini sağlar. Daha sonra DEM verisinden eğim ve baki haritalarının üretilmesi işlemlerini gerçekleştirir. Rasterize ve Proximity sekmeleri ise vektör haldeki veri tipinden raster veri tipine dönüştürülmesi ve yakınlık durumu haritalarının üretimi işlemlerini gerçekleştirir.

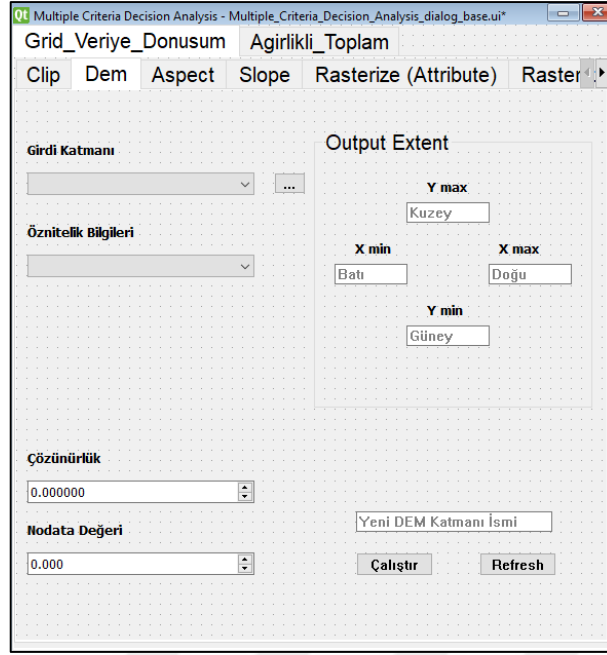
“Ağırlıklı Toplam” menüsü ise Tutarlılık Hesabı sekmesi ile ikili karşılaştırma matrisi oluşturulduktan sonra oluşturulan karşılaştırma matrisinin tutarlılık analizi yapılması ve analiz neticesinde kriter ağırlıklarının hesaplanması yapıldıktan sonra raster verilere Reclassify (yeniden sınıflama) sekmesi ile yeni piksel değerlerinin atanması ve Raster Calcultor sekmesi kullanılarak kriterlerin ağırlıklı toplamı gerçekleştirilmesi amacıyla tasarlanmıştır.

Clip sekmesi tasarımında WidgetBox'tan vektör verileri içe aktarmak için 2 adet ToolButton, çalışma sınırına ait katmanın ve ayrılacak katmanların seçilebilmesi için 2 adet ComboBox, import edilen katmanların klasör yolu ve projeksiyon bilgilerinin gösterilmesi için 2 adet LineEdit, işlem gerçekleşmeden önce ve gerçekleştikten sonra işlem ile ilgili bilgi vermek amacıyla TextEdit, gerçekleştirilen işlemin zamana bağlı bilgisinin kullanıcıya verilmesi için progressBar, seçilen katmanlar ile işlemi başlatabilmek için Çalıştır isimli ve yenilemek için Refresh isimli 2 adet PushButton ayrıca tüm Widget'lerin hangi işlemleri yerine getirdiğini tanımlamak için Label'ler kullanılmıştır. Clip sekmesi çalışma sınırlarının belirlenmesinde kullanılması amacıyla tasarlanmıştır (Şekil 3.7).



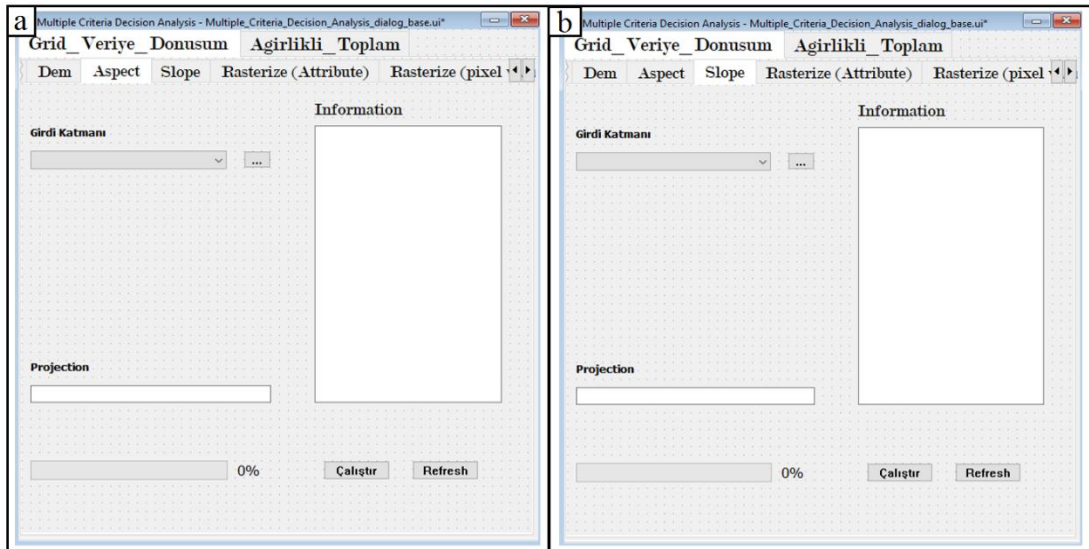
Şekil 3.7. Clip sekmesi.

Dem sekmesi tasarımında sayısal arazi modeli oluşturmada kullanılacak vektör verileri içe aktarmak için ToolButton, içe aktarılan vektör katmanına ve bu vektör dosyasının içinde yer alan öznelik bilgilerine erişim için 2 adet ComboBox, Çözünürlük ve Nodata değerleri girişi için 2 adet DoubleSpinBox, uygulama sınırının koordinatlarını göstermesi için GroupBox ve içinde 4 adet LineEdit, Dem katmanına yeni isim vermek için LineEdit, işlemi başlatabilmek için Çalıştır isimli PushButton ve tüm Widget'lerin hangi işlemleri yerine getirdiğini tanımlamak için Label'lar kullanılmıştır. Qt Designer da yer alan PropertyEditör'de GroupBox ta gösterilecek koordinat bilgilerinin hassasiyeti için coğrafi ve utm koordinat sistemleri düşünülerek maxLength 10 olarak belirlenmiştir. Ayrıca Property Editör yardımıyla Çözünürlük isimli DoubleSpinBox'ın hassasiyeti için decimals 6, minimum değeri 0, maximum değeri 1000 olarak, NoData Değeri isimli DoubleSpinBox'ın hassasiyeti için decimals 3, minimum değeri 0, maximum değeri 0.01 olarak belirlenmiştir (Şekil 3.8).



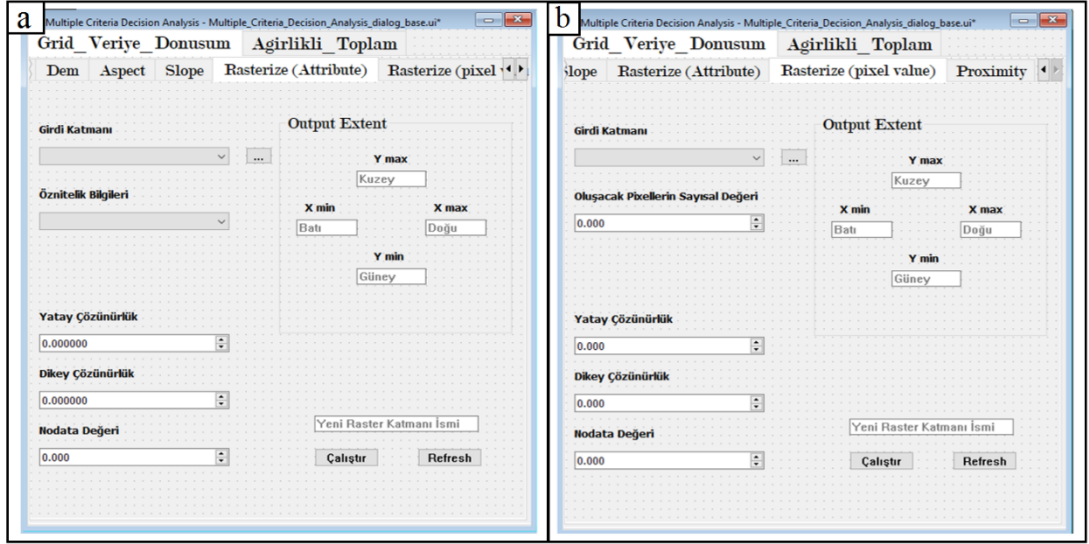
Şekil 3.8. Dem sekmesi.

Aspect ve Slope sekmeleri ToolButton, ComboBox, LineEdit, TextEdit, progressBar, PushButton ve Label'lar kullanılarak ara yüzleri özdeş olarak, kullanılan sınıflar ise Clip sekmesine benzer şekilde tasarlanmıştır (Şekil 3.9).



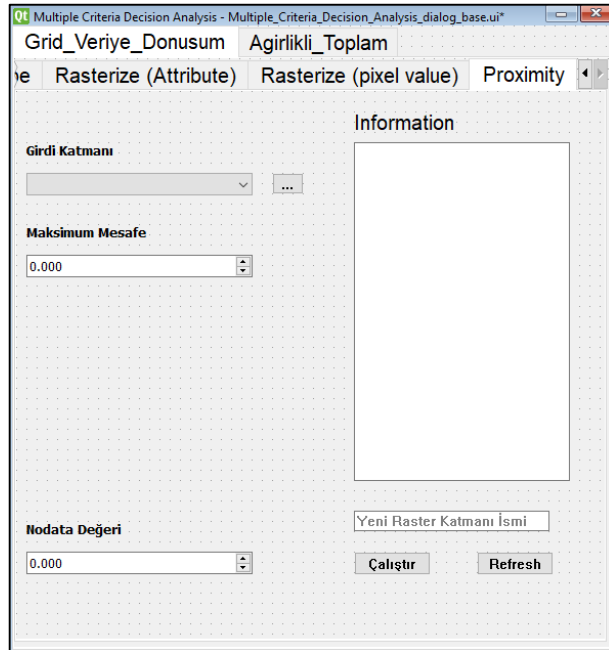
Şekil 3.9. Grid veriye dönüşüm menüsü sekmeleri; a) aspect, b) slope.

Rasterize (attribute) ve Rasterize (pixelvalue) sekmeleri tasarımlarında da Dem sekmesine benzer şekilde ToolButton, ComboBox, GroupBox, LineEdit, PushButton, DoubleSpinBox ve Label'lar kullanılarak tasarımları tamamlanmıştır (Şekil 3.10).



Şekil 3.10. Grid veriye dönüşüm menüsü sekmeleri; a) rasterize (attribute), b) rasterize (pixelvalue).

Proximity sekmesi tasarımında da diğer sekmelere benzer şekilde ToolButton, ComboBox, LineEdit, PushButton, DoubleSpinBox ve Label'lar kullanılarak tasarım tamamlanmıştır. Vektör veri ile ilgi işlemleri gerçekleştiren Grid Veriye Dönüşüm Menüsüne ait tüm sekmelerde Widget'ler konumlandırılırken birbirine benzer tasarımlar oluşturularak kullanıcı için kolay anlaşılabilir ve basit kullanımlı bir menü tasarımı amaçlanmıştır (Şekil 3.11).



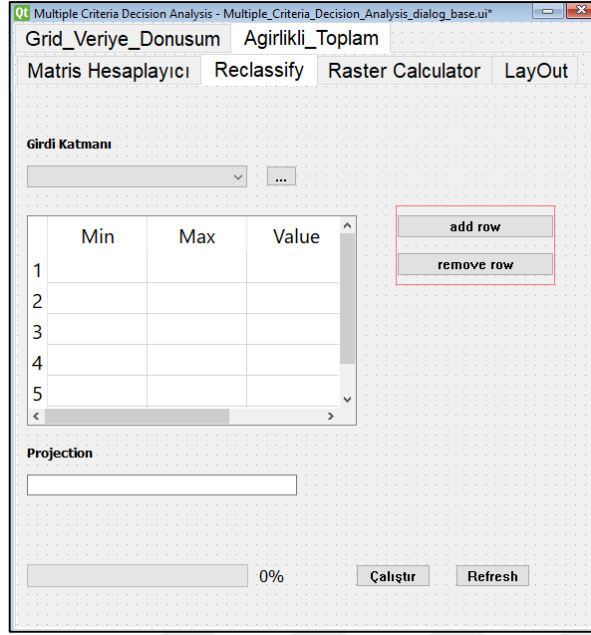
Şekil 3.11. Proximity sekmesi.

Matris Hesaplayıcı sekmesi çok ölçütlü karar analiz teknikleri kullanılırken mevcut karşılaştırma matrisi değerlerinden; normalize edilmiş matris, rassal indeks, ağırlık, tutarlılık vektörü ortalaması, tutarlılık indeksi ve tutarlılık oranı değerlerinin hesaplanması için tasarlanmıştır. Matrisin tutarlılık oranı 0,10'dan küçük olması durumunda matris tutarlı kabul edilmektedir. Tasarlanan Tutarlılık Hesaplayıcı sekmesinde gerekli hesaplamalar yapılırken kapalı kaynak kodlu Ms Excel programı vb. programlara ihtiyaç duyulmadan yapılabilmesi amaçlanmıştır. Ayrıca matris tutarlılık oranının 0,10'dan büyük olması durumunda kullanıcı, bir uyarı mesajı ile uyarılarak olası hataların önüne geçilmesi sağlanmıştır (Şekil 3.12).

$n \times n$	$n Rİ$
1	0
2	0
3	0.58
4	0.90
5	1.12
6	1.24
7	1.32
8	1.41

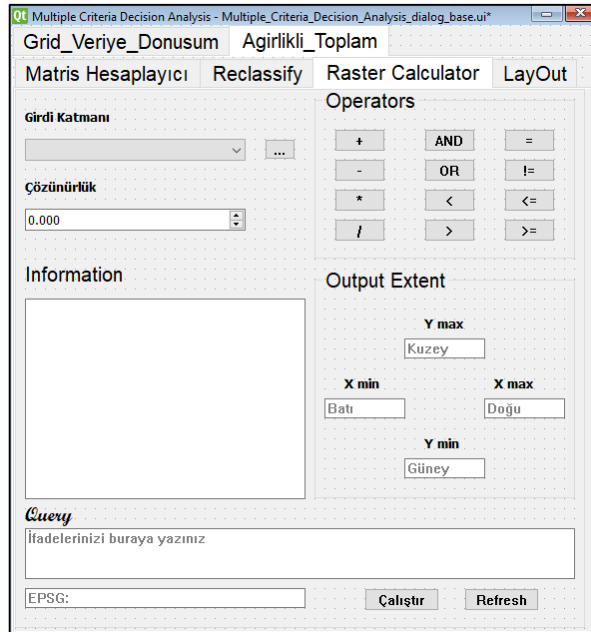
Şekil 3.12. Matris hesaplayıcı sekmesi.

Reclassify sekmesi tasarımında daha öncedeki tasarımlarda kullanılan ToolButton, ComboBox, LineEdit, PushButton 'a ek olarak piksel değerlerini yeniden atayabilmek için veri girişlerinin yapılabileceği min, max ve value olarak 3 sütüne ayrılmış TableWidget ayrıca satır ekleme, çıkarma işlemlerini yapabilmek için Vertical Layout ve etiketleme için Label'ler kullanılarak tasarım tamamlanmıştır (Şekil 3.13).



Şekil 3.13. Reclassify sekmesi.

Raster Calculator sekmesi tasarımında ToolButton, ComboBox, GroupBox, LineEdit, TextEdit, PushButton, DoubleSpinBox ve Label'ler kullanılarak tasarımları tamamlanmıştır. Raster veri ile ilgi işlemleri gerçekleştiren Ağırlıklı Toplam Menüsüne ait sekmelerde Widget'ler konumlandırılırken QGIS 3 yazılımı ara yüzüne benzer bir tasarım yapılarak kullanıcı için kolay öğrenilebilir ve basit kullanımlı bir menü tasarımı amaçlanmıştır (Şekil 3.14).



Şekil 3.14. Raster calculator sekmesi.

3.4. Eklenti Kodlama

Coğrafi işlemler için gerekli fonksiyonlar geliştirilirken PyQgis'den yararlanılmıştır. PyQgis, Python araçları içeren QGIS içindeki Python ortamıdır. Ayrıca PyQt modülüne ek olarak vektör ve raster veri formatları ile işlem yapılabilmesi için Gdal/Ogr kütüphaneleri de import edilmiştir. Qt Designer programı ile ara yüz tasarımı tamamlandıktan sonra tasarımda kullanılan tüm widget'lerin işlevlerini yerine getirebilmesi için gerekli fonksiyonların Signal-Slot bağlantılarının yapılması gerekmektedir. Signal-Slot bağlantıları MultipleCriteriaDecisionAnalysis.py dosyasının içinde yer alan "run" fonksiyonun içine yazılmıştır.

Plugin Builder3 eklentisi ile üretilen MultipleCriteriaDecisionAnalysis.py dosyasında PyQt kütüphanesindeki QtCore modülüne ait QSettings, QTranslator ve QCoreApplication sınıflarının,QtGui modülüne ait QIcon sınıfının ve QtWidgets modülüne ait QAction sınıfının import edilerek oluşturulduğu görülmektedir (Şekil 3.15).

```
# Multiple_Criteria_Decision_Analysis.py - C:\Users\user\AppData\Roaming\QGIS\QGIS3\profiles\default\python\plugins\multiple_criteria_decision_analysis\Multiple_Criteria_Decision_Analysis.py (3.8.1)
File Edit Format Run Options Window Help
1 # -*- coding: utf-8 -*-
2 """
3 /*****
4 MultipleCriteriaDecisionAnalysis
5 A QGIS plugin
6 Multiple Criteria Decision Analysis
7 Generated by Plugin Builder: http://g-sherman.github.io/Qgis-Plugin-Builder/
8 -----
9 begin : 2022-02-22
10 git sha : $Format:%H$
11 copyright : (C) 2022 by -
12 email : -
13 *****/
14
15 /*****
16 * This program is free software; you can redistribute it and/or modify *
17 * it under the terms of the GNU General Public License as published by *
18 * the Free Software Foundation; either version 2 of the License, or *
19 * (at your option) any later version. *
20 * *
21 * *****/
22 """
23
24 from qgis.PyQt.QtCore import QSettings, QTranslator, QCoreApplication
25 from qgis.PyQt.QtGui import QIcon
26 from qgis.PyQt.QtWidgets import QAction
27
28 # Initialize Qt resources from file resources.py
29 from .resources import *
30 # Import the code for the dialog
31 from .Multiple_Criteria_Decision_Analysis_dialog import MultipleCriteriaDecisionAnalysisDialog
32 import os.path
33
34
35 class MultipleCriteriaDecisionAnalysis:
36     """QGIS Plugin Implementation."""
37
38     def __init__(self, iface):
39         """Constructor.
40
41         :param iface: An interface instance that will be passed to this class
42                       which provides the hook by which you can manipulate the QGIS
43 """
```

Şekil 3.15. Üretilen .py dosyasında hazır gelen modüller ve sınıflar.

Eklenti kodlama işlemine ihtiyaç duyulan kütüphanelerdeki sınıf ve modüller import edilerek başlanmıştır. Şekil 3.16’da Multiple Criteria Decision Analysis eklentisinde kullanılan tüm modüller ve sınıflar gösterilmiştir.

```
Multiple_Criteria_Decision_Analysis.py - C:\Users\user\AppData\Roaming\QGIS\QGIS3\profiles\default\python\plugins\multiple_criteria_decision_analysis\Multiple_Criteria_Decision_Analysis.py (3.9.1)
File Edit Format Run Options Window Help
26 import os,sys,time
27 import os.path
28 import ogr,gdal
29
30 from qgis.PyQt import uic,QtGui,QtWidgets,QtCore
31 from qgis.PyQt.QtCore import QSettings, QTranslator, QCoreApplication,QVariant,Qt
32 from qgis.PyQt.QtGui import *
33 from qgis.PyQt.QtGui import QIcon,QColor
34 from qgis.PyQt.QtWidgets import *
35 from qgis.PyQt.QtWidgets import (QAction, QFileDialog,QTabWidget,QProgressBar,
36                                 QMessageBox,QDialog,QSpinBox, QApplication, QLabel,
37                                 QMainWindow, QWidget, QTabBar,QTableWidget,QTableWidgetItem)
38
39
40 from qgis import processing
41 from qgis.utils import iface
42 from qgis.core import *
43 from qgis.core import (QgsProcessing,
44                        QgsProcessingAlgorithm,
45                        QgsProcessingException,
46                        QgsProcessingOutputNumber,
47                        QgsProcessingParameterDistance,
48                        QgsProcessingParameterFeatureSource,
49                        QgsProcessingParameterVectorDestination,
50                        QgsProcessingParameterRasterDestination)
51
52 from qgis.core import (
53     QgsRasterLayer,
54     QgsProject,
55     QgsPointXY,
56     QgsRaster,
57     QgsRasterShader,
58     QgsColorRampShader,
59     QgsSingleBandPseudoColorRenderer,
60     QgsSingleBandColorDataRenderer,
61     QgsSingleBandGrayRenderer,
62 )
63
64 # import * ifadesi kütüphane içindeki tüm modüllere ve sınıflara erişimi sağlar
65 # Initialize Qt resources from file resources.py
66 from .resources import *
67 # Import the code for the dialog
```

Şekil 3.16. Eklenti için kullanılan modüller ve sınıflar.

3.4.1. Pyqgis

PyQGIS ile çalışmak için öncelikle qgis.core modülü import edilmelidir. Bu modül sayesinde QGIS yazılımında bulunan birçok CBS fonksiyonu kullanılarak etkileşimli uygulamalar geliştirilebilir. QGIS ortamıyla etkileşime geçmek için QgisInterface sınıfının iface değişkeni kullanılır. Bu ara yüz ile harita çizim alanına, menülere, araç kutularına ve QGIS programının diğer parçalarına erişilebilir (URL-6).

3.4.2. Fonksiyonların kodlanması

Akış diyagramında yer alan fonksiyonlar kodlanmadan önce her fonksiyonun ortak kullanacağı fonksiyonlar tespit edilerek kodlama işlemine başlanmıştır. Bu kapsamda dosya_ac(), vector_sec(), raster_sec(), projection_check(), name_check(), progress() isimli fonksiyonlar ilk olarak yazılmıştır.

1. dosya_ac(): Bu fonksiyon eklentideki herhangi bir fonksiyon çalıştırıldığında masaüstüne AHP isimli yeni bir klasör oluşturarak oluşacak tüm dosyaları bu klasöre kaydedecektir.
2. vector_sec (): Vektör veri formatındaki (.shp,.geojson ve .gml) verileri QGIS 3 yazılımına import etmek için yazılmıştır. Clip, Dem, Rasterize (attribute) ve Rasterize (pixelvalue) fonksiyonlarına ait toolbutton'lara tıklandığında çalışacak şekilde kodlanmıştır.
3. raster_sec(): Raster veri formatındaki (.tif, .tiff) verileri QGIS 3 yazılımına import etmek için yazılmıştır. Aspect, Slope, Proximity, Reclassify ve Raster Calculator fonksiyonlarına ait toolbutton'lara tıklandığında çalışacak şekilde kodlanmıştır.
4. projection_check(): Clip, Aspect, Slope, ve RasterCalculator fonksiyonlarında kullanılan verinin hangi projeksiyon sistemine ait olduğu bilgisini LineEdit ile kullanıcıya verir.
5. name_check(): AHP klasörünün içerisinde daha önce aynı isimle kaydedilmiş dosya olup olmadığını kontrol eder.
6. progress(): ProgressBar aracılığıyla devam eden işlemin zamana bağlı değişimini kullanıcıya gösterir.
7. Daha sonra akış diyagramında yer alan “Grid Veriye Dönüşüm” ve “Ağırlıklı Toplam” menülerinin fonksiyonları; Clip, Dem, Aspect, Slope, Rasterize (attribute), Rasterize (pixelvalue), Proximity, Matris_Calculator(), Reclassify, Raster_Calculator sırasıyla yazılmıştır.
8. Clip(): Ayırma işlemini gerçekleştirir. Çalışma sınırının belirlenmesi amacıyla kodlanmıştır.
9. Dem(): Vektör veri formatındaki yükseklik verisinden raster veri formatında sayısal arazi modeli verisi üretebilmek amacıyla kodlanmıştır.

10. Aspect(): Oluşturulan dem verisinden raster veri tipinde bakı haritası üretebilmek için kodlanmıştır.
11. Slope(): Oluşturulan dem verisinden raster veri tipinde eğim haritası üretebilmek için kodlanmıştır.
12. Rasterize (attribute): Vektör veri formatındaki çizgi ve poligon verisinden piksellerinin sayısal değeri olarak .dbf dosyasından istenilen öz nitelik bilgisinin seçilerek oluşturulacak raster veriler üretebilmek için kodlanmıştır.
13. Rasterize (pixelvalue): Vektör veri formatındaki çizgi ve poligon verisinden piksellerinin sayısal değeri olarak istenilen sayısal değerin atanabileceği raster veriler üretebilmek için kodlanmıştır.
14. Proximity(): Yakınlık durumu haritalarının üretilmesi için kodlanmıştır.
15. Matris_Calculator(): AHP Hesaplayıcı sekmesi olarak tasarlanmıştır. Çok Ölçütlü karar analiz teknikleri kullanılarak mevcut karşılaştırma matrisi değerlerinden; normalize edilmiş matris, rassal indeks, ağırlık, tutarlılık vektörü ortalaması, tutarlılık indeksi ve tutarlılık oranı değerlerinin hesaplanması için kodlanmıştır.
16. Reclassify(): Oluşturulan raster verilere yeni piksel değerlerinin atanması için kodlanmıştır.
17. Raster_Calculator(): Kriter ve alt kriterlere ait tutarlılık oranları ile piksel değerleri çarpılarak ve toplanarak yeni piksel değerlerine sahip tek bir raster verisi elde edilmesi amacıyla raster verilerin piksel değerleri üzerinde matematiksel işlemler yapmak için kodlanmıştır.

```
219 def dosya_ac():
220
221     AHP = os.path.join(os.path.expanduser('~'), 'Desktop', 'AHP')
222     desktop = os.path.join(os.path.expanduser('~'), 'Desktop')
223
224     for file in os.listdir(desktop):
225         if file.name == "AHP":
226             break
227     if file.name != "AHP":
228         os.mkdir(AHP)
229
230     if os.sep=="\\":
231         AHP=AHP.replace("\\", "/")
232         desktop=desktop.replace("\\", "/")
233     dosya_ac()
```

Şekil 3.17. Dosya_ac() fonksiyonu kodları.

Şekil 3.17’de gösterilen dosya_ac() fonksiyonuna ait kodlar Windows işletim sistemi kullanıcıları için eklenti çalıştırıldığında masaüstünde AHP isimli bir klasör oluşturmaktadır. Eklentideki her bir fonksiyon çalıştırıldığında .tif, .prj, .xml vb. uzantılı çok sayıda dosya oluştuğu için tüm output dosyalarının AHP isimli klasöre üretilmesi ile dağınık şekilde depolanmasının önüne geçilmesi amaçlanmaktadır.

```

240 def vector_sec(self):
241     shapefilePath=""
242     self.shapefilePath=shapefilePath
243
244     self.shapefilePath, shapefileType = QFileDialog.getOpenFileName(self.dlg,\
245     "Shapefile dosyasını seciniz","", "ESRI Shapefiles (*.shp *.SHP);; \
246     GeoJSON (*.geojson);; Geography Markup Language (*.GML)")
247
248     self.shp = ogr.Open(self.shapefilePath)
249     self.layer = self.shp.GetLayer(0)
250     self.name = self.layer.GetName()
251     self.layer.GetName()
252     self.layerDef = self.layer.GetLayerDefn() # içindeki tüm geospatial veriye erişmeyi sağlar
253
254
255     if self.layerDef.GetGeomType() == ogr.wkbLineString or ogr.wkbMultiLineString:
256         self.vlayer = QgsVectorLayer(self.shapefilePath,self.name,"ogr")
257         self.dlg.lineEdit.setText(self.shapefilePath)
258
259     else:
260         self.error_msj("Seçtiğiniz katman geometrisi sadece çizgi ve poligon geometrisi olabilir !")
261         self.dlg.lineEdit.setText("")
262         return
263     QgsProject.instance().addMapLayer(self.vlayer)
264
265
266 def raster_sec(self):
267     self.rasterfilePath, rasterfileType = QFileDialog.getOpenFileName(self.dlg,\
268     "Raster dosyasını seciniz","", "TIFF File (*.tiff *.tif)")
269
270     self.tif = gdal.Open(self.rasterfilePath)
271     self.rname=self.rasterfilePath.split("/")
272     self.rname=self.rname[-1]
273     self.rname2=self.rname.split(".")
274     self.rname2=self.rname2[0]
275
276     if self.rname2 != "":
277
278         self.rlayer = QgsRasterLayer(self.rasterfilePath,self.rname2,"gdal")
279         QgsProject.instance().addMapLayer(self.rlayer)
280     else:
281         pass

```

Şekil 3.18. Vektor_sec() ve raster_sec() fonksiyonları kodları.

Şekil 3.18’de vector_sec() ve raster_sec() fonksiyonlarına ait kodlar incelendiğinde; vektor_sec() fonksiyonu Clip, Dem, Rasterize(attribute) ve Rasterize(pixelvalue) fonksiyonlarına ait toolbar’lara tıklandığında PyQt. QtWidgets içindeki QFileDialog sınıfından yararlanarak vektör formattaki (.shp, .geojson ve. gml) verileri, OGR vektör veri kütüphanesi yardımıyla istenilen fonksiyon içerisinde çalıştıracak şekilde kodlanmıştır.

raster_sec() fonksiyonu ise Aspect, Slope, Proximity, Reclassify ve RasterCalculator fonksiyonlarına ait toolbar’lara tıklandığında PyQt. QtWidgets içindeki

QFileDialog sınıfından yararlanarak raster formattaki (.tif, .tiff) verileri, GDAL raster veri kütüphanesi yardımıyla istenilen fonksiyon içerisinde çalıştıracak şekilde kodlanmıştır.

```

559 #####
560
561 def Slope(self):
562
563     desktop = os.path.join(os.path.expanduser('~'), 'Desktop', 'AHP', 'slope.tif')
564     if os.sep=="\\":
565         desktop=desktop.replace("\\", "/")
566     else:
567         pass
568
569
570     processing.run("gdal:slope", {'INPUT':self.slopegirdi_katmani,'BAND':1,'SCALE':1,'AS_PERCENT':False,\
571 'COMPUTE_EDGES':False,'ZEVENBERGEN':False,'OPTIONS':'','EXTRA':'','OUTPUT':desktop})
572
573
574     slope_path = os.path.join(QgsProject.instance().homePath(),\
575 'sloped',\
576 desktop)
577
578
579     layer = iface.addRasterLayer(slope_path, 'slope.tif', 'gdal' )
580     if not layer:
581         print("Layer failed to load!")
582
583     self.slope_info = "Eğim işleminiz gerçekleştirildi, masaüstünüzde bulunan AHP klasörüne işleme ait dosyaları bulabilirsiniz."
584     self.dlg.textEdit_5.setPlainText(self.slope_info)
585
586
587 def Slope_comboBoxes(self):
588     self.dlg.comboBox_6.insertItem(0,"Seçiniz")
589     for layer in QgsProject.instance().mapLayers().values():
590         self.dlg.comboBox_6.addItem([layer.name()])
591         self.dlg.comboBox_6.itemText(0)
592
593     self.dlg.comboBox_6.currentIndexChanged['int'].connect(self.comboBox_6)
594
595 def comboBox_6(self,current_index):
596     if (current_index!=0):
597         slopegirdi_katmani=self.dlg.comboBox_6.itemText(current_index)
598         self.slopegirdi_katmani=slopegirdi_katmani
599
600 #####

```

Şekil 3.19. Slope() fonksiyonu kodları.

Şekil 3.19’da örnek slope() fonksiyonuna ait kodlar incelendiğinde; fonksiyonun ilk olarak üretilecek eğim haritası için masaüstünde yer alan AHP klasörü içerisine slope.tif isimli yeni bir raster dosyası üreteceği görülmektedir. Ana bileşen olan “processing.run” kısmı GDAL kütüphanesinden yararlanarak fonksiyonun eğim haritası üretmesini sağlamaktadır. Slope sekmesinin ara yüz tasarımıyla yer alan TextEdit ile eğim haritasının üretildiği bilgisi kullanıcıya sunulacaktır. Tasarlanan comboBox ile Dem sekmesi yardımıyla oluşturulan ve içeri aktarılan sayısal arazi modeli verisinin tabaka olarak listelenmesi ve istenildiğinde seçilebilmesi için Slope_comboBoxes() ve comboBox_6 isimli fonksiyonlar kodlanmıştır. Slope_comboBoxes() fonksiyonu ile “Seçiniz” listenin ilk elemanı, içe aktarılan Dem verileri ise listenin diğer elemanları olarak listelenecektir. comboBox_6 fonksiyonu ile listenin ilk elemanı olan “Seçiniz” yerine listenin diğer elemanlarının seçilebilmesi amaçlanmaktadır. Çalıştır butonuna basılması durumunda “processing.run”

fonksiyonu çalışarak oluşan raster formatındaki slope.tiff eğim haritası AHP klasörüne kaydedilecektir. Fonksiyonların çalışabilmesi için signal-slot bağlantılarının yapılması gerekmektedir.

3.4.3. Signal-Slot kullanımı

Signal ve Slot Qt'nin temel fonksiyonlarıdır. Arayüz programlamada bir nesne üzerinde yapılan değişiklik başka bir nesne ile iletişim kurmalıdır. Qt içindeki tüm widgetler kullanıcıdan gelen bir eylemi veya bir durum değişikliğini belirtmek için signaller yayar. Signaller herhangi bir nesnenin program içindeki anlık durumundaki değişiklikten sonra programa yayılan global nesnelerdir. Slot ise QT tarafından çağrılabilen ve alınan signal'lere göre çalışan fonksiyonlardır. Qt'de signal'ler ve slot'lar önceden tanımlanmıştır. QObject veya QWidget sınıflarının altındaki tüm nesnelerde signal ve slot bulunmaktadır. Ayrıca kullanıcılar önceden tanımlanmış signal ve slot'lardan farklı olarak kendi signal ve slot'larını tanımlayabilirler.

Signal'ler yayımlandığında eğer signal bir fonksiyona bağlı ise fonksiyon içerisindeki işlem gerçekleştirilir. Eklenti tasarımında bulunan tüm buton, combobox vb. nesneler ancak fonksiyonları tanımlanıp signal-slot bağlantıları tamamlandıktan sonra çalıştırılabilirler. Kodlama evresinde MultipleCriteriaDecisionAnalysis eklentisine ait tüm fonksiyonların kodlanması tamamlanmıştır. Ancak signal-slot bağlantıları yapılmadığı için fonksiyonları tetikleyecek signaller gönderilememektedir. Eklenti programlamanın son aşaması olan signal-slot bağlantıları kurularak, run() fonksiyonun içine yazılan kodlarla signallerin gönderilmesi ve fonksiyonların istenildiği gibi çalışabilmesi sağlanacaktır.

```
1777 #Slope
1778 self.dlg.textEdit_5.setEnabled(False)
1779 self.dlg.toolButton_7.clicked.connect(self.raster_sec)
1780 self.dlg.toolButton_7.clicked.connect(self.Slope_comboBoxes)
1781 self.dlg.toolButton_7.clicked.connect(self.projection_check)
1782 self.dlg.pushButton_10.clicked.connect(self.Slope)
```

Şekil 3.20. Slope() fonksiyonu signal-slot bağlantıları.

Şekil 3.19'da örnek slope() fonksiyonuna ait kodlar tasarlanmış ancak signal-slot bağlantıları olmadığı için fonksiyona ait kodlar çalışmamaktadır. Bu amaçla Şekil 3.20'de gösterilen Slope() fonksiyonu signal-slot bağlantıları tamamlanmıştır.

toolButton_7 butonu “clicked” sinyali ile raster_sec(), Slope_comboBoxes(), projection_check() fonksiyonlarına bağlanmaktadır. Bağlantı kurulduğu için toolButton_7 butonuna tıklandığında sırasıyla raster_sec() fonksiyonu çalışır. Raster verinin seçilmesi sağlanır. Slope_comboBoxes() fonksiyonu çalışır ve raster veriler listelenerek seçilebilir hale getirilir, son olarak ise projeksiyon bilgisi LineEdit’e yazdırılır.

pushButton_10 butonu ise “clicked” sinyali ile Slope fonksiyonuna bağlanmaktadır. Bağlantı kurulduğu için pushButton_10 (Çalıştır) butonuna tıklandığında Slope() fonksiyonu çalışarak “processing.run” fonksiyonunu çalıştıracak ve raster formattaki slope.tiff eğim haritası AHP klasörüne kaydedilecektir.



4. SONUÇ VE ÖNERİLER

Richard Stalman'ın "GNU not Unix" fikrini ortaya attığı 1983 yılından günümüze açık kaynak yazılımlar her geçen yıl artan bir ivmeyle gelişimini devam ettirmektedir. AKY'ların ticari yazılımlara göre maliyetsiz ve güvenilir oluşunun yanı sıra kullanıcılarının kendi amaçları doğrultusunda yazılımı inceleme, gereksinimleri için ise değiştirme ve yazılım üzerindeki her türlü değişikliği diğer kullanıcılara paylaşma özgürlüğünü istemeleri dünya üzerinde özgür yazılım fikrini ve felsefesini ortaya çıkarmıştır. Dünya üzerinde büyük taraftar toplayan özgür yazılım fikri, ilerleyen yıllarda kurulan topluluklarla AKY ların gelişimini sağlayarak yazılım geliştiriciliği kavramını farklı boyutlara taşımıştır. AKY ların ücretsiz oluşu ve kullanıcıları için geliştirilebilir esnek yapısı dünya üzerindeki birçok özgür yazılım geliştiricisine ilham olmuştur. Bunun neticesinde 1985 yılında özgür yazılımın gelişimi için ilk kurumsal girişim olan Özgür Yazılım Vakfı'nın kurulmasından günümüze kadar bir çok vakıf kurulmuş ve kullanıcılarına ihtiyaç duydukları farklı alanlarda AKY'lar geliştirmişlerdir. CBS nin işlevlerini yerine getirmesini sağlayan temel bileşenlerinden biri olan yazılım, özgür yazılım fikrinin ortaya çıktığı günden günümüze kadar geçen sürede bu gelişimden olumlu yönde etkilenmiş ve AKK birçok CBS yazılımı bu fikir sayesinde gelişmiştir. Bu fikirle 2006 yılında kurulan ve mekansal analiz yazılımları geliştirmeyi amaçlayan OSGeo vakfı açık kaynaklı bir CBS yazılımı olan QGIS'i geliştirmiştir.

AKY'lardaki hızlı gelişim ile birlikte bazı AKY'lar birbirine entegre edilerek kullanılmaya başlanmıştır. Python, Grass, Saga entegre edilen bu yazılımlardan bazılarıdır. Python yıllar içerisinde çok hızlı bir gelişim geçirmiş ve birçok CBS yazılımına entegre edilerek kullanılmaya başlanmıştır. Bu hızlı gelişimin sonucunda 2007 yılında QGIS 0.9'da Python desteği ile kullanıcılarına sunulmuştur. QGIS yazılımı kullanıcılarına mekansal analizlerini gerçekleştirebileceği birçok araç sunarken buna ek olarak kendi eklentilerini geliştirebileceği ve diğer dünya kullanıcıları ile buluşturabileceği bir ortamda sunmaktadır. Bu sayede birçok yeni eklenti geliştirilerek QGIS eklenti deposuna eklenmektedir.

Yapılan literatür araştırmasında son yıllarda AKK CBS yazılımlarının ücretsiz, güvenilir, değiştirilebilir, dağıtılabılır ve kolay erişilebilir gibi avantajları sayesinde

mekansal analizler yapabilmek için ticari amaçlı lisanslı yazılımlar yerine tercih edildiği tespit edilmiştir. İncelemelerde AKK CBS yazılımı olan QGIS yazılımı ile yapılan çalışmaların son onbeş yılda arttığı sonucu görülmektedir. Ayrıca yapılan araştırmalardan çıkan bir diğer sonuç ise ÇÖKV yöntemleri kullanılarak yapılan CBS projelerindeki artış olmuştur. Literatürde ÇÖKV yöntemleri kullanılan ve AKK CBS yazılımları ile yapılmış birçok araştırma ve proje tespit edilmiştir. Ancak sonuç çıktı ve görsellerinin oluşturulması amacıyla küçük kod parçalarından oluşturulmuş birtakım eklentiler haricinde özgür yazılımcılar tarafından geliştirilmiş, ÇÖKV yöntemlerinin kullanımı için tasarlanmış çok az sayıda eklentiye rastlanmamıştır. Ticari bir yazılım olan NetCAD yazılımındaki mimaride yer alan analiz modülü altında ÇÖKV yöntemleriyle bir model geliştirilmiş olmasına karşın AKK CBS yazılımları için bu çapta bir tasarım tespit edilmemiştir.

Bu çalışmada QGIS yazılımı ile ÇÖKV yöntemlerinin kullanımını kolaylaştıracak özgür bir CBS eklentisi tasarlanırken aynı zamanda eklenti geliştirme aşamalarının kullanıcılara paylaşılması ile kendi özgür eklentilerini geliştirmek isteyen kullanıcılara rehber niteliğinde olması amaçlanmıştır. Çalışma sırasında karşılaşılan problemler ve bu problemlere yönelik çözüm önerileride bu çalışma kapsamında eklenti geliştirmek isteyen kullanıcılara aktarılmaya çalışılmıştır. Eklenti geliştirme aşamaları en basit şekilde kurgulanmaya çalışılmıştır. Bu kapsamda öncelikle diğer özgür yazılımcılar tarafından tasarlanan plugin builder3 eklentisi ile yeni bir eklenti üretilmiştir. Üretilen eklenti için QGIS yazılımının beraberinde sunulan grafiksel kullanıcı ara yüzler tasarlamayı sağlayan Qt Designer yazılımı ile arayüz tasarımı tamamlanmış ve tasarımın her aşaması ekran görüntüleri aracılığıyla eklenti geliştirmek isteyen kullanıcılara detaylı olarak aktarılmıştır. Bu aşamadan sonra coğrafi araçlar açık kaynaklı python programlama dili ve QGIS API'sini ile eklenti geliştirmede kullanılan PyQGIS bağdaştırıcısından yararlanılarak kodlanmıştır. Coğrafi araçlar, ÇÖKV uygulamalarında kullanılan CBS yazılımlarının arayüz tasarımları dikkate alınarak tasarlanmıştır. Böylece kullanıcı için kolay anlaşılabilir olması hedeflenmiştir. Ağırlıklı toplam menüsünde yer alan Matris Hesaplayıcı sekmesi kodlanırken matematiksel işlemler için python'ın popüler kütüphanesi olan Numpy kütüphanesinden yararlanmak düşünülmüşse de kullanıcıların eklentiyi kullanmadan önce kütüphane import etmesinin temel seviye kullanıcılar için uygun olmadığı düşünüldükçe vazgeçilmiştir. Coğrafi araçlar ile arayüz arasındaki signal-slot

bağlantılarının kodlaması ile birlikte eklenti kullanıma hazır hale getirilmiştir. Hazır hale getirilen eklenti “<https://github.com/mreylmzz-hub/MCDA-Plugin>” adresinden tüm dünya kullanıcılarının paylaşımına sunulmuştur.



KAYNAKLAR

- Akça, M. D., 2000. Coğrafi Bilgi Sistemi ile çevresel verilerin modellenmesi: Trabzon-Değirmendere Vadisi Örneği, Yüksek Lisans Tezi, Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Trabzon.
- Akın, Ö., Eroğlu, A. ve Demirel, H., 2020. Açık Kaynak Kodlu Ağ Analiz Algoritmaları ve Arazi Örtüsü/Kullanımı simülatörü, Harita Dergisi, 163, 29-37.
- Alkış, Z., 1996. Coğrafi Bilgi Sistemi bileşenleri, Harita ve Kadastro Mühendisliği, 79, 57-63.
- Anbaroğlu, B. ve Serhan, A., 2019. Geomatik Mühendisliğinde Aktif Öğrenme için bir model önerisi: QGIS eklentisi geliştirme, TMMOB 6. Coğrafi Bilgi Sistemleri Kongresi, 23-25 Ekim 2019, Ankara.
- Ayday, C., Yaman, N., Keser, K. ve Sabah, L., 2016. Geographic Information Systems (GIS) with Open-Source Software / Açık Kaynak Kodlu Yazılımlar ile Coğrafi Bilgi Sistemleri (CBS), 4. Uluslararası Coğrafya Sempozyumu, Antalya.
- Ayvaz, U., Çoban, A., Gürüler, H. ve Peker, M., 2016. Python dilinin öznelilikleri, programlama eğitiminde ve yazılım dünyasındaki yeri, XVIII. Academic Bilişim Konferansı, Aydın.
- Beyhan, B. ve Belge, B., 2008. Özgür CBS Yazılımları üzerine karşılaştırmalı bir değerlendirme, 2. Uzaktan Algılama ve Coğrafi Bilgi Sistemleri Sempozyumu UZAL-CBS 2008, Kayseri.
- Beyhan, B., Belge, B. ve Zorlu F., 2010. Özgür ve Açık Kaynak Kodlu Masaüstü CBS yazılımları üzerine karşılaştırmalı ve sistemli bir değerlendirme, Harita Dergisi, 143.
- Bilgilioğlu, S. S., 2014. Ulusal Heyelan Haritaları için Coğrafi Veri Modelinin geliştirilmesi ve Afet Yönetiminde kullanılması, Yüksek Lisans Tezi, Aksaray Üniversitesi, Fen Bilimleri Enstitüsü, Aksaray.
- Bilgilioğlu, S. S., 2022. Coğrafi Bilgi Sistemleri ve Bulanık Analitik Hiyerarşi Süreci ile Elektrikli Araç Şarj İstasyonu Yer Seçimi, Afyon Kocatepe Üniversitesi Fen Ve Mühendislik Bilimleri Dergisi, 22, 1, 165-174.
- Canbaz, S. ve Erdemir, G., 2021. Açık Kaynak Kodlu Gerçek Zamanlı İşletim Sistemlerinin İncelenmesi. İstanbul Sabahattin Zaim Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 3, 1, 30-37
- Congedo, L., 2013. Semi-Automatic Classification Plugin For QGIS, Sapienza Univ, 125.
- Çabuk, S. N., 2015. CBS'nin yerel yönetimlerde kullanımı ve Kent Bilgi Sistemleri, Harita Teknolojileri Elektronik Dergisi, 7, 3, 69-87.

- Çalışkan, M. ve Anbaroğlu, B., 2020. Geo-MST: A geographical minimum spanning tree plugin for QGIS, *SoftwareX*, 12, 100553.
- Çetinkaya, B. ve Toz, G., 2011. Coğrafi Veri seçim işlemi sonuçlarının değerlendirilmesinde hata matrisinin kullanımı, *İTÜDERGİSİ/d*, 6, 5-6.
- Duarte, L., Silva, P. ve Teodoro, A. C., 2018. Development of a QGIS plugin to obtain parameters and elements of plantation trees and vineyards with aerial photographs, *ISPRS International Journal of Geo-Information*, 7, 3, 109.
- Ellsäßer, F., Röhl, A., Stiegler, C. ve Hölscher, D., 2020. Introducing QWaterModel, a QGIS Plugin for predicting Evapotranspiration from Land Surface Temperatures, *Environmental Modelling & Software*, 130, 104739.
- Erden, T. ve Coşkun, M. Z., 2011a. Coğrafi Bilgi Sistemleri ve Analitik Hiyerarşi Yöntemi yardımıyla İtfaiye İstasyon Yer Seçimi, *TMMOB Harita ve Kadastro Mühendisleri Odası, 13. Türkiye Harita Bilimsel ve Teknik Kurultayı. 18-22 Nisan 2011, Ankara.*
- Erden, T. ve Coşkun, M. Z., 2011b. Acil Durum Servislerinin Yer Seçimi Analitik Hiyerarşi Yöntemi ve CBS Entegrasyonu, *İTÜDERGİSİ/d*, 9, 6.
- Göçgün, Ö. F. ve Aytuğ, O., 2021. Amazon ürün değerlendirmeleri üzerinde Derin Öğrenme/Makine Öğrenmesi Tabanlı duygu analizi yapılması, *Avrupa Bilim ve Teknoloji Dergisi*, 24, 445-448.
- Güler, D. ve Yomralıoğlu, T., 2019. Açık Kaynak Kodlu CBS Yazılımı ve Bulanık Analitik Hiyerarşi Yöntemini içeren Elektrikli Araç Şarj İstasyonu yer seçimi önerisi, *Harita Dergisi*, 86, 163, 22-33.
- Gündüz, H., 2019. Açık Kaynak Kodlu Yazılımların Kamu Kurumlarında Kullanılabilirliğinin Araştırılması, Yüksek Lisans Tezi, Atatürk Üniversitesi, Sosyal Bilimler Enstitüsü, Erzurum.
- Güney, C. ve Çelik, R. N., 2004. Bilişim Projesi, İletişim ve Paylaşım Teknolojisi Olarak CBS'nin "Y" si ve Geomatik Mühendisliğinin Yeri, *Jeodezi ve Jeoinformasyon Dergisi*, 91, 46-54.
- Hanbay-Tiryaki, S. ve Balaman, F., 2021. Açık kaynak kodlu yazılımlardan scratch, arduino ve python kullanımı hakkında öğrenci görüşleri, *Journal of Computer and Education Research*, 9, 18, 831-852.
- Jansen, B. J., 1998. The Graphical User Interface, *ACM SIGCHI Bulletin*, 30, 2, 22-26.
- Jung, M., 2013. LecoS-A QGIS Plugin For Automated Landscape Ecology Analysis (No. e116v2), *PeerJ PrePrints*.

- Kalkan, S., 2019. Açık Kaynak Kodlu CBS Yazılımlarının Gayrimenkul Yönetim Sisteminde Kullanımı Pirgis Örneği, Yüksek Lisans Tezi, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Kocaeli.
- Kalkınma Bakanlığı, 2013. Onuncu Kalkınma Planı 2014-2018.
- Kapluhan, E., 2014. Coğrafi Bilgi Sistemleri'nin (CBS) coğrafya öğretiminde kullanımının önemi ve gerekliliği, Marmara Coğrafya Dergisi, 29, 34-59.
- Karataş, İ. ve Kırbaş, İ., 2015. Özgür ve Açık Kaynak Kod Coğrafi Bilgi Sistemi yazılımlarının karşılaştırmalı değerlendirmesi, Akademik Bilişim Konferansı, Ocak 2015.
- Kepoğlu, O. V. ve Düzgün, H. Ş., 2008. Özgür CBS yazılımı geliştirme: Quantum GIS Örneği, 2. Uzaktan Algılama ve Coğrafi Bilgi Sistemleri Sempozyumu, UZAL-CBS 2008, Kayseri.
- Khan, S. ve Mohiuddin, K., 2018. Evaluating the parameters of ArcGIS and QGIS for GIS Applications, International Journal of Advance Research in Science and Engineering, 7, 582-594.
- Khoirom, S., Sonia, M., Laikhuram, B., Laishram, J. ve Singh, T. D., 2020. Comparative analysis of Python and Java for beginners. International Research Journal of Engineering and Technology (IRJET), 7, 8, 4384-4407.
- Köktürk, E., 2003. Coğrafi Bilgi Sistemi (CBS) Ne Değildir, TUJK Bilimsel Toplantısı, Coğrafi Bilgi Sistemleri ve Jeodezik Ağlar Çalıştayı, Selçuk Üniversitesi, Konya, 24-25-26 Eylül 2003, 55-68.
- Kucher, V., 2016. The development of a cross platform by using Qt Framework, Uluslararası Bilimsel ve Pratik Konferansı, NM Andriychuk, 76-78.
- Kuka, S. ve Bushati, J., 2014. On the role of GIS technology in Geospatial Education (QGIS vs ARGIS), UBT International Conference, 64.
- Kuruüzüm, A. ve Atsan, N., 2001. Analitik Hiyerarşi Yöntemi ve işletmecilik alanındaki uygulamaları, Akdeniz IIBF dergisi, 1, 1, 83-105.
- Malczewski, J., 1999. GIS and multicriteria decision analysis. John Wiley & Sons, New York.
- Mileu, N. ve Queirós, M., 2018. Development of a QGIS plugin to dasymetric mapping. In Free and open source software for geospatial (FOSS4G) conference proceedings, 18, 1, 9.
- Nielsen, A., Hu, F. R. S., Schnedler-Meyer, N. A., Bolding, K., Andersen, T. K. ve Trolle, D., 2021. Introducing QWET–A QGIS-Plugin For Application, Evaluation And Experimentation With The WET Model: Environmental Modelling and Software. Environmental Modelling and Software, 135, 104886.

- Onyıl, H. İ. ve Ulukavak, M., 2021. Açık Kaynak Kodlu Yazılımlarla Alternatif Web Tabanlı Mobil Mekansal Veri Toplama Arayüzü Geliştirilmesi, Harran Üniversitesi Mühendislik Dergisi, 6, 2, 111-120.
- Oxoli, D., Zurbarán, M., Shaji, S. ve Muthusamy, A. K., 2016. Hotspot analysis: a first prototype Python plugin enabling exploratory spatial data analysis into QGIS. In Open Source Geospatial Research and Education Symposium (OGRS2016), PeerJ Inc. San Francisco, USA, 1-6.
- Ölmez, H. S. ve Dai, N., 2016. Özgür Açık Kaynak yazılım stratejilerinin Türkiye bilgi toplumu hedefleri içerisindeki önemi, Working Paper / Technical Report, Sabanci University.
- Önder, Y. E. ve Kavzoğlu, T., 2019. İstanbul ilinde trafik kaza ve yoğunluk analizlerinin Açık Kaynak Kodlu CBS yazılımları ile yapılması, Yüksek Lisans Tezi, Gebze Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Gebze.
- Rischpater, R., 2013. Application development with Qt Creator (7-9), Packt Publishing, Birmingham.
- Saraloğlu, E., Yıldırım, D. ve Güngör, O., 2014. Uzaktan Algılama Araştırmacılarına Yönelik Python Ara Yüzü ve ArcGIS yazılımı eklentisi, Uzaktan Algılama-CBS Sempozyumu (UZAL-CBS 2014), 14-17 Ekim 2014, İstanbul.
- Saaty, T. L., 2000. Fundamentals of Decision Making and priority theory with the Analytic Hierarchy Process, RWS Publications.
- Stallman, R., 2003. Free Software Foundation (fsf). In Encyclopedia of Computer Science, 732-733.
- Stępnia, C. ve Turek, T., 2020. Possibilities of using GIS technology for dynamic planning of investment processes in cities. Procedia Computer Science, 176, 3225-3234.
- Turek, T. ve Stępnia, C., 2021. Areas of Integration of GIS Technology and Smart City Tools. Research findings, Procedia Computer Science, 192, 4681-4690.
- Şahinsoy, A. K., 2011. Açık Kaynak Kodlu CBS yazılımlarının imar planı verilerinin yönetilmesinde kullanımı İstanbul Örneği, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, Geomatik Mühendisliği Anabilim Dalı, İstanbul.
- Şen, B., Atasoy, F. ve Aydın, N., 2010. Düşük maliyetli Web Tabanlı uzaktan eğitim sistemi uygulaması, Akademik Bilişim, 10, 12.
- Taşyürek, M., 2021. Mekânsal verilerin sıklıkla güncellendiği coğrafi bilgi sistemleri arama işleminde denormalizasyon yöntemi, Avrupa Bilim ve Teknoloji Dergisi, 24, 18-23.

- Uluğtekin, N. ve Doğru, A. Ö., 2005. Coğrafi bilgi sistemi ve harita: kartografya, Ege Üniversitesi CBS Sempozyumu, 27-29 Nisan.
- Ünaldık, S. B., 2019. Çok Kriterli Karar Verme Yöntemi ile Yer Seçimi Kararı Üretimi'nde Coğrafi Bilgi Sistemleri'nin kullanımı, Yapı Bilgi Modelleme, 1, 2, 2687-4660.
- Vankova, L., Krejzaa, Z., Kocourkova, G. ve Laciga, J., 2022. Geographic Information System usage options in Facility Management, Procedia Computer Science, 196, 708-716.
- Vitalis, S., Arroyo Ogori, K. ve Stoter, J., 2020. CityJSON in QGIS: Development of An Open- Source Plugin, Transactions in GIS, 24, 5, 1147-1164.
- Yalçın, C. ve Sabah, L., 2017. Açık Kaynak Kodlu CBS ve Analitik Hiyerarşi Süreci (AHP) yöntemi kullanılarak Edirne sanayi işletmelerinin deprem tehlike analizi, Düzce Üniversitesi Bilim ve Teknoloji Dergisi, 5, 2, 524-537.
- Yılmaz, N., 2017. Açık Kaynak Yazılımlarda bakım yapılabilirliği ve güvenilirliği ölçmek için iki boyutlu değerlendirme metodu, Yüksek Lisans Tezi, Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.
- Yılmaz, N. ve Tarhan, A., 2019. Açık Kaynak Yazılımlarda Bakım Yapılabilirliği Ve Güvenilirliği Ölçmek İçin İki Boyutlu Değerlendirme Metodu, Journal of the Faculty of Engineering and Architecture of Gazi University, 34, 4, 1807-1829.
- Yılmaz, O. Y., 2006. Coğrafi Bilgi Teknolojilerinde Özgür Yazılım, HKM Jeodezi, Jeoinformasyon ve Arazi Yönetimi Dergisi, 2006/2, 95.
- Yılmaz, O. Y., 2012. Sanal doku üzerinden coğrafi bilgi sistemi yazılımları komutlarını çalıştırma, IV. Uzaktan Algılama ve Coğrafi Bilgi Sistemleri Sempozyumu (UZAL-CBS 2012), 16-19 Ekim 2012, Zonguldak.
- Yılmaz, T., 2019. Raspberry Pi Kartı Kullanarak Nesne Tespit ve Takip Robotunun Tasarımı ve Modellenmesi, Yüksek Lisans Tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elazığ.
- Yomralıoğlu, T., 2000. Coğrafi Bilgi Sistemleri Temel Kavramlar ve Uygulamaları Kitabı, Akademi Kitabevi, Trabzon.
- Zare, A., Hoboubib, N., Farahbakhsh, S. ve Jahangiri, M., 2022. Applying Analytic Hierarchy Process and Failure Likelihood Index Method (AHP-FLIM) to assess human reliability in critical and sensitive jobs of a petrochemical industry, Heliyon, 8, 5. e09509.
- Zeydan, M., Bostancı, B., Oralhan, B., Eroğlu, D. ve Aydın, U., 2020. Mekânsal Bulanık Karar Destek Sisteminin Geliştirilmesi. Avrupa Bilim ve Teknoloji Dergisi, Özel Sayı, 418-429.

URL-1 <<https://qgis.org/tr/site/about/index.html>>, Erişim tarihi: 24.01.2022

URL-2 <<https://www.ibm.com/tr-tr/topics/open-source>>, Erişim tarihi: 24.01.2022

URL-3 <<https://www.python.org/downloads>>, Erişim tarihi: 24.01.2022

URL-4 <https://wiki.qt.io/Qt_History>, Erişim tarihi: 21.03.2022

URL-5 <https://doc.qt.io/qt-5/qtgui_index.html>, Erişim tarihi: 21.03.2022

URL-6 <https://docs.qgis.org/3.10/tr/docs/pyqgis_developer_cookbook>, Erişim tarihi: 21.03.2022

URL-7 <<https://gdal.org/>>, Erişim tarihi: 21.03.2022



ÖZGEÇMİŞ

Adı ve Soyadı : Emre YILMAZ

EĞİTİM BİLGİLERİ (Kurum ve Yıl)

Lisans : İstanbul Teknik Üniversitesi, Geomatik Müh. 2008-2012

Yüksek Lisans : Aksaray Üniversitesi, Harita Müh. ABD, 2020-2022

TEZDEN ÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER

Ulusal Hakemli Dergilerde Yayımlanan Makaleler

1. XI. TUFUAB 2022 Türkiye Ulusal Fotogrametri ve Uzaktan Algılama Birliği Teknik Sempozyumu, Çok Ölçütlü Karar Verme Analizleri İçin Açık Kaynak Kodlu Eklenti Geliştirilmesi Sunumu, 13 Mayıs 2022