

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

**DÖKÜMANLAR VE PROGRAM KODLARI ARASI  
BENZERLİK ANALİZİ VE UYGULAMASI**

**YÜKSEK LİSANS TEZİ**

**Uğur ŞİMŞEK**

**Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM MÜH.**

**Tez Danışmanı : Yrd. Doç. Dr. Murat İSKEFİYELİ**

**Aralık 2013**

T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

**DÖKÜMANLAR VE PROGRAM KODLARI ARASI  
BENZERLİK ANALİZİ VE UYGULAMASI**

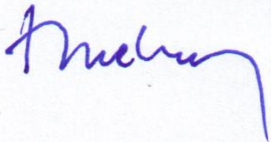
**YÜKSEK LİSANS TEZİ**

**Uğur ŞİMŞEK**

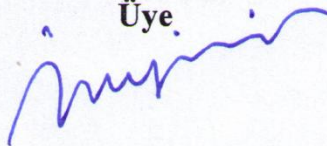
**Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM MÜH.**

Bu tez 27/12/2013 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.

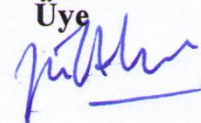
**Doç. Dr.  
İbrahim ÖZÇELİK  
Jüri Başkanı**



**Yrd. Doç. Dr.  
Murat İSKEFİYELİ  
Üye**



**Yrd. Doç. Dr.  
Gültekin ÇAĞIL  
Üye**



## ÖNSÖZ

Araştırma süresince, çalışmalarımda yardımlarını ve desteğini esirgemeyen değerli danışmanım Sayın Yrd. Doç. Dr. Murat İSKEFİYELİ'ye teşekkürlerimi sunmayı bir borç bilirim.

Bölümümüzdeki değerli hocalarıma, maddi ve manevi yardımlarını esirgemeyen, hayat tecrübeleriyle beni yönlendiren sevgili aileme teşekkürlerimi sunarım.

# İÇİNDEKİLER

ÖNSÖZ.....	ii
İÇİNDEKİLER.....	iii
SİMGELER VE KISALTMALAR LİSTESİ .....	vi
ÖZET .....	vii
ABSTRACT .....	vii
ŞEKİLLER LİSTESİ.....	ix
BÖLÜM 1. GİRİŞ.....	1
BÖLÜM 2. KAYNAK KOD İNTİHALI .....	3
2.1. İntihale Neden Olan Unsurlar .....	3
2.2. Kaynak Kod Klonlama Yöntemleri ve Tespit Araçları .....	4
2.2.1. Moss.....	5
2.2.2. JPlag.....	5
2.2.3. CCFinder .....	6
2.2.4. Sherlock.....	6
2.2.5. Sim .....	7
2.2.6. Simian.....	7
2.2.7. Cpd .....	8
2.2.8. Duplo .....	8
2.2.9. Plaggie .....	8
2.3. Sonuç.....	9
BÖLÜM 3. UYGULAMANIN AVANTAJLARI VE İŞLEVSEL ÖZELLİKLERİ..	10
3.1. Uygulamanın Avantajları .....	10
3.2. Uygulamanın Akışı .....	11

3.2.1. Uygulamanın metin belgeleri için akışı .....	11
3.2.2. Uygulamanın kod dosyaları için akışı .....	12
3.3. Uygulamanın Kullanımı .....	13
3.3.1. Notepad tipindeki metin dosyalarının karşılaştırılması.....	13
3.3.2. Notepad tipindeki metin belgelerinde benzerlik oranlarının hesaplanması .....	25
3.3.3. Word belgesi tipindeki metin belgelerinin karşılaştırılması.....	27
3.3.4. Word belgesi tipindeki metin dosyalarında benzerlik oranlarının hesaplanması.....	36
3.3.5. Kaynak kod dosyalarının karşılaştırılması.....	38
3.3.6. Kaynak kod dosyalarında benzerlik oranlarının hesaplanması.....	48
3.4. Uygulamanın Cpd ile Karşılaştırılması .....	49
3.5. Sonuç.....	51

BÖLÜM 4. UYGULAMADA KULLANILAN VERİTABANI TABLOLARI, PROSEDÜRLER VE KULLANIM AMAÇLARI.....	52
4.1. Uygulamada Kullanılan Tablolar ve Kullanım Amaçları .....	52
4.2. Uygulamada Kullanılan Prosedürler ve Kullanım Amaçları.....	58
4.2.1. Veritabanı Tablolarını Temizleyen Prosedür .....	58
4.2.2. Kullanılan Kelimeleri ve Kullanım Sayısı Bilgilerini Kaydeden Prosedür .....	59
4.2.3. Karşılaştırma İşlemi Yapılan Dosya Bilgilerini Kaydeden Prosedür...	60
4.2.4. Kullanılan Değişken Bilgilerini Kaydeden Prosedür .....	61
4.2.5. Kullanılan Cümleleri ve Kullanım Sayısı Bilgilerini Kaydeden Prosedür	61
4.2.6. Programlama Diline Ait Özel Terimlerin Kullanım Bilgisini Kaydeden Prosedür.....	62
4.2.7. Metin Belgeleri için Katsayı Ağırlık Bilgilerini Kaydeden Prosedür ..	63
4.2.8. Kod Dosyaları için Katsayı Ağırlık Bilgilerini Kaydeden Prosedür ....	64
4.2.9. Metin Belgeleri için Katsayı Ağırlık Bilgilerini Getiren Prosedür.....	66
4.2.10. Kod Dosyaları için Katsayı Ağırlık Bilgilerini Getiren Prosedür .....	66
4.2.11. Boyut ve Yazar Bilgilerini Getiren Prosedür .....	67
4.2.12. Kod Dosyalarına Ait Bilgileri Getiren Prosedür.....	67
4.2.13. Metin Belgelerinde Kullanılan Cümle Bilgilerini Getiren Prosedür..	68

4.2.14. Kod Dosyalarında Kullanılan Değişken Bilgilerini Getiren Prosedür.	68
4.2.15. Metin Belgelerinde Kullanılan Kelime Bilgilerini Getiren Prosedür...	69
4.2.16. Filtreleme Kombosundaki Bilgileri Getiren Prosedür .....	70
4.3. Sonuç.....	70
<b>BÖLÜM 5. SONUÇ VE ÖNERİLER.....</b>	<b>71</b>
<b>KAYNAKLAR.....</b>	<b>73</b>
<b>ÖZGEÇMİŞ.....</b>	<b>75</b>

## **SİMGELER VE KISALTMALAR LİSTESİ**

BO : Benzerlik Oranı  
KDS : Kullanılan Değişken Sayısı

## ÖZET

Anahtar Kelimeler: Kopya Ödev, Ödev Benzerlik Tespiti, Metin Karşılaştırma

Kopya çekmek, öğrenci başarısının yanlış değerlendirilmesine yol açan ve son yıllarda ilköğretimden üniversiteye hatta lisansüstü eğitime kadar her düzeyde sıkça gündemde olan, eğitim ve öğretim sürecinin en önemli sorunları arasında sayılabilecek bir davranıştır. Türkiye’de okuyan nüfusun fazla olması nedeniyle eğitici başına düşen öğrenci sayısı çok fazladır. Bu da eğiticilerin, öğrencileri değerlendirme aşamasında yeterince doğru karar verememeleri sonucunu doğurmaktadır.

Bu çalışmada ASP.NET, C# ve SQL Server 2008 kullanılarak yapılan web tabanlı uygulama ile öğrencilerden metin belgesi veya kod dosyaları şeklinde alınan ödevlerin birbirleriyle karşılaştırılarak benzer kısımların bulunması ve ardından ödevler arasında kullanıcının belirlediği ağırlıklara göre benzerlik oranı hesaplayarak eğiticilere değerlendirme aşamasında yardımcı olunması amaçlanmıştır.



## **ABSTRACT**

Key words: Copy Homework, Homework Similarity Detection, Text Comparison

Copy, cause to a misinterpretation of the student's success and frequently on the agenda at all levels from primary school to university exams even graduate education, is a behavior can be considered among the most important training and education process's problems in recent years. The number of students per trainer is too much because of high number of studying population in Turkey. This cause the results that trainers could not take enough time and could not decide correctly during the evaluation stage.

In this thesis, is purpose to assist trainers in evolution stage by finding similar part and calculating similarity rate between homeworks which are the form of text according to weights are determined by the user with a web-based application developed using ASP.NET, C# and SQL Server 2008.

## ŞEKİLLER LİSTESİ

Şekil 3.1. Metin belgeleri için akış diyagramı .....	12
Şekil 3.2. Kod dosyalar için akış diyagramı .....	13
Şekil 3.3. Uygulama dil seçim ekranı .....	14
Şekil 3.4. Dosya tür seçim ekranı (Türkçe).....	14
Şekil 3.5. Dosya tür seçim ekranı (İngilizce).....	15
Şekil 3.6. Klasör yolu seçim ekranı .....	16
Şekil 3.7. Katsayı seçim ekranı .....	16
Şekil 3.8. Klasör yolu seçim penceresi .....	17
Şekil 3.9. Klasör yolu onay ekranı .....	18
Şekil 3.10. Sonuç gösterim ekranı.....	19
Şekil 3.11. Sonuçların filtrelenmesi .....	20
Şekil 3.12. Filtrelenmiş sonuç gösterim ekranı .....	21
Şekil 3.13. Detay gözlem ekranı .....	23
Şekil 3.14. Kayıt sayfalama yapısı .....	25
Şekil 3.15. Word tipindeki metin belgeleri için katsayı seçim ekranı .....	27
Şekil 3.16. Word tipindeki metin belgeleri için sonuç gösterim ekranı.....	30
Şekil 3.17. Sonuçların filtrelenmesi(Word) .....	31
Şekil 3.18. Filtrelenmiş sonuç gösterim ekranı(Word) .....	32
Şekil 3.19. Word belgesi detay gözlem ekranı.....	34
Şekil 3.20. Kayıt sayfalama yapısı(Word) .....	36
Şekil 3.21. Kod dosyaları için katsayı seçim ekranı .....	38
Şekil 3.22. Klasör yolu onay ekranı(Kod dosyası) .....	39
Şekil 3.23. Kod dosyaları için sonuç gösterim ekranı.....	40
Şekil 3.24. Sonuçların filtrelenmesi(Kod dosyası) .....	41
Şekil 3.25. Filtrelenmiş sonuç gösterim ekranı(Kod dosyası) .....	43
Şekil 3.26. Kod dosyası detay gözlem ekranı .....	45

Şekil 3.27. Kayıt sayfalama yapısı(Kod dosyası) .....	47
Şekil 3.28. Kayıt filtreleme(Kod dosyası).....	47
Şekil 3.29. Cpd karşılaştırma ekranı .....	50
Şekil 3.30. Karşılaştırılan klasörler .....	51
Şekil 4.1. Değişken bilgilerinin tutulduğu tablo .....	52
Şekil 4.2. Değişken bilgilerinin tutulduğu tablodan kesit .....	53
Şekil 4.3. Değişken tipleri ve kullanım sayılarının tutulduğu tablo.....	53
Şekil 4.4. Değişken tipleri ve kullanım sayılarının tutulduğu tablodan kesit .....	54
Şekil 4.5. Metin belgeleri için katsayıların tutulduğu tablo .....	54
Şekil 4.6. Metin belgeleri için katsayıların tutulduğu tablodan kesit.....	55
Şekil 4.7. Kod dosyaları için katsayıların tutulduğu tablo .....	55
Şekil 4.8. Kod dosyaları için katsayıların tutulduğu tablodan kesit.....	55
Şekil 4.9. Klasör içerik bilgisinin tutulduğu tablo .....	56
Şekil 4.10. Klasör içerik bilgisinin tutulduğu tablodan kesit .....	56
Şekil 4.11. Cümle kullanım bilgilerinin tutulduğu tablo.....	57
Şekil 4.12. Cümle kullanım bilgilerinin tutulduğu tablodan kesit .....	57
Şekil 4.13. Kelime kullanım bilgilerinin tutulduğu tablo .....	57
Şekil 4.14. Kelime kullanım bilgilerinin tutulduğu tablodan kesit .....	58
Şekil 4.15. Veritabanı tablolarını temizleyen prosedür .....	59
Şekil 4.16. Kelimeleri ve kullanım sayı bilgilerini kaydeden prosedür .....	60
Şekil 4.17. Karşılaştırma işlemi yapılan dosya bilgilerini kaydeden prosedür .....	60
Şekil 4.18. Değişken bilgilerini kaydeden prosedür .....	61
Şekil 4.19. Cümleleri ve kullanım sayı bilgilerini kaydeden prosedür .....	62
Şekil 4.20. Özel terimleri ve kullanım sayısı bilgilerini kaydeden prosedür .....	62
Şekil 4.21. Metin belgeleri için katsayı ağırlık bilgisini kaydeden prosedür .....	64
Şekil 4.22. Kod dosyaları için katsayı ağırlık bilgisini kaydeden prosedür .....	65
Şekil 4.23. Metin belgeleri için katsayı ağırlık bilgisini getiren prosedür .....	66
Şekil 4.24. Kod dosyaları için katsayı ağırlık bilgisini getiren prosedür .....	66
Şekil 4.25. Boyut ve yazar bilgilerini getiren prosedür .....	67
Şekil 4.26. Kod dosyalarına ait bilgileri getiren prosedür .....	68
Şekil 4.27. Metin belgelerinde kullanılan cümle bilgilerini getiren prosedür.....	68
Şekil 4.28. Kod dosyalarında kullanılan değişken bilgilerini getiren prosedür .....	69
Şekil 4.29. Metin belgelerinde kullanılan kelime bilgilerini getiren prosedür .....	69

Şekil 4.30. Filtreleme Kombosunu Dolduran Prosedür .....	70
--	----

## **BÖLÜM 1. GİRİŞ**

Teknolojinin gelişmesiyle birlikte bilgisayar günlük yaşamın hemen hemen her alanında hayatımıza girmiştir. Özellikle internet günlük hayatımızdaki yerini hızla almıştır. Bu gelişmeler öğrenilen bilgi ve yaşantıların kalıcılığını sağlamak veya bu bilgi ve yaşantıları genişletmek amacıyla eğitimciler tarafından öğrencilere genelde ders dışı zamanlarda hazırlamaları için verilen çalışmalar olan ödevlerin değerlendirilmesi işleminin bilgisayar ortamında, daha hızlı ve güvenli bir şekilde yapılmasını gündeme getirmiştir.

Günlük hayatta yaygın bir kelime olarak kullanılan kopyanın (intihal) sözlük anlamı istediğini dürüst olmayan yollarla elde etme davranışıdır. Akademik anlamda ise kopya, öğrencinin notunu yükseltmek için kanun ve mevzuata uygun olmayan kaynaklardan bilgileri elde etmesi, kullanması ve kendisinin olmayan çalışmaları kendisinin gibi düşünmesidir.

Kaynak kodların benzerlik tespiti diğer edebi türlerdeki intihali tespit etmekten daha zordur. Bunun nedeni programlama dillerinin kendine has gramerinin normal dillerdeki gramerden daha kurallı ve formal oluşudur. Bu özelliğinden dolayı normal dillerdeki intihali tespit eden yazılımların kaynak kodlardaki intihali tespit etmek için kullanımı sağlıklı sonuçlar vermeyecektir.

Kaynak kodlardaki intihali tespit etmek amacıyla geliştirilen araçlar, yazılım mühendisliğinde birden fazla kaynak kod dosyasına kopyalanmış fonksiyonların tek bir fonksiyon altında toplanması amacıyla kullanılmaktadır. Yazılım mühendisliğinde buna ek olarak programı anlama, kaynak kod kalite analizi, yazılım gelişimi analizi, kod sıkıştırma, virüs algılama ve hata algılama gibi alanlarda da kaynak kod karşılaştırılması yapılmaktadır. Hukuksal anlamda ise benzerlik tespit araçları kaynak kodlardaki intihali tespit etmek amacıyla kullanılmaktadır.

Kanun tarafından koruma sađlanan kaynak kodların kopyalanması veya intihal edilmesi durumunda, eser sahibi olan bilgisayar programcısı diđer eser sahiplerine tanınan hakların tamamına sahiptir. İntihalin varlığı veya intihal şüphesi durumunda eser sahibi mahkemeden intihalin tespitini isteyebilir. Mahkemelerin intihali tespit amacıyla görevlendirdiđi bilirkişilerin bu araçları kullanması ile intihalin varlığına ilişkin daha sađlıklı sonuçlar alınabilmektedir.

Kaynak kod intihali sadece yazılım sektöründe deđil akademide de önemli bir sorundur. Öğrencilere verilen programlama ödevlerinin kopyalanarak ve üzerinde az bir deđişiklik yapılarak sunulması bu alanda geliştirilen yazılımların önemini daha da artırmaktadır.

Kaynak kod intihalini tespit eden yazılımların kullanımı sayılan nedenlerden dolayı hem bilirkişiler, hem akademisyenler hem de yazılım sektörü çalışanları için önem taşımaktadır.

Bu çalışmada, klasik yöntemlerle ödev deđerlendirmenin zorlukları göz önüne alınarak web tabanlı örnek bir uygulama geliştirilmiştir. Geliştirilen uygulama ile, metin belgesi ve bilgisayar programının programcı tarafından; düz metin şeklinde bir programlama diliyle yazılmış ve o programlama diline ait komutları barındıran yazılımın bir biçimi olan kaynak kod dosyası şeklinde hazırlanan ödevlerin benzer kısımlarının bulunarak benzerlik oranının hesaplanması, böylece deđerlendirilme işleminin daha kısa zamanda minimum hata ile yapılması amaçlanmıştır.

## **BÖLÜM 2. KAYNAK KOD İNTİHALI**

Kaynak kod intihali, kaynak kod dosyalarının veya kaynak kod kısımlarının asıl eser sahibinin izni olmaksızın kullanılmasıdır.

Edebiyat, sanat, musiki ve diğer güzel sanat dallarında intihal yapılan eserleri fark etmek, kaynak kod intihali yapılan bilgisayar programını fark etmekten daha kolaydır. Çünkü diğer sanat dallarında üretilmiş eserlerde, o konunun uzmanları veya meraklıları, söz konusu eserde intihal olduğunu herhangi bir teknik incelemeye gerek kalmadan fark edebilirler. Oysa bilgisayar programlarının kaynak kodları görünmediğinden teknik bir inceleme söz konusu olmaksızın intihali tespit etmek zordur. Bu gibi durumlarda, kaynak kodlar arasında intihal olduğunun tespiti için birbirine benzeyen kaynak kodların karşılaştırılması gerekmektedir.

### **2.1. İntihale Neden Olan Unsurlar**

İntihale neden olan unsurlar incelendiğinde eğitim eksikliği ilk sırada gelmektedir. Konuyla ilgili yapılan araştırmalar intihali “bilerek” ve “bilmeden” yapılan hatalar olarak göstermekte ve intihal yapma nedenleri içinde eğitim eksikliğinin önemini vurgulamaktadır. Genelde intihal, yapılmaması gereken bir davranış olarak bilinmesine rağmen, yapılması gerekenlerin neler olduğunun tam olarak bilinmemesi hataların devam etmesine neden olmaktadır.

Bu konuya eğitimin farklı aşamalarında gereken önemin verilerek öğrencilerin eğitilmesi, ödevlerin denetimi, kopyaya neden olan unsurların (çok ödev verilmesi, zaman baskısı vb.) dikkate alınarak azaltılması, intihal uygulamalarında denetim ve yaptırımların yerine getirilmesi ileriki yıllarda bireylerin davranışlarının şekillenmesi ve olumlu alışkanlık kazanmaları açısından önemlidir. Ancak intihal sadece eğitim eksikliğinden kaynaklanmamaktadır. Eğitim dışında intihale neden olan çok sayıda

faktör vardır. Alıntıların belli bir standarda dayanarak verilmemesi, yazım hataları, orijinal kaynağa gitmeden alıntıdan alıntı yapmak yapılan hataların bir zincir şeklinde devam etmesine neden olmaktadır. Ayrıca, bireylerin mesleki hırsları, rekabet ortamı, ahlak eksikliği ve kişilik özelliklerinden kaynaklanan sorunlar da intihale neden olabilmektedir. İntihallerin artmasında bir başka etken de internettir. İnternette yer alan bilginin “anonim bilgi” “kamu bilgisi” “orta malı” olarak algılanması; bilgi kaynağının belirtilmemesi, kaynak göstermede görülen eksiklikler ve hatalar intihallerin artmasına neden olmaktadır.

## **2.2. Kaynak Kod Klonlama Yöntemleri ve Tespit Araçları**

Kaynak kod klonlama yöntemlerinden bazıları, isim değişikliği (değişken, fonksiyon isimleri vs), kod bloklarını tekrar sıralama, kodu farklı biçimde ifade etme, kaynak kodu farklı bir programlama diline çevirme, kaynak koda düzmece kod ekleme/silme, karma kaynak kod kullanmadır.

Kaynak kod, normal metinlere göre çok daha kurallı bir yapıya sahiptir. Bu nedenle kaynak kodlar arasında benzerlik karşılaştırması yapmak ve var olan benzerliği tespit etmek normal metinlere göre daha zordur. Bu nedenle benzerlik ölçümüyle ilgili daha güvenilir sonuçlar alabilmek için özel olarak kaynak kodlardaki benzerliği tespit eden araçların kullanılması gerekmektedir [1].

Kaynak kodlar arasında benzerliği tespit etmek için piyasada MOSS, JPlag, Sherlock, SIM, Simian, CPD, Duplo, Plaggie gibi araçlar kullanılmaktadır. Bu araçlardan bazıları satır bazlı karşılaştırma işlemi yaptığı için aynı satırda ifadelerin yerinin değiştirilmesi şeklinde yapılan intihalleri tespit edememekte, bazıları ise sadece benzer kısımları göstermekte fakat benzerlik oranı hesaplamamakta, birçoğu ise çoklu dosya karşılaştırmayı desteklememekte karşılaştırılmak istenen dosyaları sadece ikili olarak karşılaştırmaktadır.

Kaynak kod benzerliğinin tespiti amacıyla çeşitli kaynak kod benzerlik ve klon tespit araçları geliştirilmiştir. Bu araçlardan MOSS ve JPlag gibi intihal tespit yazılımları



benzerlik oranı vermekte, CPD ve Duplo gibi klon kod tespit yazılımları ise sadece kopyalanan satırları göstermektedir [1].

### **2.2.1. Moss**

Web tabanlı kaynak kod karşılaştırması yapan MOSS (Measure of Software Similarity), kaynak kod benzerliği tespitinde kullanılan en popüler araçlardan biridir. MOSS benzerlik tespitini kullanıcının bilgisayarında değil gönderim scripti ile kendi sunucusunda yapmaktadır. MOSS, en fazla dilde karşılaştırma yapan benzerlik tespit araçlarından biridir. MOSS aralarında C, C++, Java, C#, Python, Visual Basic, Javascript, FORTRAN, Python, PL/Sql 'in olduğu 27 programlama dilinde kaynak kod benzerliği tespiti yapabilmektedir. MOSS benzerlik tespit aracı, karşılaştırma sonucu benzerlik oranını HTML sayfası olarak oluşturmakta ve karşılaştırılan kaynak kodları yan yana listeleterek benzerlik olan kısımları farklı renklerle göstermektedir [1, 19].

Moss, bu çalışmada geliştirilen uygulamadan farklı olarak metin belgelerindeki benzerliklere bakmayıp sadece kod dosyalarındaki benzerliklere bakmaktadır ve kod dosyaları için benzerlik oranı hesaplarken görsel kısımlardaki benzerlikleri dikkate almamaktadır. Ayrıca bu program klasör bazlı değil dosya bazlı çalışmaktadır yani çok sayıda kod dosyası içeren projelerin karşılaştırılması işlemini desteklemeyip kod dosyalarını tek tek karşılaştırmaktadır bu da Moss'un geliştirilen uygulama karşısındaki diğer bir dezavantajıdır. Moss kod dosyalarını satır bazlı karşılaştırdığı için aynı satırlarda küçük değişiklikler yapılması sonucunda oluşan intihalleri tespit edememektedir.

### **2.2.2. JPlag**

JPlag, sunucuya gönderilen her bir kaynak kod dosyasını, ait olduğu programlama dilinin gramerini dikkate alarak inceler ve simge dizilerine dönüştürür. Bu simge dizileri de Greedy String Tiling algoritması kullanılarak birbirleriyle karşılaştırılır ve benzerlik ölçümü hesaplanır. JPlag C, C++, Java, C#, Scheme ve doğal dillerde benzerlik tespiti yapabilmektedir. Grafik arayüzden kullanılan JPlag, karşılaştırma

sonuçlarını HTML sayfası olarak sunmaktadır. Karşılaştırılan kaynak kodlar arasındaki benzerlik yüzde olarak verilmekte ve benzerlik bulunan kısımlar MOSS'ta olduğu gibi yan yana listelenmektedir. JPlag, MOSS uygulamasındaki gibi kaynak kod karşılaştırma işlemini Web tabanlı servis şeklinde kendi sunucusunda online olarak yapmaktadır. Karşılaştırılacak kaynak kodlar JPlag'ın sunucusuna gönderilmekte, karşılaştırma sonuçları ise kullanıcının bilgisayarına gelmektedir [1, 20].

JPlag, MOSS uygulamasındaki gibi metin belgelerindeki benzerliklere bakmayıp sadece kod dosyalarındaki benzerliklere bakmaktadır ve kod dosyaları için benzerlik oranı hesaplarken görsel kısımlardaki benzerlikleri dikkate almamaktadır, klasör bazlı değil dosya bazlı çalışmaktadır yani çok sayıda kod dosyası içeren projelerin karşılaştırılması işlemini desteklemeyip kod dosyalarını tek tek karşılaştırmaktadır.

### **2.2.3. CCFinder**

Büyük boyutlu yazılım sistemlerindeki kaynak kod benzerliğini tespit edebilen CCFinder (Code Clone Finder), karşılaştırılacak kaynak kodu simgelere çevirir ve karşılaştırmayı bu simgeler üzerinden yapar. Java, C/C++, C#, Visual Basic (VB) ve COBOL dillerinde karşılaştırma yapabilen CCFinder, farklı kod satırlarındaki farklı isimlendirmelere sahip klonları bulabilmektedir. CCFinder benzerlik sonuçlarını çizim üzerinde noktalama yöntemi (dotplotting) kullanarak görselleştirir [1, 21].

CCFinder, sadece kaynak kod dosyaları arasında karşılaştırma işlemi yapmaktadır. Benzerlik sonuçlarını şekil üzerinde noktalama yöntemi kullanarak gösterdiğinden karşılaştırılan dosyaların benzerlik oranları ve benzer olan kısımlar hakkında net bilgi verememektedir.

### **2.2.4. Sherlock**

Sherlock, kaynak kodlarda simgeleştirme yöntemini kullanarak metinlerde ise string eşleme yöntemini kullanarak benzerlik tespiti yapan açık kaynak kodlu bir araçtır. Sherlock C, C++ Java, Pascal, HTML ve metin dosyaları arasındaki benzerlik

oranlarını vermektedir. Sherlock, kaynak kodda kelimelerin dizilimine bakarak hash algoritması ile dijital sayı dizileri üretmekte ve bunları karşılaştırmaktadır [1, 22].

Sherlock, geliştirdiğimiz uygulamaya benzer olarak hem metin dosyalarını hem de kaynak kod dosyalarını karşılaştırabilmektedir. Fakat yapılan karşılaştırma işlemi sonucunda dosyalar arası benzerlik oranları verilmiş benzer olan kısımlar gösterilmemiştir.

### **2.2.5. Sim**

Sim C, Java, Pascal, Modula-2, Lisp, Miranda ve metin dosyalarında benzerlik tespiti yapan bir araçtır. SIM isim değişikliklerini ve program bloklarının yer değişimini fark edebilmekte, boşlukları ise dikkate almamaktadır [1, 23].

Sim, Sherlock gibi metin dosyalarını ve kaynak kod dosyalarını karşılaştırabilmekte, benzerlik oranlarını hesaplamakta fakat benzer olan kısımları göstermemektedir. Ayrıca kod dosyalarını karşılaştırırken görsel kısımları dikkate almamaktadır.

### **2.2.6. Simian**

Java, C, Objective-C, C++, C#, JavaScript (ECMAScript), COBOL, ABAP, Ruby, Lisp, SQL, Visual Basic, Groovy dillerinde tam olarak JSP, ASP, HTML, XML dillerinde de kısmi olarak karşılaştırma yapan Simian (Similarity Analyser) metin tabanlı karşılaştırma yapan kaynak kod benzerlik tespit aracıdır.

Simian, karşılaştırılan kodların benzerlik oranlarını vermemekte, sadece karşılaştırılan kaynak kod dosyalarında birbirinden kopyalanmış satırları göstermektedir. Simian sadece dosya bazlı değil klasör tabanlı karşılaştırma da yapabilmektedir [1, 24].

Simian, bu çalışmadaki uygulamaya benzer olarak klasör tabanlı karşılaştırma da yapabilmektedir. Fakat benzerlik oranlarını hesaplamayıp sadece kopyalanan satırları göstermektedir.

### 2.2.7. Cpd

Cpd (Copy/Paste Dedector), PMD projesine bütünleşik simge tabanlı klon kod tespiti yapabilen ve son versiyonu “Karp-Rabin” metin eşleme algoritmasını kullanan açık kaynak bir yazılımdır. CPD C, C++, C#, Java, JSP, Ruby, Fortran, PHP ve seçilen uzantıya sahip kaynak kod dosyalarında karşılaştırma yapabilmektedir. Grafik arayüz ve konsoldan kullanılabilen CPD, kaç satırın klonlandığını ve bunların hangi aralıkta olduğu bilgisini vermekte; benzerlik ölçümü yapmamakta sadece klon kod tespiti yapmaktadır. BSD lisansı ile yayımlanan CPD aralarında Eclipse, NetBeans, JDeveloper, JEdit, BlueJ ve IntelliJ IDEA gibi birçok IDE’ye eklenti olarak da kurulabilmektedir [1, 25].

Bu uygulama, karşılaştırılan dosyalar arasındaki benzerlik oranı vermemekte sadece kopyalanan satır sayısı bilgisini vermektedir.

### 2.2.8. Duplo

Duplo, kaynak kod veya program olarak indirilebilen ve açık kaynak bir yazılımdır. C, C++, C#, Java ve VB.NET dillerinde kopyalanmış kod bloklarını tespit etmek için kullanılmaktadır. Duplo, kaynak kodlarda benzerlik ölçümü vermemekte sadece kodlarda tekrarlanan kod bloklarını tespit etmektedir. Duplo karşılaştırılacak dosyaların adını bir dosyadan okumakta ve yine karşılaştırma sonuçlarını bir dosyaya yazmaktadır [1, 26].

Bu uygulama, çoğu uygulamada olduğu gibi benzerlik oranı vermemekte, karşılaştırma işlemini kod blokları arasında yapmakta ve dosya bazlı çalışmaktadır. Bu yüzden yapılan küçük değişiklikleri algılamamaktadır.

### 2.2.9. Plaggie

Sadece Java kodlarında intihal tespiti yapabilen Plaggie GPL lisansına sahip açık kaynak kodlu benzerlik tespit yazılımıdır. JPlag temel alınarak geliştirilen Plaggie, Running-Karp-Rabin Greedy-String-Tiling (RKR-GST) algoritmasını kullanarak

simgeleştirme tekniğine dayalı karşılaştırma yapmaktadır. Plaggie, karşılaştırma işlemini kurulduğu bilgisayarda yapmakta ve benzerlik sonuçlarını HTML sayfası şeklinde kullanıcıya sunmaktadır [1, 27].

Plaggie, metin belgeleri arasında karşılaştırma işlemi gerçekleştirmemekte ayrıca kod dosyalarının da görsel kısımlarındaki benzerlikleri dikkate almamaktadır.

### **2.3. Sonuç**

Metin dosyalarından ve kaynak kodlarından oluşan ödevlerin değerlendirilmesi sürecinde, yapılan intihallerin tespit edilememesi ödev değerlendirme işleminin hatalı yapılmasına neden olmaktadır. Bu yüzden piyasada, karşılaştırma işlemi yapan çok sayıda program ve algoritma mevcuttur. Fakat bu zamana kadar yapılan çalışmaların hiç biri intihal tespiti yaparken programlama ödevlerinin görsel kısımlarındaki benzerlikleri dikkate almamaktadır. Ayrıca bu programların çoğu dosya bazlı karşılaştırma işlemi yaptığı için içerisinde çok sayıda kod dosyası içeren web tabanlı programlama ödevlerinin karşılaştırılması işlemini desteklememektedir. Yine bu programlardan çoğu, satır bazlı karşılaştırma işlemi yaptığından dolayı aynı satırlarda küçük değişiklikler yapılması sonucu oluşan intihalleri tespit edememektedir.

Bu bölümde, yukarıda bahsedilenlerin ışığı altında piyasa bulunan bazı karşılaştırma programları ile geliştirmiş olduğumuz uygulamanın farklılıkları ortaya konulmuştur.

## **BÖLÜM 3. UYGULAMANIN AVANTAJLARI VE İŞLEVSEL ÖZELLİKLERİ**

Bu bölümde, ASP.NET, C# ve SQL Server 2008 kullanılarak hazırlanan web tabanlı uygulamanın avantajlarından ve işlevsel özelliklerinden bahsedilecektir.

ASP.NET, Microsoft tarafından geliştirilmiş bir web uygulama gelişimi teknolojisidir. Dinamik web sayfaları, web uygulamaları ve XML tabanlı web hizmetleri yaratılmasına olanak sağlar. Geliştirilen uygulamada ASP.NET 4.0 versiyonu kullanılmıştır.

C# programlama dili , Microsoft'un .NET teknolojisi için geliştirmiş olduğu yeni nesil programlama dilidir.

SQL Server bir ilişkisel veritabanı yönetim sistemidir. SQL Server ile yoğun verileri işleyebilir, saklayıp analiz edebilir ve yeni uygulamalar geliştirilebilmektedir. SQL Server OLTP ve OLAP için gerekli olan veri saklama ürünlerini ve teknolojilerini destekler. Geliştirilen uygulamada SQL Server 2008 kullanılmıştır.

### **3.1. Uygulamanın Avantajları**

Uygulamayı kullanarak yapılan ödev değerlendirme işleminin klasik değerlendirmeye göre pek çok avantajları mevcuttur. Bunlar aşağıdaki gibi sıralanabilir:

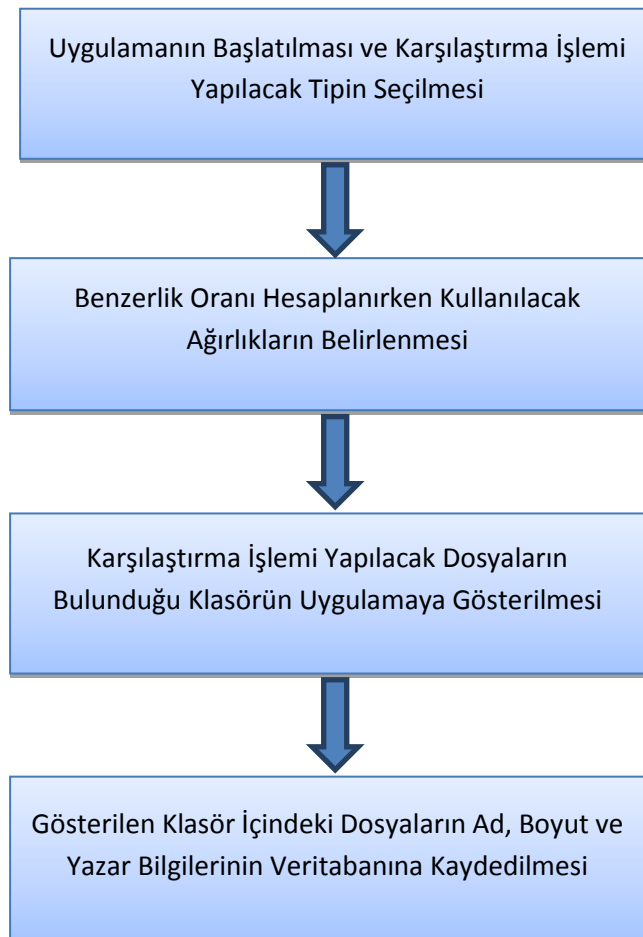
- Kopya ödevlerin kod karmaşasına girilmeden kolaylıkla bulunması sağlanır.
- Ödev karşılaştırma işlemi kullanışlı arayüz sayesinde kolayca ve minimum hata ile yapılır.
- Ödevlerde ortak olan kısımları bulmak için harcanan süre ortadan kaldırılır.

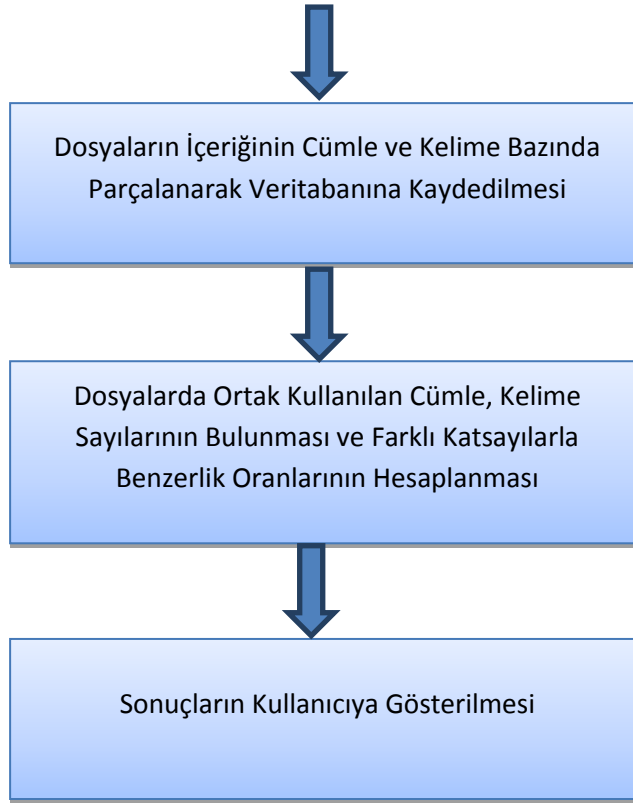
- Ödev değerlendirmede standardı yakalamaya ve değerlendirme kalitesinin arttırılmasına yardımcı olur.
- Belgeleme, dosyalama işlemlerini hızlandırır. Belge ve bilgiye kolay ulaşılmasını sağlar.
- Değerlendirme esnasında harcanan zamandan tasarruf sağlar.
- Yapılan çalışmaları depolama imkanı verir, depolanan bilgilere ve dokümanlara anında ulaşma imkanı sağlar.
- Değerlendirme işleminin objektif yapılmasını sağlar.

### 3.2. Uygulamanın Akışı

Akış diyagramı metin belgeleri ve kod dosyaları için ayrı ayrı oluşturulmuştur. Şekil 3.1 metin belgeleri için akış diyagramını, Şekil 3.2 ise kod dosyaları için akış diyagramını içermektedir.

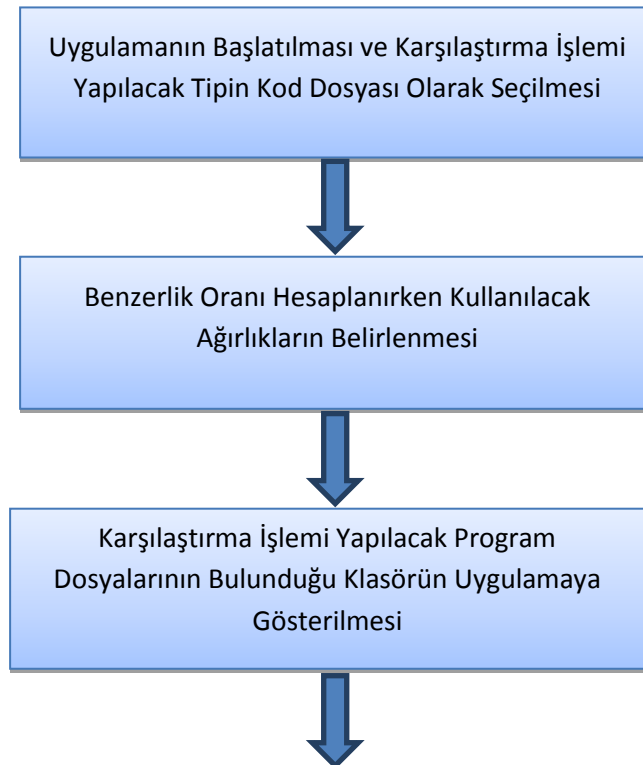
#### 3.2.1. Uygulamanın metin belgeleri için akışı



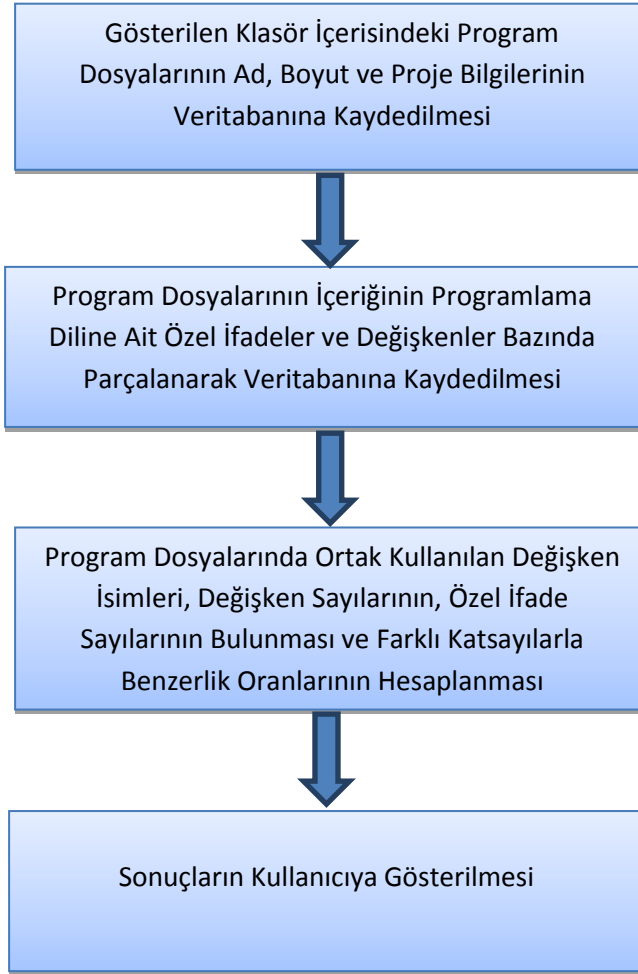


Şekil 3.1. Metin belgeleri için akış diyagramı

### 3.2.2. Uygulamanın kod dosyaları için akışı







Şekil 3.2. Kod dosyaları için akış diyagramı

### 3.3. Uygulamanın Kullanımı

Geliştirmiş olduğumuz uygulamada notepad, word tipindeki metin belgeleri ve C# ile yazılmış web projelerinin incelenmesi işlemi yapılmaktadır.

#### 3.3.1. Notepad tipindeki metin dosyalarının karşılaştırılması

Geliştirilen uygulamada kullanıcıya Türkçe ve İngilizce olmak üzere iki farklı dil desteği sunulmaktadır. Uygulama çalıştırıldığında karşımıza Şekil 3.3'teki dil seçim ekranı çıkmaktadır. Bu ekran vasıtası ile kullanıcı uygulamayı hangi dil desteği ile kullanmak istediğini seçebilir. Yapılan seçimin ardından uygulamanın içeriği ve menüler kullanıcıya seçilen dilde sunulur.



Şekil 3.3. Uygulama dil seçim ekranı

Şekil 3.3'teki dil seçim ekranından uygulama dilinin Türkçe seçilmesi halinde Şekil 3.4'teki dosya tür seçim ekranı açılmaktadır.



Şekil 3.4. Dosya tür seçim ekranı(Türkçe)

Uygulama dilinin İngilizce seçilmesi durumunda ise kullanıcının karşısına Şekil 3.5'deki ekran çıkar. Şekil 3.4 ve Şekil 3.5'deki ekranlar işlevsellik bakımından birbirinin aynısıdır, iki ekran arasındaki tek fark ekranlarda kullanılan metinlerin ve sunulan içeriğin dilinin farklı olmasıdır.

*Please choose file type you want to compare...*

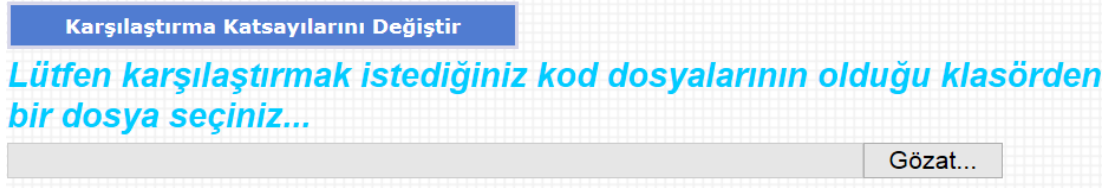


Şekil 3.5. Dosya tür seçim ekranı (İngilizce)

Geliştirilen uygulamanın proje dosyasında, anlatımda gösterilen ekranların her biri için Türkçe ve İngilizce olmak üzere iki farklı versiyonu mevcuttur, kullanıcının yapmış olduğu dil seçimine göre hangi ekranın gösterileceğine karar verilmektedir. Bundan sonraki uygulama içeriği ve menüler, kullanıcının dil seçimini Türkçe olarak yaptığı kabul edilerek anlatılacaktır.

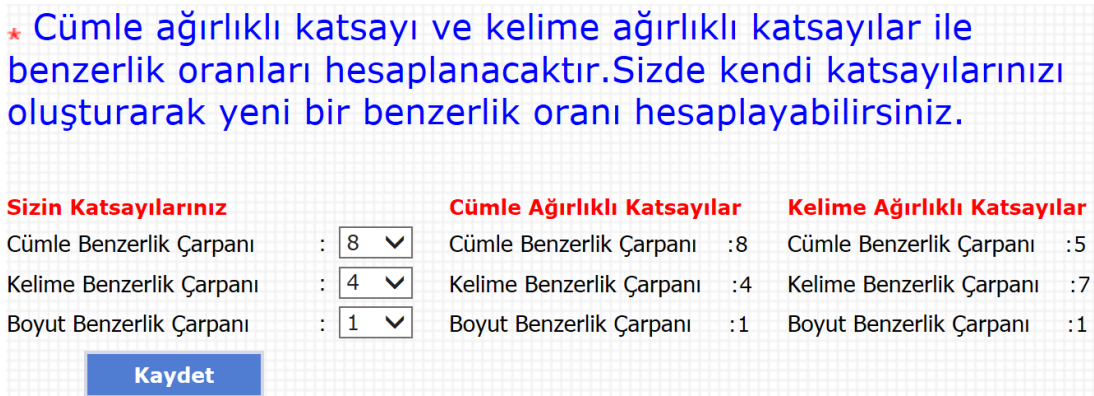
Dil seçim ekranından seçim yapıldıktan sonra Şekil 3.5'deki dosya tür seçim ekranından karşılaştırma işlemi yapmak istenilen dosya türü seçilir. Bu kısımda karşılaştırma işlemi yapılacak dosya türümüzün txt uzantılı dosyalar olduğu varsayılarak ekrandan seçimin "Notepad" olarak yapıldığını düşünelim.

Dosya tür seçim ekranında karşılaştırma işlemi yapılacak dosya türü "Notepad" olarak seçildikten sonra Şekil 3.6'daki klasör yolu seçim ekranı açılmaktadır. Bu ekrandan "Karşılaştırma Katsayılarını Değiştir" butonuna tıklanması halinde benzerlik oranları hesaplanırken kullanılan katsayıların ağırlıklarının değiştirileceği katsayı seçim ekranına gidilmekte, "Gözet" butonuna tıklanması halinde ise açılan ekran vasıtası ile uygulamaya karşılaştırma işlemi yapılacak olan metin belgelerinin yer aldığı klasörün yolu gösterilmektedir.



Şekil 3.6. Klasör yolu seçim ekranı

Yukarıdaki klasör yolu seçim ekranından “Karşılaştırma Katsayılarını Değiştir” butonuna tıklanması halinde Şekil 3.7’deki katsayı seçim ekranı açılır.



Şekil 3.7. Katsayı seçim ekranı

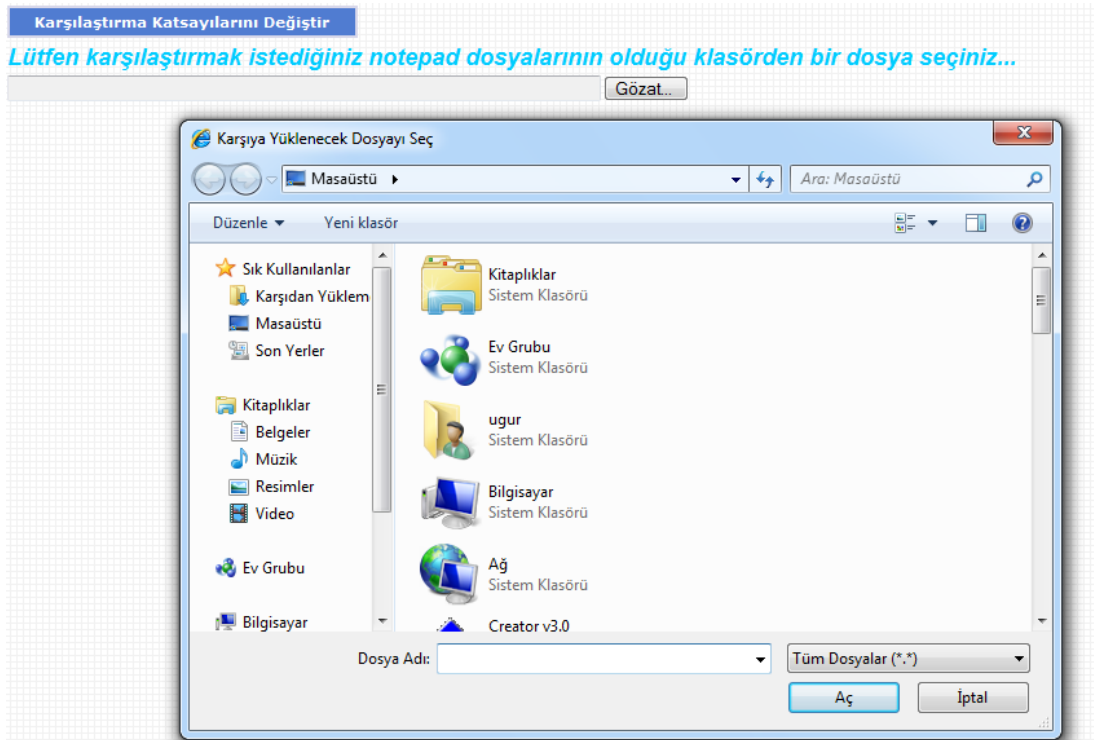
Metin belgesi tipindeki dosyalar için benzerlik oranları hesaplanırken; cümle benzerlik oranı, kelime benzerlik oranı ve boyut benzerlik oranı olmak üzere üç farklı benzerlik oranı kullanılmaktadır.

Metin belgeleri için kullanıcının kendisinin belirlemiş olduğu katsayılar, cümle ağırlıklı katsayılar ve kelime ağırlıklı katsayılar olmak üzere üç farklı katsayı grubuna göre üç tane benzerlik oranı hesaplanmaktadır. Uygulamada cümle ağırlıklı katsayılar ve kelime ağırlıklı katsayılar adı altındaki katsayılar kullanıcıya sabit olarak sunulmaktadır. Ekranın sol kısmında yer alan “Sizin Katsayılarınız” alanında yer alan seçim kutucukları vasıtası ile kullanıcıdan kendi katsayı ağırlıklarını belirlemesi istenmektedir. Böylece farklı ağırlıklardan oluşan üç çeşit katsayı grubuna göre farklı benzerlik oranları hesaplanmış olmaktadır. Farklı ağırlıklarla hesaplanan bu benzerlik oranları sayesinde kullanıcının değerlendirme aşamasında daha sağlıklı karar verebilmesine yardımcı olunur.

Cümle ağırlıklı katsayılar kısmında cümle benzerlik çarpanına daha fazla ağırlık verildiği için karşılaştırılan metin belgelerinde ortak olarak kullanılan cümlelerin fazlaca yer alması halinde bu benzerlik oranının daha yüksek çıkması beklenmektedir. Aynı şekilde kelime ağırlıklı katsayılar kısmında ise kelime benzerlik çarpanına daha fazla ağırlık verildiğinden karşılaştırılan metin belgelerinde ortak kullanılan kelimelerin yer alması halinde bu benzerlik oranının yüksek çıkması beklenmektedir.

Genellikle birbirine çok yakın olan çalışmalarda ortak cümle sayısı fazla olacağından dolayı cümle ağırlıklı katsayılarla hesaplanan benzerlik oranı sayesinde bu tip çalışmaların yakalanması hedeflenmektedir. Yine aynı şekilde biraz emek harcayarak metin belgesinde yer alan cümlelerde kelimelerin yerlerinin değiştirilmesi şeklinde yapılan çalışmaları da kelime ağırlıklı katsayılar ile hesaplanan benzerlik oranı sayesinde yakalanması amaçlanmaktadır.

Şekil 3.6'daki klasör yolu seçim ekranı yer alan "Gözet" butonuna basılması halinde aşağıda görülen ve karşılaştırma işlemi yapılacak metin belgelerinin yer aldığı klasörün yolunun uygulamaya gösterildiği Şekil 3.8'deki pencere açılmaktadır.



Şekil 3.8. Klasör yolu seçim penceresi

Şekil 3.7’deki katsayı seçim ekranından benzerlik katsayıları belirlendikten ve klasör yolu seçim penceresinden karşılaştırma işlemi yapılacak metin belgelerinin yer aldığı klasör uygulamaya gösterildikten sonra Şekil 3.9’daki klasör yolu onay ekranı açılır.

**Dosyaların yükleneceği klasörün yolu:** C:\Users\ugur\Desktop\Örnek **Yolu Onayla**

Şekil 3.9. Klasör yolu onay ekranı

Bu ekranda uygulamaya gösterilen ve içerisinde karşılaştırma işlemi yapılacak olan txt uzantılı metin belgelerinin yer aldığı klasörün yolu gösterilmektedir. Ekranda yer alan “Yolu Onayla” butonuna basılması halinde uygulamaya gösterilen klasör içerisinde yer alan metin belgeleri birbirleriyle karşılaştırılarak benzerlik oranları hesaplanır, ortak olarak kullanılan cümleler ve kelimeler tespit edilir. Ardından Şekil 3.10’da gösterilen sonuç gösterim ekranı açılmaktadır.

Gösterdiğiniz klasör içinde toplam 7 tane dosya bulunmaktadır. Bunlardan 4 tanesi karşılaştırma işlemi yapılacak olan .txt uzantılı notepad dosyasıdır.

D1=Kaynak Dosya  
D2=Karşılaştırdığı Dosya

Sayı Yazdır

Dosya Adı : Seçiniz... v

D1 Adı	D2 Adı	D1 Cümle Sayısı	D2 Cümle Sayısı	D1 Kelime Sayısı	D2 Kelime Sayısı	D1 Boyutu	D2 Boyutu	Ortak Cümle Sayısı	Ortak Kelime Sayısı	Benzerlik Oranı	Cümle Ağırlıklı Benzerlik Oranı	Kelime Ağırlıklı Benzerlik Oranı	Detaylar
deneme1.txt	deneme2.txt	4	4	81	65	623bayt	495bayt	3	44	61,94	62,77	57,92	
deneme1.txt	deneme3.txt	6	6	81	52	623bayt	452bayt	0	15	7,41	5,54	9,69	
deneme1.txt	deneme4.txt	4	4	81	74	623bayt	629bayt	0	5	8,35	9,54	10,92	
deneme2.txt	deneme1.txt	4	4	65	81	495bayt	623bayt	3	44	67,29	66,77	64,92	
deneme2.txt	deneme3.txt	6	6	65	52	495bayt	452bayt	0	9	5,35	4	7	
deneme2.txt	deneme4.txt	4	4	65	74	495bayt	629bayt	0	6	3,71	2,77	4,85	
deneme3.txt	deneme1.txt	6	4	52	81	452bayt	623bayt	0	15	11,53	8,62	15,08	
deneme3.txt	deneme2.txt	6	4	52	65	452bayt	495bayt	0	9	7	5,23	9,15	
deneme3.txt	deneme4.txt	6	4	52	74	452bayt	629bayt	0	2	1,24	0,92	1,62	
deneme4.txt	deneme1.txt	4	4	74	81	629bayt	623bayt	0	5	8,35	9,54	10,92	
deneme4.txt	deneme2.txt	4	4	74	65	629bayt	495bayt	0	6	3,29	2,46	4,31	
deneme4.txt	deneme3.txt	6	6	74	52	629bayt	452bayt	0	2	0,82	0,62	1,08	

Şekil 3.10. Sonuç gösterim ekranı

Sonuç gösterim ekranında uygulamaya gösterilen klasör içerisinde yer alan txt uzantılı tüm metin belgelerinin birbirleriyle karşılaştırılması sonucu elde edilen bilgiler yer almaktadır. Ekranda karşılaştırılan metin belgelerinde kullanılan cümle sayıları, kelime sayıları, metin belgelerinin boyut bilgisi, karşılaştırılan metin belgelerinde ortak olarak kullanılmış cümle sayıları, ortak kelime sayıları ve değişik ağırlıklarla hesaplanmış olan benzerlik oranları gösterilmektedir.

Ekranın en üst kısmında uygulamaya gösterilen klasörde yer alan belge sayısı ve bunlardan kaç tanesinin karşılaştırma işlemi yapılan txt uzantılı metin belgeleri olduğu bilgisi yer almaktadır.

Ekranında yer alan “Sayfayı Yazdır” butonu ile sonuç gösterim ekranında gösterilen bilgilerin çıktısı alınabilmektedir. Ayrıca “Dosya Adı” kısmında yer alan seçim kutucuğuna tıklanması halinde Şekil 3.11’de görülen ve uygulamaya gösterilen klasör içerisinde yer alan tüm txt uzantılı metin belgelerinin isimlerinin listelendiği kısım açılmaktadır.

D1 Adı	D2	Cümle	D2 Cümle Sayısı
deneme1.txt	deneme2.txt	4	4
deneme1.txt	deneme3.txt	4	6
deneme1.txt	deneme4.txt	4	4

Şekil 3.11. Sonuçların filtrelenmesi

Bu kısımdan incelenmek istenilen metin belgesi seçilerek sonuç gösterim ekranına sadece seçilen metin belgesine ait bilgilerin gelmesi sağlanır. Örneğin, seçim kutusundan ”deneme1.txt” isimli metin belgesi seçildiğinde karşımıza aşağıda gösterilen ekran gelmektedir.



Gösterdiğiniz klasör içinde toplam 7 tane dosya bulunmaktadır. Bunlardan 4 tanesi karşılaştırma işlemi yapılacak olan .txt uzantılı notepad dosyasıdır.

D1=Kaynak Dosya  
D2=Karşılaştırdığı Dosya

Sayfayı Yazdır

Dosya Adı : deneme1.txt

D1 Adı	D2 Adı	D1 Cümle Sayısı	D2 Cümle Sayısı	D1 Kelime Sayısı	D2 Kelime Sayısı	D1 Boyutu	D2 Boyutu	Ortak Cümle Sayısı	Ortak Kelime Sayısı	Benzerlik Oranı	Cümle Ağırlıklı Benzerlik Oranı	Kelime Ağırlıklı Benzerlik Oranı	Detaylar
deneme1.txt	deneme2.txt	4	4	81	65	623bayt	495bayt	3	44	61,94	62,77	57,92	
deneme1.txt	deneme3.txt	4	6	81	52	623bayt	452bayt	0	15	7,41	5,54	9,69	
deneme1.txt	deneme4.txt	4	4	81	74	623bayt	629bayt	0	5	8,35	9,54	10,92	

Şekil 3.12. Filtrelenmiş sonuç gösterim ekranı

Yukarıdaki ekranda deneme1 isimli metin belgesinin karşılaştırma işlemi yapılan klasörde yer alan deneme2, deneme3 ve deneme4 isimli metin belgelerine ayrıntılı karşılaştırma sonuçları verilmiştir. Ekranda yer alan ilk satır deneme1 isimli metin belgesinde, 4 tane cümle, 81 tane kelime kullanıldığını, boyutunun ise 623 bayt olduğunu; deneme2 isimli dosyada ise 4 tane cümle, 65 tane kelime kullanıldığını, boyutunun 495 bayt olduğunu ve bu iki metin belgesinde 3 tane cümlenin, 44 tane kelimenin ortak olarak kullanıldığını gösterir. Ayrıca ekrandan farklı katsayı ağırlıklarınla hesaplanan benzerlik oranlarının 57, 62.77 ve 57.92 olduğu görülebilmektedir.

Sonuç gösterim ekranında her satırın en sağ kısmında yer alan “Detaylar” isimli sütündeki ikonlara tıklanması halinde ise tıklanan satırdaki metin belgelerine ait detaylı karşılaştırma bilgilerinin yer aldığı detay gözlem ekranı açılmaktadır. Yukarıdaki ekranda ilk satırdaki detay butonuna tıklanması halinde deneme1 ve deneme2 isimli metin belgeleri için detaylı karşılaştırma sonuçlarının yer aldığı aşağıdaki ekran açılmaktadır.

Toplam benzerlik oranı:61.94

### deneme1.txt kelimeleri

<< < Sayfa 1 > >> Kayıt Sayısı : 62

Kelime	Kullanım Sayısı
Bir	1
sistemin	2
yapımında	2
ya	4
da	4
bir	5
amaçın	1
yerine	2
getirilmesinde	2
arzu	1

### deneme2.txt kelimeleri

<< < Sayfa 1 > >> Kayıt Sayısı : 58

Kelime	Kullanım Sayısı
Bu	1
faktörlerin	1
tamamının	1
göz	1
önüne	1
alınmasının	1
ve	1
karar	1
verilmesinin	1
ana	1

Kelime benzerlik oranı:54

### Ortak Kullanılan Kelimeler

<< < Sayfa 1 > >> Kayıt Sayısı : 37

Kelime	Kullanım Sayısı
sistemin	2
yapımında	1
ya	3
da	2
bir	3
yerine	1
getirilmesinde	1
veya	1
gerekli	1
olan	2

### deneme1.txt cümleleri

<< < Sayfa 1 > >> Kayıt Sayısı : 4

Cümle	Kullanım Sayısı
Bir sistemin yapımında ya da bir amaç yerine getirilmesinde arzu edilen maliyet ve çalışma, belirli değişkenlerin bir fonksiyonu olarak ifade edilebildiği için, optimizasyon tekniği, bir fonksiyonu maksimize ya da minimize edecek olan kontrol sinyallerinin ya da parametrelerinin bulunması olarak tarif edilebilir	1
Daha genel manada, optimizasyon, fiziksel bir sistemin yapımında ya da bir kontrol işleminin yerine getirilmesinde en uygun çözümü elde etmektir	1
Optimizasyon problemlerinin çözümünde mevcut olan çok sayıda metod vardır	1
Bu metodlar daha ileri bölümlerde incelenecektir	1

### deneme2.txt cümleleri

<< < Sayfa 1 > >> Kayıt Sayısı : 4

Cümle	Kullanım Sayısı
Bu faktörlerin tamamının göz önüne alınmasının ve karar verilmesinin ana amacı ya fiziksel bir sistemin yapımındaki kazanç maksimum yapmak ya da bunun için gerekli olan maliyeti veya çalışmayı minimum yapmaktır	1
Daha genel manada, optimizasyon, fiziksel bir sistemin yapımında ya da bir kontrol işleminin yerine getirilmesinde en uygun çözümü elde etmektir	1
Optimizasyon problemlerinin çözümünde mevcut olan çok sayıda metod vardır	1
Bu metodlar daha ileri bölümlerde incelenecektir	1

### Ortak Kullanılan Cümleler

<< < Sayfa 1 > >> Kayıt Sayısı : 3

Cümle	Kullanım Sayısı
Daha genel manada, optimizasyon, fiziksel bir sistemin yapımında ya da bir kontrol işleminin yerine getirilmesinde en uygun çözümü elde etmektir	1
Optimizasyon problemlerinin çözümünde mevcut olan çok sayıda metod vardır	1
Bu metodlar daha ileri bölümlerde incelenecektir	1

Cümle benzerlik oranı:75

Şekil 3.13. Detay gözlem ekranı

Yukarıda görülen detay gözlem ekranında karşılaştırma işlemi yapılan deneme1 ve deneme2 isimli metin belgelerine ait detaylı bilgiler yer almaktadır. Bu ekrandan deneme1 isimli metin belgesinde kullanılan kelimeler, cümleler ve bunların kullanım sayısı bilgileri aynı şekilde deneme2 isimli metin belgesinde kullanılan kelimeler, cümleler ve bunların kullanım sayısı bilgileri yer almaktadır. Ayrıca ekranda her iki metin belgesinde ortak kullanılan kelimeler, cümleler ve bunların kullanım sayısı bilgileri de gösterilmektedir. Kullanıcı bütün bu bilgileri ilgili seçim kutucukları ve butonlar sayesinde kolayca filtreleyerek incelemek istediği bilgiye kolayca ulaşabilmektedir.

Ekranın en üst kısmında hesaplanan toplam benzerlik oranı, ekranın yan kısımlarında ise toplam benzerlik oranının hesaplanmasında rol alan kelime benzerlik oranı ve cümle benzerlik oranı gibi bilgiler sunulmaktadır.

Şekil 3.14'teki ekranda görüldüğü gibi karşılaştırma sonuçlarına ait bilgilerin gösterildiği kısımlarda kayıt sayısı bilgisi yer almakta ve kullanıcıya varsayılan olarak bu kayıtlardan ilk 10 tanesi gösterilmektedir. Ekranda gösterilen bu 10 kayıt dışındaki diğer kayıtların görülebilmesi için ilgili seçim kutusundan görmek istenilen sayfa seçilmeli veya seçim kutucuğunun yan kısımlarında yer alan butonlara tıklamak suretiyle listelenen kayıt seti değiştirilebilmektedir.

**deneme1.txt kelimeleri**

<< < Sayfa 1 > >> Kayıt Sayısı : 62

Kel	Kullanım Sayısı
Bir	1
sistemin	2
yapımında	2
ya	4
da	4
bir	5
amacın	1
yerine	2
getirilmesinde	2
arzu	1

Şekil 3.14. Kayıt sayfalama yapısı

### 3.3.2. Notepad tipindeki metin belgelerinde benzerlik oranlarının hesaplanması

Notepad tipindeki metin belgeleri için benzerlik oranı hesaplanırken cümle benzerlik çarpanı, kelime benzerlik çarpanı ve boyut benzerlik çarpanı olmak üzere üç farklı çarpan kullanılmaktadır.

Karşılaştırma işlemi yapılırken öncelikle kullanıcı tarafından uygulamaya gösterilen ve içerisinde karşılaştırma işlemi yapılacak olan txt uzantılı metin belgelerinin olduğu klasördeki dosyaların içerikleri cümle ve kelime bazında parçalanır ve ardından elde edilen bu bilgiler veritabanına kaydedilir. Daha sonra karşılaştırma işlemi yapılacak olan metin belgelerine ait cümle, kelime ve boyut bilgisi veritabanından çekilerek, karşılaştırılmak istenen her iki metin belgesinde ortak olarak kullanılan cümleler, kelimeler ve bunların kullanım sayıları tespit edilir.

Metin belgeleri arasındaki cümle benzerlik oranını bulmak için, karşılaştırma işlemi yapılan belgelerdeki ortak cümle sayısı baz alınarak doğru orantı kullanılır. Yani, cümle benzerlik oranı basitçe, aşağıdaki formülle ifade edilebilir.

$$\text{Cümle BO} = \frac{100 \times \text{Ortak cümle sayısı}}{\text{Karşılaştırma işlemi yapılan metin belgesindeki cümle sayısı}}$$

Benzer şekilde, kelime benzerlik oranı da karşılaştırma işlemi yapılan belgelerdeki ortak kelime sayısı baz alınarak basitçe, aşağıdaki formülle ifade edilebilir.

$$\text{Kelime BO} = \frac{100 \times \text{Ortak kelime sayısı}}{\text{Karşılaştırma işlemi yapılan metin belgesindeki kelime sayısı}}$$

Karşılaştırma işlemi yapılan metin belgeleri için boyut benzerlik oranı hesaplamak için ise karşılaştırma işlemi yapılan metin belgelerinin boyutları arasındaki fark alınır. Eğer bu fark 10 bayttan daha az ise dosyaların boyut benzerlik oranı %100, 10-20 bayt arasında ise dosyaların boyut benzerlik oranı %50, 20-30 bayt arasında ise dosyaların boyut benzerlik oranı %25 diğer tüm durumlar için ise bu oran %0 olarak alınmaktadır.

Ardından hesaplanan cümle benzerlik oranı kullanıcının katsayı seçim ekranında belirlemiş olduğu cümle benzerlik çarpanıyla, kelime benzerlik oranı kullanıcının belirlemiş olduğu kelime benzerlik çarpanıyla, aynı şekilde boyut benzerlik oranı ise kullanıcının belirlemiş olduğu boyut benzerlik çarpanıyla çarpılarak bu üç değer toplamı alınmaktadır. Bu toplamın, kullanıcının katsayı seçim ekranında girmiş olduğu cümle benzerlik çarpanı, kelime benzerlik çarpanı ve boyut benzerlik çarpanlarının toplamına bölünmesi sonucunda txt uzantılı metin belgeleri için benzerlik oranı hesaplanmış olur.

Yukarıda kullanıcının belirlediği ağırlıklar için benzerlik oranının nasıl hesaplandığı anlatılmıştır. Şekil 3.7'deki katsayı seçim ekranında kullanıcının belirlediği ağırlıklar dışında cümle ağırlıklı katsayılar ve kelime ağırlıklı katsayılar olmak üzere uygulamada sabit olan sunulan iki farklı ağırlık seti daha bulunmaktadır. Uygulama cümle ağırlıklı katsayılar ve kelime ağırlıklı katsayılar içinde benzerlik oranlarını aynı adımlardan geçerek hesaplamaktadır ve böylece kullanıcının belirlediği katsayılar, cümle ağırlıklı katsayılar ve kelime ağırlıklı katsayılar setleri için olmak üzere toplamda üç değişik benzerlik oranı hesaplanmış olmaktadır.

### 3.3.3. Word belgesi tipindeki metin belgelerinin karşılaştırılması

Uygulama çalıştırıldığında karşımıza Şekil 3.4'teki dosya tür seçim ekranı çıkmaktadır. Karşılaştırma işlemi yapılacak dosya türünün doc veya docx uzantılı metin belgeleri olduğu varsayılarak dosya tür seçim ekranından seçimin “Word” olarak yapıldığını düşünelim.

Dosya tür seçim ekranında karşılaştırma işlemi yapılacak dosya türü “Word” olarak seçildikten sonra Şekil 3.6'daki klasör yolu seçim ekranı açılmaktadır. Bu ekrandan “Karşılaştırma Katsayılarını Değiştir” butonuna tıklanması halinde benzerlik oranları hesaplanırken kullanılan katsayıların ağırlıklarının değiştirilebileceği katsayı seçim ekranına gidilmekte, “Gözet” butonuna tıklanması halinde ise açılan ekran vasıtası ile uygulamaya karşılaştırma işlemi yapılacak olan word belgelerinin yer aldığı klasörün yolu gösterilmektedir.

Şekil 3.6'daki klasör yolu seçim ekranından “Karşılaştırma Katsayılarını Değiştir” butonuna tıklanıldığında benzerlik oranları hesaplanırken kullanılan katsayıların ağırlıklarının değiştirileceği Şekil 3.15'de gösterilen word tipindeki metin belgeleri için katsayı seçim ekranı açılmaktadır.

Word tipindeki metin belgeleri için benzerlik oranları hesaplanırken; cümle benzerlik oranı, kelime benzerlik oranı, boyut benzerlik oranı ve yazar benzerlik oranı olmak üzere dört farklı benzerlik oranı kullanılmaktadır.

\* Cümle ağırlıklı katsayı ve kelime ağırlıklı katsayılar ile benzerlik oranları hesaplanacaktır. Sizde kendi katsayılarınızı oluşturarak yeni bir benzerlik oranı hesaplayabilirsiniz.

Sizin Katsayılarınız		Cümle Ağırlıklı Katsayılar		Kelime Ağırlıklı Katsayılar	
Cümle Benzerlik Çarpanı	: 9	Cümle Benzerlik Çarpanı	:8	Cümle Benzerlik Çarpanı	:5
Kelime Benzerlik Çarpanı	: 7	Kelime Benzerlik Çarpanı	:4	Kelime Benzerlik Çarpanı	:7
Boyut Benzerlik Çarpanı	: 1	Boyut Benzerlik Çarpanı	:1	Boyut Benzerlik Çarpanı	:1
Yazar Benzerlik Çarpanı	: 9	Yazar Benzerlik Çarpanı	:5	Yazar Benzerlik Çarpanı	:5

**Kaydet**

Şekil 3.15. Word tipindeki metin belgeleri için katsayı seçim ekranı

Word belgesi tipindeki metin belgeleri için kullanıcının kendisinin belirlemiş olduğu katsayılar, cümle ağırlıklı katsayılar ve kelime ağırlıklı katsayılar olmak üzere üç farklı katsayı grubuna göre üç tane benzerlik oranı hesaplanmaktadır. Uygulamada cümle ağırlıklı katsayılar ve kelime ağırlıklı katsayılar adı altındaki katsayılar kullanıcıya sabit olarak sunulmaktadır. Ekranın sol kısmında yer alan “Sizin Katsayılarınız” alanında yer alan seçim kutucukları vasıtası ile kullanıcıdan kendi katsayı ağırlıklarını belirlemesi istenmektedir. Böylece farklı ağırlıklardan oluşan üç katsayı grubuna göre farklı benzerlik oranları hesaplanmış olur. Farklı ağırlıklarla hesaplanan bu benzerlik oranları sayesinde kullanıcının değerlendirme aşamasında daha sağlıklı karar verebilmesine yardımcı olunur.

Cümle ağırlıklı katsayılar kısmında cümle benzerlik çarpanına daha fazla ağırlık verildiği için karşılaştırılan word belgelerinde ortak olarak kullanılan cümlelerin fazlaca yer alması halinde bu benzerlik oranının daha yüksek çıkması beklenmektedir. Aynı şekilde kelime ağırlıklı katsayılar kısmında ise kelime benzerlik çarpanına daha fazla ağırlık verildiği için karşılaştırılan metin belgelerinde ortak kullanılan kelimelerin fazlaca yer alması halinde bu benzerlik oranının yüksek çıkması beklenir.

Genellikle birbirine çok yakın olan çalışmalarda ortak cümle sayısı fazla olacağından dolayı cümle ağırlıklı katsayılarla hesaplanan benzerlik oranı sayesinde bu tip çalışmaların yakalanması hedeflenmektedir. Yine aynı şekilde biraz emek harcayarak metin belgesinde yer alan cümlelerde kelimelerin yerlerinin değiştirilmesi şeklinde yapılan çalışmaları da kelime ağırlıklı katsayılar ile hesaplanan benzerlik oranı sayesinde yakalanması hedeflenmektedir.

Şekil 3.6'daki klasör yolu seçim ekranında yer alan “Gözet” butonuna tıklanması halinde karşılaştırma işlemi yapılacak word belgelerinin yer aldığı klasörün yolunun uygulamaya gösterildiği Şekil 3.8'de görülen pencere açılmaktadır.

Şekil 3.15'deki word tipindeki metin belgeleri için katsayı seçim ekranından benzerlik katsayıları belirlendikten ve klasör yolu seçim penceresinden karşılaştırma



işlemi yapılacak olan word tipindeki metin belgelerinin yer aldığı klasör uygulamaya gösterildikten sonra Şekil 3.9'daki klasör yolu onay ekranı açılmaktadır.

Klasör yolu onay ekranında uygulamaya gösterilen ve içerisinde karşılaştırma işlemi yapılacak olan doc veya docx uzantılı word tipindeki metin belgelerinin yer aldığı klasörün yolu gösterilmektedir. Ekranda yer alan “Yolu Onayla” butonuna basılması halinde uygulamaya gösterilen klasör içerisinde yer alan word belgeleri birbirleriyle karşılaştırılarak benzerlik oranları hesaplanmakta, aynı yazar ismine sahip belgeler, ortak olarak kullanılan cümleler ve kelimeler tespit edilmektedir. Ardından Şekil 3.16'da gösterilen word tipindeki metin belgeleri için sonuç gösterim ekranı açılmaktadır.

Gösterdiğiniz klasör içinde toplam 7 tane dosya bulunmaktadır. Bunlardan 3 tanesi karşılaştırma işlemi yapılacak olan .docx uzantılı word dosyasıdır.

D1=Kaynak Dosya  
D2=Karşılaştırıldığı Dosya

Sayıyı Yazdır

Dosya Adı : Seçiniz...

D1 Adı	D2 Adı	D1 Cümle Sayısı	D2 Cümle Sayısı	D1 Kelime Sayısı	D2 Kelime Sayısı	D1 Boyutu	D2 Boyutu	Ortak Cümle Sayısı	Ortak Kelime Sayısı	Benzerlik Oranı	Cümle Ağırlıklı Benzerlik Oranı	Kelime Ağırlıklı Benzerlik Oranı	D1 Yazar	D2 Yazar	Detaylar
Ödev1.docx	Ödev2.docx	4	4	80	65	19542bayt	11321bayt	3	45	41,04	45,78	42,61	Zeynep	ugur	
Ödev1.docx	Ödev3.docx	4	6	80	51	19542bayt	11314bayt	0	15	4,85	4	7	Zeynep	ugur	
Ödev2.docx	Ödev1.docx	4	4	65	80	11321bayt	19542bayt	3	45	44,54	48,67	47,67	ugur	Zeynep	
Ödev2.docx	Ödev3.docx	4	6	65	51	11321bayt	11314bayt	0	9	41,96	36,22	38,39	ugur	ugur	
Ödev3.docx	Ödev1.docx	6	4	51	80	11314bayt	19542bayt	0	15	7,81	6,44	11,28	ugur	Zeynep	
Ödev3.docx	Ödev2.docx	6	4	51	65	11314bayt	11321bayt	0	9	43,04	37,11	39,94	ugur	ugur	

Şekil 3.16. Word tipindeki metin belgeleri için sonuç gösterim ekranı

Word tipindeki metin belgeleri için sonuç gösterim ekranında uygulamaya gösterilen klasör içerisinde yer alan doc veya docx uzantılı tüm word belgelerinin birbirleriyle karşılaştırılması sonucu elde edilen bilgiler yer almaktadır. Ekranında karşılaştırılan word belgelerinde kullanılan cümle sayıları, kelime sayıları, metin belgesinin boyut bilgisi, karşılaştırılan metin belgelerinde ortak olarak kullanılmış cümle sayıları, ortak kelime sayıları, yazar isim bilgileri ve farklı katsayı ağırlıklarınla hesaplanmış olan benzerlik oranları gösterilmektedir.

Ekranın en üst kısmında uygulamaya gösterilen klasör içerisinde yer alan belge sayıları ve bunlardan kaç tanesinin karşılaştırma işlemi yapılan doc veya docx uzantılı word belgesi olduğu bilgisi yer almaktadır.

Ekranında yer alan “Sayfayı Yazdır” butonu ile word tipindeki metin belgeleri için sonuç gösterim ekranında gösterilen bilgilerin çıktısı alınabilmektedir. Ayrıca “Dosya Adı” kısmında yer alan seçim kutucuğuna tıklanması halinde Şekil 3.17’de görülen ve uygulamaya gösterilen klasör içerisinde yer alan doc veya docx uzantılı tüm word belgelerinin isimlerinin listelendiği kısım açılmaktadır.

**D1=Kaynak Dosya**  
**D2=Karşılaştırıldığı Dosya**

Sayfayı Yazdır

Dosya Adı : Seçiniz...

D1 Adı	D2 Adı	Ortak Kelime Sayısı	D2 Cümle Sayısı
Ödev1.docx	Ödev2.docx	4	4
Ödev1.docx	Ödev3.docx	4	6
Ödev2.docx	Ödev1.docx	4	4

Şekil 3.17. Sonuçların filtrelenmesi(Word)

Bu kısımdan incelenmek istenilen metin belgesi seçilerek word tipindeki metin belgeleri için sonuç gösterim ekranına sadece seçilen metin belgesine ait bilgilerin gelmesi sağlanmaktadır. Örneğin seçim kutusundan ”Ödev1.docx” isimli metin belgesi seçildiğinde aşağıda ekran açılmaktadır.

Gösterdiğiniz klasör içinde toplam 7 tane dosya bulunmaktadır. Bunlardan 3 tanesi karşılaştırma işlemi yapılacak olan .docx uzantılı word dosyasıdır.

D1=Kaynak Dosya  
D2=Karşılaştırıldığı Dosya

Sayfayı Yazdır

Dosya Adı : Ödev1.docx

D1 Adı	D2 Adı	D1 Cümle Sayısı	D2 Cümle Sayısı	D1 Kelime Sayısı	D2 Kelime Sayısı	D1 Boyutu	D2 Boyutu	Ortak Cümle Sayısı	Ortak Kelime Sayısı	Benzerlik Oranı	Cümle Ağırlıklı Benzerlik Oranı	Kelime Ağırlıklı Benzerlik Oranı	D1 Yazar	D2 Yazar	Detaylar
Ödev1.docx	Ödev2.docx	4	4	80	65	19542bayt	11321bayt	3	45	62,76	63,38	59	Zeynep	ugur	
Ödev1.docx	Ödev3.docx	4	6	80	51	19542bayt	11314bayt	0	15	7,41	5,54	9,69	Zeynep	ugur	

Şekil 3.18. Filtrelenmiş sonuç gösterim ekranı(Word)

Şekil 3.18’de gösterilen ekranda, Ödev1 isimli word belgesinin karşılaştırma işlemi yapılan klasörde yer alan Ödev2 ve Ödev3 isimli word belgelerle ayrıntılı karşılaştırma sonuçları verilmiştir. Ekranda yer alan ilk satır Ödev1 isimli word belgesinde 4 tane cümle, 80 tane kelime kullanıldığını, boyutunun 19226 bayt olduğunu ve yazar isminin “Zeynep” olduğunu; Ödev2 isimli word belgesinde ise 4 tane cümle, 65 tane kelime kullanıldığını, boyutunun 10988 bayt olduğunu, yazar isminin “Uğur” olduğunu ve bu iki word belgesinde 3 tane cümlenin, 45 tane kelimenin ortak olarak kullanıldığı gösterilmektedir. Ayrıca ekrandan farklı katsayı ağırlıklarınla hesaplanan benzerlik oranlarının 62.76, 63.38 ve 59 olduğu görülebilmektedir.

Word tipindeki metin belgeleri için sonuç gösterim ekranındaki her satırın en sağ kısmında yer alan “Detaylar” isimli sütundaki ikonlara tıklanması halinde ise tıklanan satırdaki word belgelerine ait detaylı karşılaştırma bilgilerinin yer aldığı word belgesi detay gözlem ekranı açılmaktadır. Yukarıdaki ekranda ilk satırdaki detay butonuna tıklandığında Ödev1 ve Ödev2 isimli word belgeleri için detaylı karşılaştırma sonuçlarının yer aldığı aşağıdaki ekran açılmaktadır.

Toplam benzerlik oranı:41.04

Ödev1.docx dosyasının yazarı: Zeynep

Ödev2.docx dosyasının yazarı: ugr

### Ödev1.docx kelimeleri

<< < > >> Sayfa 1 / 1 Kayıt Sayısı : 60

Kelime	Kullanım Sayısı
Bir	1
sistemin	2
yapımında	2
ya	4
da	4
bir	5
amaç	1
yerine	2
getirilmesinde	2
arzu	1

### Ödev2.docx kelimeleri

<< < > >> Sayfa 1 / 1 Kayıt Sayısı : 56

Kelime	Kullanım Sayısı
Bu	2
faktörlerin	1
tamamının	1
göz	1
önüne	1
alınmasının	1
ve	1
karar	1
verilmesinin	1
ana	1

Kelime benzerlik oranı:56

### Ortak Kullanılan Kelimeler

<< < > >> Sayfa 1 / 1 Kayıt Sayısı : 38

Kelime	Kullanım Sayısı
sistemin	2
yapımında	1
ya	3
da	2
bir	3
yerine	1
getirilmesinde	1
veya	1
gerekli	1
olan	2

Cümle benzerlik oranı:75

### Ödev1.docx cümleleri

<< < > >> Sayfa 1 / 1 Kayıt Sayısı : 4

Cümle	Kullanım Sayısı
Bir sistemin yapımında ya da bir amaç yerine getirilmesinde arzu edilen yarar veya gerekli olan maliyet ve çalışma, belirli değışkenlern bir fonksiyonu olarak ifade edilebildiđi için, optimizasyon tekniđi, bir fonksiyonu maksimize ya da minimize edecek olan kontrol sinyallerinin ya da parametrelerinin bulunması olarak tarif edilebilir	1
Daha genel manada, optimizasyon, fiziksel bir sistemin yapımında ya da bir kontrol işleminin yerine getirilmesinde en uygun çözümü elde etmektir	1
Optimizasyon problemlerinin çözümünde mevcut olan çok sayıda metod vardır	1
Bu metodlar daha ileri bölümlerde incelenecektir	1

### Ödev2.docx cümleleri

<< < > >> Sayfa 1 / 1 Kayıt Sayısı : 4

Cümle	Kullanım Sayısı
Bu faktörlerin tamamının göz önüne alınmasının ve karar verilmesinin ana amacı ya fiziksel bir sistemin yapımındaki kazancı maksimum yapmak ya da bunun için gerekli olan maliyeti veya çalışmayı minimum yapmaktır	1
Daha genel manada, optimizasyon, fiziksel bir sistemin yapımında ya da bir kontrol işleminin yerine getirilmesinde en uygun çözümü elde etmektir	1
Optimizasyon problemlerinin çözümünde mevcut olan çok sayıda metod vardır	1
Bu metodlar daha ileri bölümlerde incelenecektir	1

### Ortak Kullanılan Cümleler

<< < > >> Sayfa 1 / 1 Kayıt Sayısı : 3

Cümle	Kullanım Sayısı
Daha genel manada, optimizasyon, fiziksel bir sistemin yapımında ya da bir kontrol işleminin yerine getirilmesinde en uygun çözümü elde etmektir	1
Optimizasyon problemlerinin çözümünde mevcut olan çok sayıda metod vardır	1
Bu metodlar daha ileri bölümlerde incelenecektir	1

Şekil 3.19. Word belgesi detay gözlem ekranı

Yukarıda görülen word belgesi detay gözlem ekranında karşılaştırma işlemi yapılan Ödev1 ve Ödev2 isimli word belgelerine ait detaylı bilgiler yer almaktadır. Bu ekranda Ödev1 isimli metin belgesinde kullanılan kelimeler, cümleler, bunların kullanım sayısı bilgisi ve Ödev1 isimli word belgesinin yazar isim bilgisi aynı şekilde Ödev2 isimli metin belgesinde kullanılan kelimeler, cümleler, bunların kullanım sayısı bilgisi ve Ödev2 isimli word belgesinin yazar isim bilgileri yer almaktadır. Ayrıca ekranda her iki word belgesinde ortak kullanılan kelimeler, cümleler ve ortak kullanım sayısı bilgisi de bulunmaktadır. Kullanıcı bütün bu bilgileri ilgili seçim kutucukları ve butonlar sayesinde kolayca filtreleyerek rahatça inceleyebilmektedir.

Ekranın en üst kısmında hesaplanan toplam benzerlik oranı, yan kısımlarda ise toplam benzerlik oranının hesaplanmasında rol alan kelime benzerlik oranı ve cümle benzerlik oranı gibi bilgiler sunulmaktadır.

Şekil 3.20'deki ekranda görüldüğü gibi karşılaştırma sonuçlarına ait bilgilerin gösterildiği kısımlarda kayıt sayısı bilgisi yer almakta ve kullanıcıya varsayılan olarak bu kayıtlardan ilk 10 tanesi gösterilmektedir. Ekranda gösterilen bu 10 kayıt dışındaki diğer kayıtların görülebilmesi için ilgili seçim kutusundan görmek istenilen sayfa seçilmeli veya seçim kutucuğunun yan kısımlarında yer alan butonlara tıklamak suretiyle listelenen kayıt seti değiştirilebilmektedir.

**Ödev1.docx kelimeleri**

<< < Sayfa 1 > >> Kayıt Sayısı : 60

Kel	Kullanım Sayısı
Bir	1
sistemin	2
yapımında	2
ya	4
da	4
bir	5
amacın	1
yerine	2
getirilmesinde	2
arzu	1

Şekil 3.20. Kayıt sayfalama yapısı(Word)

### 3.3.4. Word belgesi tipindeki metin dosyalarında benzerlik oranlarının hesaplanması

Word belgesi tipindeki metin belgeleri için benzerlik oranı hesaplanırken; cümle benzerlik çarpanı, kelime benzerlik çarpanı, yazar benzerlik çarpanı ve boyut benzerlik çarpanı olmak üzere dört farklı çarpan kullanılmaktadır.

Karşılaştırma işlemi yapılırken öncelikle kullanıcı tarafından uygulamaya gösterilen ve içerisinde karşılaştırma işlemi yapılacak olan doc veya docx uzantılı metin belgelerinin olduğu klasördeki dosyalar tek tek okunarak, içerikleri cümle ve kelime bazında parçalanır ardından elde edilen bu bilgiler veritabanına kaydedilir. Daha sonra karşılaştırma işlemi yapılacak olan metin belgelerinin cümleleri, kelimeleri, yazar ve boyut bilgisi veritabanından çekilerek karşılaştırılan her iki word belgesinde ortak kullanılan cümleler, kelimeler ve bunların kullanım sayıları tespit edilir.

Word belgeleri arasındaki cümle benzerlik oranını bulmak için, karşılaştırma işlemi yapılan belgelerdeki ortak cümle sayısı baz alınarak doğru orantı kullanılır. Yani, cümle benzerlik oranı basitçe, aşağıdaki formülle ifade edilebilir.



$$\text{Cümle BO} = \frac{100 \times \text{Ortak cümle sayısı}}{\text{Karşılaştırma işlemi yapılan word belgesindeki cümle sayısı}}$$

Benzer şekilde, karşılaştırma işlemi yapılan belgelerdeki ortak kelime sayısı baz alınır ise kelime benzerlik oranı basitçe aşağıdaki formül ile ifade edilir.

$$\text{Kelime BO} = \frac{100 \times \text{Ortak kelime sayısı}}{\text{Karşılaştırma işlemi yapılan word belgesindeki kelime sayısı}}$$

Karşılaştırma işlemi yapılan word belgeleri için boyut benzerlik oranı hesaplanırken karşılaştırma işlemi yapılan word belgelerinin boyutları arasındaki fark alınır. Eğer bu fark 10 bayttan daha az ise dosyaların boyut benzerlik oranı %100, eğer fark 10 bayt ile 20 bayt arasında ise dosyaların boyut benzerlik oranı %50, eğer fark 20 bayt ile 30 bayt arasında ise dosyaların boyut benzerlik oranı %25 diğer tüm durumlar için ise bu oran %0 olarak alınmaktadır.

Yazar benzerlik oranı eğer her iki metin belgesinin yazar ismi aynı ise oran yüzde yüz farklı ise yüzde sıfır olarak kabul edilmektedir.

Yazar benzerlik oranı word belgeleri için karşılaştırma işlemi yapılırken çok önemli bir role sahiptir. Yazar adı word belgelerinin özellikler kısmında yer alan ve word belgesinin oluşturulduğu bilgisayarın adını alan bir özelliktir. Bu sebeple A bilgisayarında oluşturulmuş daha sonra B bilgisayarında üzerinde değişiklik yapmak suretiyle hazırlanan intihallerin tespitinde önemli bir faktördür. Ardından hesaplanan cümle benzerlik oranı kullanıcının word tipindeki metin belgeleri için katsayı seçim ekranında belirlemiş olduğu cümle benzerlik çarpanıyla, kelime benzerlik oranı kullanıcının belirlemiş olduğu kelime benzerlik çarpanıyla, yazar benzerlik oranı kullanıcının belirlemiş olduğu yazar benzerlik çarpanıyla, aynı şekilde boyut benzerlik oranı ise kullanıcının belirlemiş olduğu boyut benzerlik çarpanıyla çarpılarak bu dört değer toplamı alınmaktadır. Bu toplamın kullanıcının word tipindeki metin belgeleri için katsayı seçim ekranında girmiş olduğu cümle benzerlik çarpanı, kelime benzerlik çarpanı, yazar benzerlik çarpanı ve boyut benzerlik

çarpınlarının toplamına bölünmesi sonucunda doc veya docx uzantılı word belgeleri için benzerlik oranı hesaplanmıř olur.

### 3.3.5. Kaynak kod dosyalarının karşılaştırılması

Uygulama çalıştırıldığında karşımıza Şekil 3.4'teki dosya tür seçim ekranı çıkmaktadır. Karşılaştırma işlemi yapılacak dosya türünün kod dosyaları olduğu varsayılarak ekrandan seçimin "Kod" olarak yapıldığını varsayalım.

Dosya tür seçim ekranında karşılaştırma işlemi yapılacak dosya türü "Kod" olarak seçildikten sonra Şekil 3.6'daki klasör yolu seçim ekranı açılmaktadır. Bu ekrandan "Karşılaştırma Katsayılarını Değiştir" butonuna tıklanması halinde benzerlik oranları hesaplanırken kullanılan katsayıların ağırlıklarının değiştirilebileceği katsayı seçim ekranına gidilmekte, "Gözet" butonuna tıklanması halinde ise açılan ekran vasıtası ile uygulamaya karşılaştırma işlemi yapılacak olan kod dosyalarının yer aldığı klasörün yolu gösterilmektedir.

Şekil 3.6'daki klasör yolu seçim ekranından "Karşılaştırma Katsayılarını Değiştir" butonuna tıklandığında benzerlik oranları hesaplanırken kullanılan katsayıların ağırlıklarının değiştirilebileceği Şekil 3.21'deki kod dosyaları için katsayı seçim ekranı açılmaktadır.

Varsayılan katsayılara göre bir benzerlik oranları hesaplanacaktır.Sizde kendi katsayılarınızı oluşturarak yeni bir benzerlik oranı hesaplayabilirsiniz.

Sizin Katsayılarınız		Varsayılan Katsayılar	
for-if-while Sayısı Benzerlik Çarpanı	: 5	for-if-while Sayısı Benzerlik Çarpanı	:4
Değişken İsimleri(String,İnteger ...) Benzerlik Çarpanı	: 9	Değişken İsimleri(String,İnteger ...) Benzerlik Çarpanı	:8
Değişken(int,string,char,bool) Sayısı Benzerlik Çarpanı	: 5	Değişken(int,string,char,bool) Sayısı Benzerlik Çarpanı	:4
Textbox,buton,label İsim Benzerlik Çarpanı	: 9	Textbox,buton,label İsim Benzerlik Çarpanı	:8
Textbox,buton,label Sayısı Benzerlik Çarpanı	: 5	Textbox,buton,label Sayısı Benzerlik Çarpanı	:4
Boyut Benzerlik Çarpanı	: 5	Boyut Benzerlik Çarpanı	:1

**Kaydet**

Şekil 3.21. Kod dosyaları için katsayı seçim ekranı

Kod dosyaları için benzerlik oranları hesaplanırken; for-if-while sayısı benzerlik oranı, kod kısmında kullanılan değişken isimleri benzerlik oranı ve kullanım sayıları

benzerlik oranı, görsel kısımda kullanılan textbox-buton-label isimleri benzerlik oranı ve kullanım sayıları benzerlik oranı, boyut benzerlik oranı katsayıları olmak üzere toplam altı farklı benzerlik oranı kullanılmaktadır.

Kod dosyaları için kullanıcının kendisinin belirlemiş olduğu katsayılar ve varsayılan katsayılar olmak üzere iki farklı katsayı grubuna göre iki tane benzerlik oranı hesaplanmaktadır. Uygulamada varsayılan katsayılar adı altındaki katsayılar kullanıcıya sabit olarak sunulmaktadır. Ekranın sol kısmında yer alan “Sizin Katsayılarınız” alanında yer alan seçim kutucukları vasıtası ile kullanıcıdan kendi katsayı ağırlıklarını belirlemesi istenmektedir. Böylece farklı ağırlıklardan oluşan iki çeşit katsayı grubuna göre benzerlik oranları hesaplanmış olur. Farklı ağırlıklarla hesaplanan bu benzerlik oranları sayesinde kullanıcının değerlendirme aşamasında daha sağlıklı karar verebilmesine yardımcı olunur.

Şekil 3.6’daki klasör yolu seçim ekranında yer alan “Gözet” butonuna basıldığında karşılaştırma işlemi yapılacak kod dosyalarının yer aldığı klasörün yolunun uygulamaya gösterildiği Şekil 3.8’de görülen pencere açılmaktadır.

Şekil 3.21’deki kod dosyaları için katsayı seçim ekranından benzerlik katsayıları belirlendikten ve klasör yolu seçim penceresinden karşılaştırma işlemi yapılacak olan kod dosyalarının yer aldığı klasör uygulamaya gösterildikten sonra Şekil 3.22’deki klasör yolu onay ekranı açılmaktadır.

**Dosyaların yükleneceği klasörün yolu:** C:\Users\ugur\Desktop\Örnek\Ödev\Proje1\Yeni klasör **Yolu Onayla**

Şekil 3.22. Klasör yolu onay ekranı(Kod dosyası)

Klasör yolu onay ekranında uygulamaya gösterilen ve içerisinde karşılaştırma işlemi yapılacak olan kod dosyalarının yer aldığı klasörün yolu gösterilmektedir. Ekranda yer alan “Yolu Onayla” butonuna basılması halinde uygulamaya gösterilen klasör içerisinde yer alan kod dosyaları birbirleriyle karşılaştırılarak benzerlik oranları hesaplanmakta ve benzer olan kısımları tespit edilmektedir. Ardından Şekil 3.23’te gösterilen kod dosyaları için sonuç gösterim ekranı açılmaktadır.

Gösterdiğiniz klasör içinde toplam 4 tane karşılaştırma işlemi yapılacak kod proje dosyası bulunmaktadır.

D1=Kaynak Dosya  
D2=Karşılaştırıldığı Dosya

Sayıyı Yazdır

Dosya Adı : Seçiniz... ▾

D1 Adı	D2 Adı	D1 for-if-while Sayısı	D2 for-if-while Sayısı	D1 Değişken Sayısı	D2 Değişken Sayısı	D1 Görsel Kısımındaki Değişken Sayısı	D2 Görsel Kısımındaki Değişken Sayısı	D1 Boyutu	D2 Boyutu	Aynı İsimli Değişken Sayısı	Aynı İsimli Görsel Değişken Sayısı	Benzerlik Oranı	Varsayılan Katsayılı Benzerlik Oranı	Detaylar
Proje1	Proje2	39	66	9	36	6	12	18230bayt	31500bayt	6	2	45,92	46,08	
Proje1	Proje4	39	49	9	47	6	5	18230bayt	29492bayt	9	4	70,46	71,09	
Proje1	Proje3	39	51	9	40	6	4	18230bayt	29770bayt	8	3	60,79	61,16	
Proje2	Proje1	66	39	36	9	12	6	31500bayt	18230bayt	12	4	35,66	36,89	
Proje2	Proje4	66	49	36	47	12	5	31500bayt	29492bayt	34	4	65,12	63,53	
Proje2	Proje3	66	51	36	40	12	4	31500bayt	29770bayt	34	4	68,49	66,37	
Proje4	Proje1	49	39	47	9	5	6	29492bayt	18230bayt	9	4	49,18	52,47	
Proje4	Proje2	49	66	47	36	5	12	29492bayt	31500bayt	17	2	48,98	49,29	
Proje4	Proje3	49	51	47	40	5	4	29492bayt	29770bayt	39	3	78,94	78,92	
Proje3	Proje1	51	39	40	9	4	6	29770bayt	18230bayt	9	3	47,09	49,74	
Proje3	Proje2	51	66	40	36	4	12	29770bayt	31500bayt	17	2	56,99	56,64	
Proje3	Proje4	51	49	40	47	4	5	29770bayt	29492bayt	40	3	88,22	87,75	

Şekil 3.23. Kod dosyaları için sonuç gösterim ekranı

Kod dosyaları için sonuç gösterim ekranında uygulamaya gösterilen klasör içerisinde yer alan cs ve aspx uzantılı tüm kod dosyalarının birbirleriyle karşılaştırılması sonucu elde edilen bilgiler yer almaktadır. Ekranda karşılaştırılan kod dosyalarında kullanılan for-if-while sayıları, kod kısmında kullanılan değişken sayıları, görsel kısımda kullanılan textbox-label-buton sayıları, boyut bilgileri, iki kod dosyasında ortak kullanılan değişken sayıları, görsel kısımda ortak kullanılan textbox-label-buton sayıları ve değişik ağırlıklarla hesaplanmış olan benzerlik oranları gösterilmektedir.

Ekranın en üst kısmında uygulamaya gösterilen klasör içerisinde yer alan dosya sayıları ve bunlardan kaç tanesinin karşılaştırma işlemi yapılan cs ve aspx uzantılı kod dosyası olduğu bilgisi yer almaktadır.

Ekranında yer alan “Sayfayı Yazdır” butonu ile kod dosyaları için sonuç gösterim ekranında gösterilen bilgilerin çıktısı alınabilmektedir. Ayrıca “Dosya Adı” kısmında yer alan seçim kutucuğuna tıklanması halinde Şekil 3.24’te görülen ve uygulamaya gösterilen klasör içerisinde yer alan cs ve aspx uzantılı kod dosyalarının isimlerinin listelendiği kısım açılmaktadır.

**D1=Kaynak Dosya**  
**D2=Karşılaştırıldığı Dosya**

Sayfayı Yazdır

Dosya Adı : Seçiniz...

D1 Adı	D2 Adı	if-while	D2 for-if-while Sayısı
Proje1	Proje2	39	66
Proje1	Proje4	39	49
Proje1	Proje3	39	51

Şekil 3.24. Sonuçların filtrelenmesi(Kod dosyası)

Bu kısımdan incelenmek istenilen kod dosyası seçilerek kod dosyaları için sonuç gösterim ekranına sadece seçilen kod dosyasına ait bilgilerin gelmesi sağlanmaktadır.

Örneğin seçim kutusundan Proje1 isimli kod dosyası seçildiğinde karşımıza aşağıda gösterilen ekran gelmektedir.

Gösterdiğiniz klasör içinde toplam 1 tane karşılaştırma işlemi yapılacak kod proje dosyası bulunmaktadır.

D1=Kaynak Dosya  
D2=Karşılaştırdığı Dosya

Sayfayı Yazdır

Dosya Adı : Proje1

D1 Adı	D2 Adı	D1 for-if-while Sayısı	D2 for-if-while Sayısı	D1 Değişken Sayısı	D2 Değişken Sayısı	D1 Görsel Kısımdeki Değişken Sayısı	D2 Görsel Kısımdeki Değişken Sayısı	D1 Boyutu	D2 Boyutu	Aynı İsimli Değişken Sayısı	Aynı İsimli Görsel Değişken Sayısı	Benzerlik Oranı	Varsayılan Katsayılı Benzerlik Oranı	Detaylar
Proje1	Proje2	39	66	9	36	6	12	18230bayt	31500bayt	6	2	45,92	46,08	
Proje1	Proje4	39	49	9	47	6	5	18230bayt	29492bayt	9	4	70,46	71,09	
Proje1	Proje3	39	51	9	40	6	4	18230bayt	29770bayt	8	3	60,79	61,16	

Şekil 3.25. Filtrelenmiş sonuç gösterim ekranı(Kod dosyası)

Yukarıdaki ekranda Proje1 isimli kod dosyasının karşılaştırma işlemi yapılan klasörde yer alan Proje2, Proje3 ve Proje4 isimli kod dosyalarıyla ayrıntılı karşılaştırma sonuçları verilmiştir. Ekranda yer alan ilk satırda Proje1 isimli kod dosyasında 39 tane for-if-while, kod kısmında 9 tane değişken, görsel kısımda 6 tane textbox-label-buton kullanıldığı ve boyutunun ise 18230 bayt olduğu; Proje2 isimli dosyada 66 tane for-if-while, kod kısmında 36 tane değişken, görsel kısımda 12 tane textbox-label-buton kullanıldığı ve boyutunun ise 31500 bayt olduğu gösterilmektedir. Ayrıca karşılaştırma işlemi yapılan bu iki dosyanın kod kısmında 6 tane aynı isimli değişken kullanıldığı, görsel kısımda 2 tane aynı isimli değişken kullanıldığı ve farklı ağırlıklarla hesaplanan benzerlik oranlarının ise 45.92 ve 46.08 olduğu görülebilmektedir.

Kod dosyaları için sonuç gösterim ekranında gösterilen her satırın en sağ kısmında yer alan “Detaylar” isimli sütundaki ikonlara tıklanması halinde ise tıklanan satırdaki kod dosyalarına ait detaylı karşılaştırma bilgilerinin yer aldığı kod dosyası detay gözlem ekranı açılmaktadır. Yukarıdaki ekranda ilk satırdaki detay butonuna tıklandığında Proje1 ve Proje2 isimli kod dosyaları için detaylı karşılaştırma sonuçlarının yer aldığı aşağıdaki ekran açılmaktadır.



Toplam benzerlik oranı:45,92

### Proje1 projesindeki değişkenler

<< < Sayfa 1 > >> Kayıt Sayısı : 15

Seçiniz... ▼

Değişken Adı	Değişken Tipi
b	integer
i	integer
j	integer
path	string
a	string
file_name	string
file_name2	string
a	bool
btnKatsayi	asp:Button

### Ortak Kullanılan Değişkenler

<< < Sayfa 1 > >> Kayıt Sayısı : 8

Seçiniz... ▼

Değişken Adı	Değişken Tipi
b	integer
i	integer
j	integer
a	string
file_name	string
file_name2	string
btnKatsayi	asp:Button
txtOnay	asp:Button

### Proje2 projesindeki değişkenler

<< < Sayfa 1 > >> Kayıt Sayısı : 48

Seçiniz... ▼

Değişken Adı	Değişken Tipi
txtsayisi	integer
dosyasayisi	integer
KelimeSayisi	integer
nokta_index	integer
soruisareti_index	integer
unlem_index	integer
counter	integer
counter2	integer
b	integer
i	integer

	Proje1	Proje2
If Sayısı	23	44
For Sayısı	3	4
While Sayısı	13	18
Int Sayısı	4	22
String Sayısı	4	14
Bool Sayısı	1	0
Long Sayısı	0	0
Char Sayısı	0	0
asp:Button Sayısı	3	4
asp:Label Sayısı	1	4
asp:TextBox Sayısı	2	4

Şekil 3.26. Kod dosyası detay gözlem ekranı

Yukarıda görülen kod dosyası detay gözlem ekranında karşılaştırma işlemi yapılan Proje1 ve Proje2 isimli kod dosyalarına ait detaylı bilgiler bulunmaktadır.

Bu ekrandan Proje1 isimli kod dosyasında for-if-while, textbox-label-buton ve değişkenlerin(int, string, bool, char, long) kullanım sayısı bilgileri aynı şekilde Proje2 isimli kod dosyasında for-if-while, textbox-label-buton ve değişkenlerin(int, string, bool, char, long) kullanım sayısı bilgileri yer almaktadır. Ayrıca ekranın sağ tarafındaki kısımdan her bir projede kullanılan değişken isimleri ve görsel kısımda kullanılan textbox-label-buton isimlerine ulaşılabilir. Ekranda yer alan “Ortak Kullanılan Değişkenler” kısmından ise her iki projede aynı isimle kullanılmış değişkenler görülebilmektedir. Kullanıcı bütün bu bilgileri ilgili seçim kutucukları ve butonlar sayesinde kolayca filtreleyerek rahatça inceleyebilmektedir.

Ekranın en üst kısmında hesaplanan toplam benzerlik oranı bilgisi sunulmaktadır. Şekil 3.27’teki ekranda görüldüğü gibi karşılaştırma sonuçlarına ait bilgilerin gösterildiği kısımlarda kayıt sayısı bilgisi yer almakta ve kullanıcıya varsayılan olarak bu kayıtlardan ilk 10 tanesi gösterilmektedir. Ekranda gösterilen bu 10 kayıt dışındaki diğer kayıtların görülebilmesi için ilgili seçim kutusundan görmek istenilen sayfa seçilmeli veya seçim kutusunun yan kısımlarında yer alan butonlara tıklamak suretiyle listelenen kayıt seti değiştirilebilmektedir.

## Proje1 projesindeki deęişkenler

<< < Sayfa 1 v > >> Kayıt Sayısı : 15

Deęişken Adı	Deęişken Tipi
b	integer
i	integer
i	integer
j	integer
path	string
a	string
file_name	string
file_name2	string
a	bool
btnKatsayi	asp:Button

Seęiniz...  
integer  
string  
bool  
long  
char  
asp:Button  
asp:Label  
asp:TextBox

Şekil 3.27. Kayıt sayfalama yapısı(Kod dosyası)

Aşaęıda gösterilen ve karşılaştırma işlemi yapılan kod dosyalarında ortak kullanılan deęişkenlerin gösterildięi kısımlardan her iki kod dosyasında ortak kullanılan deęişkenler gözlemlenebilmekte, seęim kutucuęundan incelenmek istenilen deęişken tipi seęilerek kayıtlara filtre konulabilmektedir.

## Ortak Kullanılan Deęişkenler

<< < Sayfa 1 v > >> Kayıt Sayısı : 8

Deęişken Adı	Deęişken Tipi
b	integer
i	integer
i	integer
a	string
file_name	string
file_name2	string
btnKatsayi	asp:Button
txtOnay	asp:Button

Seęiniz...  
integer  
string  
bool  
long  
char  
asp:Button  
asp:Label  
asp:TextBox

Şekil 3.28. Kayıt filtreleme(Kod dosyası)

### 3.3.6. Kaynak kod dosyalarında benzerlik oranlarının hesaplanması

Kod dosyaları için benzerlik oranı hesaplanırken; for-if-while sayısı benzerlik çarpanı, değişken isimleri benzerlik çarpanı, değişken sayısı benzerlik çarpanı, textbox-buton-label isim benzerlik çarpanı, textbox-buton-label sayısı benzerlik çarpanı ve boyut benzerlik çarpanı olmak üzere altı farklı çarpan kullanılmaktadır.

Karşılaştırma işlemi yapılırken öncelikle kullanıcı tarafından uygulamaya gösterilen ve içerisinde karşılaştırma işlemi yapılacak olan cs ve aspx uzantılı kod dosyalarının olduğu klasördeki dosyalar tek tek okunarak, kullanılan değişken isimleri, değişken sayıları, dosya boyut bilgisi ve gerekli diğer bilgiler veritabanına kaydedilir. Ardından karşılaştırma işlemi yapılacak olan kod dosyalarının bilgileri veritabanından çekilerek karşılaştırılan her iki kod dosyasında ortak kullanılan değişkenler, görsel kısımda ortak olan kısımlar ve bunların kullanım sayıları gibi bilgiler tespit edilmektedir.

Kod dosyaları arasındaki for-if-while benzerlik oranını, değişken isimleri benzerlik oranını ve değişken sayıları benzerlik oranını bulmak için sırasıyla aşağıdaki formüller verilebilir.

$$\text{for – if – while BO} = \frac{100 \times \text{İkinci dosyadaki for – if – while sayısı}}{\text{Karşılaştırma işlemi yapılan kod dosyasındaki for – if – while sayısı}}$$

$$\text{Değişken isimleri BO} = \frac{100 \times \text{Ortak değişken sayısı}}{\text{Karşılaştırma işlemi yapılan kod dosyasında KDS}}$$

$$\text{Değişken sayıları BO} = \frac{100 \times \text{İkinci dosyadaki değişken sayısı}}{\text{Karşılaştırma işlemi yapılan kod dosyasında KDS}}$$

Benzer şekilde kod dosyaları arasındaki görsel kısımdaki değişken isimleri benzerlik oranını ve değişken sayıları benzerlik oranını aşağıdaki formüllerle ifade edilebilir.

$$\text{Görsel kısımdaki değişken isimleri BO} = \frac{100 \times \text{Ortak değişken sayısı}}{\text{Karşılaştırma işlemi yapılan kod dosyasının görsel kısmında KDS}}$$

$$\text{Görsel kısımdaki değişken sayıları BO} = \frac{100 \times \text{İkinci dosyadaki görsel değişken sayısı}}{\text{Karşılaştırma işlemi yapılan kod dosyasının görsel kısmında KDS}}$$

Karşılaştırma işlemi yapılan kod dosyaları için boyut benzerlik oranı hesaplanırken karşılaştırma işlemi yapılan kod dosyalarının boyutları arasındaki fark alınır. Eğer bu fark 2000 bayttan daha az ise dosyaların boyut benzerlik oranı %100, eğer fark 2000 bayt ile 4000 bayt arasında ise dosyaların boyut benzerlik oranı %50, eğer fark 4000 bayt ile 8000 bayt arasında ise dosyaların boyut benzerlik oranı %25 diğer tüm durumlar için ise bu oran %0 olarak kabul edilmektedir.

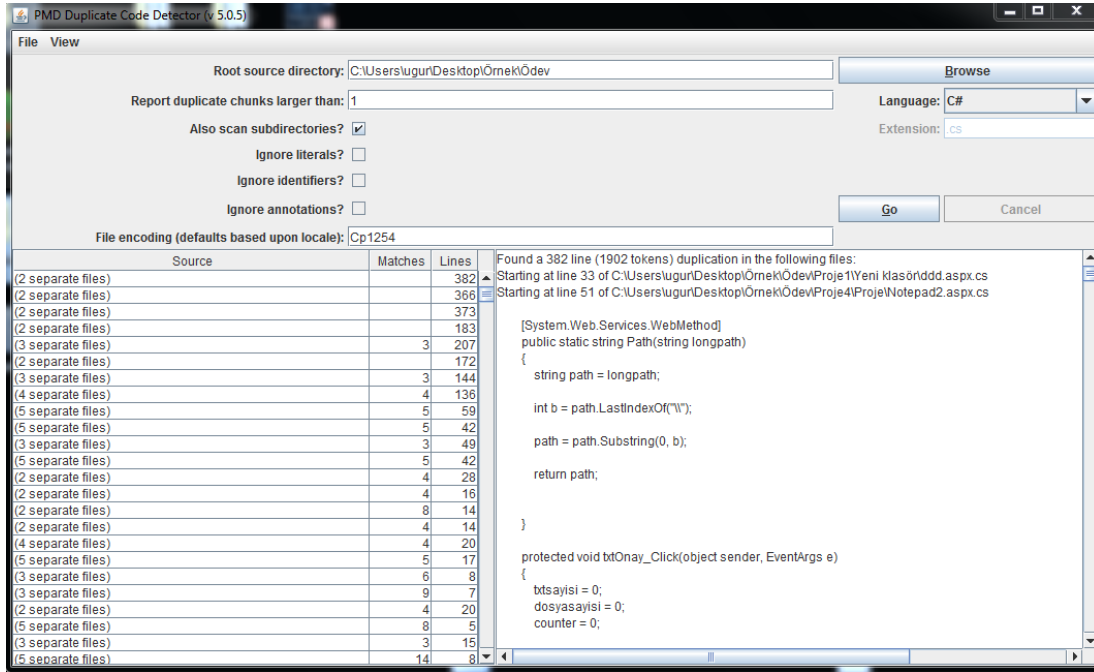
Ardından hesaplanan for-if-while benzerlik oranı kullanıcının kod dosyaları için katsayı seçim ekranında belirlemiş olduğu for-if-while sayısı benzerlik çarpanıyla, değişken isimleri benzerlik oranı kullanıcının belirlemiş olduğu değişken isimleri benzerlik çarpanıyla, değişken sayısı benzerlik oranı kullanıcının belirlemiş olduğu değişken sayısı benzerlik çarpanıyla, textbox-buton-label isim benzerlik oranı kullanıcının belirlemiş olduğu textbox-buton-label isim benzerlik çarpanıyla, textbox-buton-label sayısı benzerlik oranı kullanıcının belirlemiş olduğu textbox-buton-label sayısı benzerlik çarpanıyla, aynı şekilde boyut benzerlik oranı ise kullanıcının belirlemiş olduğu boyut benzerlik çarpanıyla çarpılarak bu altı değer toplamı alınmaktadır. Bu toplamın kullanıcının kod dosyalar için katsayı seçim ekranında girmiş olduğu for-if-while sayısı benzerlik çarpanı, değişken isimleri benzerlik çarpanı, değişken sayısı benzerlik çarpanı, textbox-buton-label isim benzerlik çarpanı, textbox-buton-label sayısı benzerlik çarpanı ve boyut benzerlik çarpanlarının toplamına bölünmesiyle cs ve aspx uzantılı kod dosyaları için benzerlik oranı hesaplanmış olur.

### 3.4. Uygulamanın Cpd ile Karşılaştırılması

Bu kısımda geliştirmiş olduğumuz uygulama ile cpd programının karşılaştırması yapılmıştır. Çalışmadaki uygulama ile karşılaştırmış olduğumuz C# dosyaları cpd programı ile tekrar karşılaştırılmış ve aşağıdaki sonuçlar elde edilmiştir.

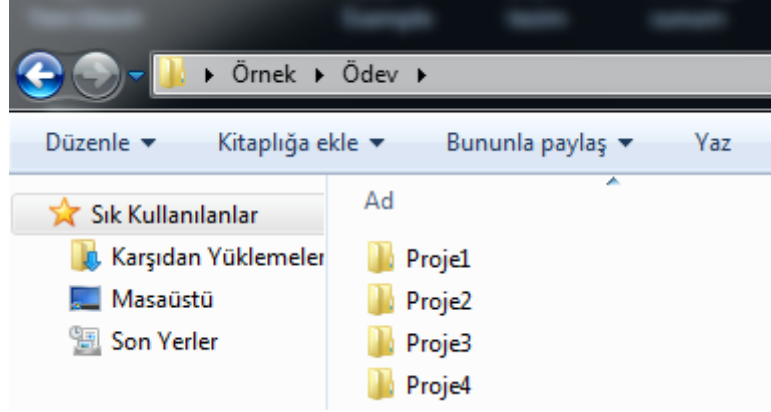
Cpd (Copy/Paste Dedector) ile karşılaştırma işlemi yapabilmek için öncelikle karşılaştırma işlemi yapılacak olan dosyaların olduğu klasör uygulamaya

gösterilmeli, ardından dil seçimi yapılmalıdır. Burada dil seçiminin C# olarak yapıldığını varsayalım. Gerekli seçim işlemleri yapıldıktan sonra ekranda yer alan “Go” butonuna tıkladığında karşılaştırma işlemi yapılmaktadır. Aşağıda bu işlemlerin yapıldığı cpd programının ekran görüntüsü verilmiştir.



Şekil 3.29. Cpd karşılaştırma ekranı

Sonuç olarak cpd bu çalışmada geliştirilen uygulamadan farklı olarak sadece cs uzantılı dosyalar için karşılaştırma işlemi yapmakta olup aspx uzantılı dosyaların karşılaştırılması işlemini desteklememektedir. Dolayısı ile bu program görsel kısımda yapılan intihalleri tespit edememektedir. Ayrıca, cpd karşılaştırma işlemi sonucu olarak kopyalanmış satırları ve bu satırların dosyalarda hangi kısımlarda yer aldığı bilgisini vermekte fakat karşılaştırılan dosyalar için herhangi bir benzerlik oranı vermemektedir. Cpd, dosya bazlı bir uygulamadır. Örneğin aşağıdaki şekilde görüldüğü üzere karşılaştırma işlemi yapılmak istenilen klasörün içerisinde ödev klasörlerinin olduğunu kabul edelim. Bu ödev klasörlerinden herhangi bir tanesi birden çok cs uzantılı dosyadan oluşabilir. Bu durumda cpd klasör bazlı çalışmadığından klasörler arası karşılaştırma işlemi mümkün olmamaktadır.



Şekil 3.30. Karşılaştırılan klasörler

### 3.5. Sonuç

Çalışmanın ana bölümü olan bu kısımda, çalışmanın avantajlarından ve akışından bahsedilmiştir. Ayrıca bu bölümde, metin dosyaları ve C# ile geliştirilmiş web tabanlı uygulamalar için programın kullanımı, çalışma mantığı, benzerlik oranlarının nasıl hesaplandığı anlatılmış, uygulamanın örnek dosyalar için çalıştırılması sonucu elde edilen görseller paylaşılmıştır.

## **BÖLÜM 4. UYGULAMADA KULLANILAN VERİTABANI TABLOLARI, PROSEDÜRLER VE KULLANIM AMAÇLARI**

### **4.1. Uygulamada Kullanılan Tablolar ve Kullanım Amaçları**

“Table\_Project\_Variable” isimli tabloda karşılaştırma işlemi yapılan kod dosyalarında kullanılan değişkenlerin tür ve isim bilgileri yer almaktadır. Tabloda karşılaştırma işlemi yapılan kod dosyalarının ait olduğu projenin adının tutulduğu “Project\_Name” sütunu, kod dosyasının adının tutulduğu “Page\_Name” sütunu, değişken tip bilgisinin tutulduğu “Variable\_Type” sütunu ve değişken ad bilgisinin tutulduğu “Variable\_Name” sütunu olmak üzere toplam dört sütun bulunmaktadır.

<b>Table_Project_Variable</b>	
	Project_Name
	Page_Name
	Variable_Type
	Variable_Name

Şekil 4.1. Değişken bilgilerinin tutulduğu tablo



Tabloda yer alan verilerden küçük bir kesit aşağıda verilmiştir.

	Project_Name	Page_Name	Variable_Type	Variable_Name
1	Project1	trial.aspx.cs	integer	b
2	Project1	trial.aspx.cs	integer	i
3	Project1	trial.aspx.cs	integer	i
4	Project1	trial.aspx.cs	integer	j
5	Project1	trial.aspx.cs	string	path
6	Project1	trial.aspx.cs	string	a
7	Project1	trial.aspx.cs	string	file_name
8	Project1	trial.aspx.cs	string	file_name2
9	Project1	trial.aspx.cs	bool	a
10	Project2	suplier.aspx.cs	integer	bttsayisi
11	Project2	suplier.aspx.cs	integer	dosyasayisi

Şekil 4.2. Değişken bilgilerinin tutulduğu tablodan kesit

“Table\_Structural\_Count” isimli tabloda karşılaştırma işlemi yapılan kod dosyalarında kullanılan değişken tip bilgisi ve bunların kullanım sayısı bilgileri ayrıca kod dosyalarında for-if-while yapılarının kullanım sayı bilgileri yer almaktadır. Tabloda karşılaştırma işlemi yapılan kod dosyasının ait olduğu projenin adının tutulduğu “Project\_Name” sütunu, kod dosyasının adının tutulduğu “Page\_Name” sütunu, tip bilgisinin tutulduğu “Structural\_Type” sütunu ve kullanım sayı bilgisinin tutulduğu “Structural\_Count” sütunu olmak üzere toplam dört sütun bulunmaktadır.

Table_Structural_Count	
	Project_Name
	Page_Name
	Structural_Type
	Structural_Count

Şekil 4.3. Değişken tipleri ve kullanım sayılarının tutulduğu tablo

Tabloda yer alan verilerden küçük bir kesit aşağıda verilmiştir.

	Project_Name	Page_Name	Structural_Type	Structural_Count
1	Project 1	trial.aspx.cs	int	4
2	Project 1	trial.aspx.cs	string	4
3	Project 1	trial.aspx.cs	bool	1
4	Project 1	trial.aspx.cs	long	0
5	Project 1	trial.aspx.cs	char	0
6	Project 1	trial.aspx.cs	for	3
7	Project 1	trial.aspx.cs	while	13
8	Project 1	trial.aspx.cs	if	23
9	Project2	supplier.aspx.cs	int	11
10	Project2	supplier.aspx.cs	string	7
11	Project2	supplier.aspx.cs	bool	0

Şekil 4.4. Değişken tipleri ve kullanım sayılarının tutulduğu tablodan kesit

“Tbl\_Factors” isimli tabloda karşılaştırma işlemi yapılan word belgeleri ve notepad tipindeki metin belgeleri için benzerlik hesaplaması yapılırken kullanılan katsayı ağırlıkları tutulmaktadır. Tabloda kelime katsayısının tutulduğu “WORDFACTOR” sütunu, cümle katsayısının tutulduğu “SENTENCEFACTOR” sütunu, boyut katsayısının tutulduğu “SIZEFACTOR” sütunu ve belgelerin yazar katsayısının tutulduğu “WRITERFACTOR” sütunu olmak üzere toplam dört sütun bulunmaktadır.

<b>Tbl_Factors</b>	
	WORDFACTOR
	SENTENCEFACTOR
	SIZEFACTOR
	WRITERFACTOR

Şekil 4.5. Metin belgeleri için katsayıların tutulduğu tablo

Tabloda yer alan verilerden küçük bir kesit aşağıda verilmiştir.

	WORDFACTOR	SENTECEFACTOR	SIZEFACTOR	WRITERFACTOR
1	7	9	1	9

Şekil 4.6. Metin belgeleri için katsayıların tutulduğu tablodan kesit

“Tbl\_Factors4Code” isimli tabloda karşılaştırma işlemi yapılan kod dosyaları için benzerlik hesaplaması yapılırken kullanılan katsayı ağırlıkları tutulmaktadır. Tabloda for-if-while sayısı katsayısının tutulduğu “FOR\_IF\_COUNT” sütunu, değişken isim katsayısının tutulduğu “VARIABLE\_NAMES” sütunu, değişken sayısı katsayısının tutulduğu “VARIABLE\_COUNT” sütunu, textbox-buton-label isim katsayısının tutulduğu “BUTON\_LABEL\_NAMES” sütunu, textbox-buton-label sayısı katsayısının tutulduğu “BUTON\_LABEL\_COUNT” sütunu ve boyut katsayısının tutulduğu “SIZEFACTOR” sütunu olmak üzere toplam altı sütun bulunmaktadır.

Tbl_Factors4Code	
	FOR_IF_COUNT
	VARIABLE_NAMES
	VARIABLE_COUNT
	BUTON_LABEL_NAMES
	BUTON_LABEL_COUNT
	SIZE_FACTOR

Şekil 4.7. Kod dosyaları için katsayıların tutulduğu tablo

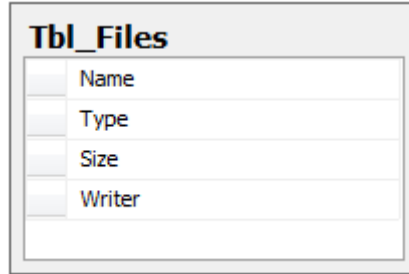
Tabloda yer alan verilerden küçük bir kesit aşağıda verilmiştir.

	FOR_IF_COUNT	VARIABLE_NAMES	VARIABLE_COUNT	BUTON_LABEL_NAMES	BUTON_LABEL_COUNT	SIZE_FACTOR
1	3	8	4	7	3	1

Şekil 4.8. Kod dosyaları için katsayıların tutulduğu tablodan kesit

“Tbl\_Files” isimli tabloda uygulamaya gösterilen ve karşılaştırma işlemi yapılacak olan klasör içerisinde yer alan metin belgelerinin ve kod dosyalarının isim bilgileri, tür bilgileri, boyut ve yazar bilgileri yer almaktadır. Tabloda isim bilgisinin tutulduğu

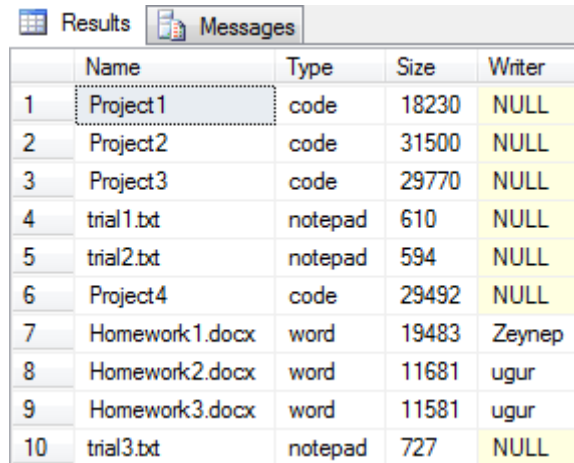
“Name” sütunu, tip bilgisinin tutulduğu “Type” sütunu, boyut bilgisinin tutulduğu “Size” sütunu ve yazar bilgisinin tutulduğu “Writer” sütunu olmak üzere toplam dört sütun bulunmaktadır.



Tbl_Files				
	Name			
	Type			
	Size			
	Writer			

Şekil 4.9. Klasör içerik bilgisinin tutulduğu tablo

Tabloda yer alan verilerden küçük bir kesit aşağıda verilmiştir.



	Name	Type	Size	Writer
1	Project1	code	18230	NULL
2	Project2	code	31500	NULL
3	Project3	code	29770	NULL
4	trial1.txt	notepad	610	NULL
5	trial2.txt	notepad	594	NULL
6	Project4	code	29492	NULL
7	Homework1.docx	word	19483	Zeynep
8	Homework2.docx	word	11681	ugur
9	Homework3.docx	word	11581	ugur
10	trial3.txt	notepad	727	NULL

Şekil 4.10. Klasör içerik bilgisinin tutulduğu tablodan kesit

“Tbl\_Sentences” isimli tabloda karşılaştırma işlemi yapılan word belgesi ve notepad tipindeki metin belgelerinde kullanılan cümleler ve kullanım sayısı bilgileri yer almaktadır. Tabloda belge adının tutulduğu “DOCUMENT\_NAME” sütunu, belge tipinin tutulduğu “DOCUMENT\_TYPE” sütunu, kullanılan cümle bilgisinin tutulduğu “SENTENCE” sütunu ve cümle kullanım sayı bilgisinin tutulduğu “COUNTS” sütunu olmak üzere toplam dört sütun bulunmaktadır.

Tbl_Sentences	
DOCUMENT_NAME	
DOCUMENT_TYPE	
SENTENCE	
COUNTS	

Şekil 4.11. Cümle kullanım bilgilerinin tutulduğu tablo

Tabloda yer alan verilerden küçük bir kesit aşağıda verilmiştir.

Results		Messages		
	DOCUMENT_NAME	DOCUMENT_TYPE	SENTENCE	COUNTS
1	Homework1.docx	word	<a href="#">It was rather surprising to discover a deep vein of...</a>	1
2	Homework1.docx	word	<a href="#">fried and his lawyer for many years and had watc...</a>	1
3	Homework1.docx	word	<a href="#">He had been moderately succesfull in business, s...</a>	1
4	Homework1.docx	word	<a href="#">from business and to trawel about the world a littl...</a>	1
5	Homework1.docx	word	<a href="#">he and Esther were content to sit night after night...</a>	1
6	Homework2.docx	word	<a href="#">It was rather surprising to discover a deep vein of...</a>	1
7	Homework2.docx	word	<a href="#">fried and his lawyer for many years and had watc...</a>	1
8	Homework2.docx	word	<a href="#">Looking back over the years of my friendship with...</a>	1
9	Homework2.docx	word	<a href="#">The marriage was a very nice affair There followe...</a>	1
10	Homework3.docx	word	<a href="#">It was rather surprising to discover a deep vein of...</a>	1

Şekil 4.12. Cümle kullanım bilgilerinin tutulduğu tablodan kesit

“Tbl\_Words” isimli tabloda karşılaştırma işlemi yapılan word belgelerinde ve notepad tipindeki metin belgelerinde kullanılan kelimeler ve kullanım sayısı bilgileri yer almaktadır. Tabloda belge adının tutulduğu “DOCUMENT\_NAME” sütunu, belge tipinin tutulduğu “DOCUMENT\_TYPE” sütunu, kullanılan kelime bilgisinin tutulduğu “WORD” sütunu ve kelime kullanım sayı bilgisinin tutulduğu “COUNTS” sütunu olmak üzere toplam dört sütun bulunmaktadır.

Tbl_Words	
DOCUMENT_NAME	
DOCUMENT_TYPE	
WORD	
COUNTS	

Şekil 4.13. Kelime kullanım bilgilerinin tutulduğu tablo

Tabloda yer alan verilerden küçük bir kesit aşağıda verilmiştir.

	DOCUMENT_NAME	DOCUMENT_TYPE	WORD	COUNTS
1	trial3.txt	notepad	we	1
2	trial3.txt	notepad	found	1
3	trial3.txt	notepad	less	2
4	trial3.txt	notepad	incommon	1
5	trial3.txt	notepad	passed	1
6	trial4.txt	notepad	it	1
7	trial4.txt	notepad	was	1
8	trial4.txt	notepad	rather	1
9	trial4.txt	notepad	surprising	1
10	trial4.txt	notepad	to	2

Şekil 4.14. Kelime kullanım bilgilerinin tutulduğu tablodan kesit

## 4.2. Uygulamada Kullanılan Prosedürler ve Kullanım Amaçları

### 4.2.1. Veritabanı Tablolarını Temizleyen Prosedür

Aşağıdaki şekilde “P\_CLEAR\_TABLES” isimli prosedür yer almaktadır. Bu prosedür uygulama her çalıştırıldığında veritabanı tablolarında yer alan eski değerlerin temizlenmesi amacıyla kullanılmaktadır. Prosedür input olarak varchar tipinde “P\_TYPE” isimli bir tane parametre almaktadır. Prosedüre gelen bu parametrenin değerinden uygulamanın hangi tür dosyalar için çalıştırıldığı anlaşılmakta ve gelen bu parametrenin değerine göre veritabanı tablolardan hangi verilerin silineceğine karar verilmektedir. Uygulamanın notepad tipindeki metin belgeleri için çalıştırıldığını düşünürsek prosedür “P\_TYPE” isimli parametreye notepad değeri verilerek çağrılmaktadır. Prosedürün çalışması ile birlikte dosya isim bilgilerinin, boyut bilgilerinin ve yazar bilgilerinin bulunduğu “Tbl\_Files” isimli tablodan notepad tipindeki veriler için kaydedilmiş bilgiler, kullanılan cümleler ve kullanım sayı bilgilerinin tutulduğu “Tbl\_Sentences” isimli tablodan notepad tipindeki veriler için kaydedilmiş bilgiler, kullanılan kelimeler ve kullanım sayı bilgilerinin tutulduğu “Tbl\_Words” isimli tablodan notepad tipindeki veriler için kaydedilmiş bilgiler silinerek tablolar yeni gelecek veriler için hazırlanmaktadır.

```

CREATE PROCEDURE [dbo].[P_CLEAR_TABLES]
(
  @P_TYPE          varchar(100)
)

AS
BEGIN

  IF @P_TYPE ='notepad'
  BEGIN
    DELETE FROM Tbl_Files      WHERE Type='notepad';
    DELETE FROM Tbl_Sentences  WHERE DOCUMENT_TYPE='notepad';
    DELETE FROM Tbl_Words      WHERE DOCUMENT_TYPE='notepad';
  END
  ELSE IF @P_TYPE ='word'
  BEGIN
    DELETE FROM Tbl_Files      WHERE Type='word';
    DELETE FROM Tbl_Sentences  WHERE DOCUMENT_TYPE='word';
    DELETE FROM Tbl_Words      WHERE DOCUMENT_TYPE='word';
  END
  ELSE IF @P_TYPE ='code'
  BEGIN
    DELETE FROM Tbl_Files      WHERE Type='code';
    DELETE FROM Table_Project_Variable;
    DELETE FROM Table_Structural_Count;
  END

END

END

```

Şekil 4.15. Veritabanı tablolarını temizleyen prosedür

#### 4.2.2. Kullanılan Kelimeleri ve Kullanım Sayısı Bilgilerini Kaydeden Prosedür

Aşağıdaki şekilde “P\_INSERT“ isimli prosedür yer almaktadır. Bu prosedür uygulama notepad ve word türündeki metin belgelerini karşılaştırmak için çalıştırıldığında metin belgelerinde kullanılan kelimeleri ve kullanım sayısı bilgilerini kaydetmek amacıyla kullanılmaktadır. Prosedür input olarak varchar tipinde “P\_DOCUMENT\_NAME”, “P\_DOCUMENT\_TYPE”, “P\_WORD” ve “P\_COUNT” isimli dört tane parametre almaktadır. Prosedür aldığı bu dört değeri kullanılan kelimeler ve kullanım sayısı bilgilerinin tutulduğu “Tbl\_Words” isimli tabloya kaydetmektedir.

```

CREATE PROCEDURE [dbo].[P_INSERT]
(
  @P_DOCUMENT_NAME      varchar(100),
  @P_DOCUMENT_TYPE      varchar(100),
  @P_WORD                varchar(100),
  @P_COUNT               varchar(100)
)
AS
BEGIN
  INSERT INTO Tbl_Words (DOCUMENT_NAME, DOCUMENT_TYPE, WORD, COUNTS)
  values (@P_DOCUMENT_NAME, @P_DOCUMENT_TYPE, @P_WORD, @P_COUNT);
END

```

Şekil 4.16. Kelimeleri ve kullanım sayı bilgilerini kaydeden prosedür

#### 4.2.3. Karşılaştırma İşlemi Yapılan Dosya Bilgilerini Kaydeden Prosedür

Aşağıdaki şekilde “P\_INSERT\_FILE” isimli prosedür yer almaktadır. Bu prosedür kullanıcının uygulamaya göstermiş olduğu ve içerisinde karşılaştırma işlemi yapılacak olan dosyaların yer aldığı klasördeki dosyaların tam adını, tür bilgilerini, boyut ve yazar adı bilgilerini kaydetmek amacıyla kullanılmaktadır. Prosedür input olarak varchar tipinde “P\_FILE\_NAME”, “P\_FILE\_TYPE”, “P\_SIZE”, “P\_WRITER” ve “P\_TYPE” isimli beş tane parametre almaktadır. Prosedür aldığı değerleri dosya bilgilerinin tutulduğu “Tbl\_Files” isimli tabloya kaydetmektedir.

```

CREATE PROCEDURE [dbo].[P_INSERT_FILE]
(
  @P_FILE_NAME          varchar(100),
  @P_FILE_TYPE          varchar(100),
  @P_SIZE                varchar(100),
  @P_WRITER              varchar(100),
  @P_TYPE                varchar(100)
)
AS
BEGIN
  IF @P_TYPE = 'I'
    BEGIN
      INSERT INTO Tbl_Files (Name, Type, Size)
      values (@P_FILE_NAME, @P_FILE_TYPE, @P_SIZE );
    END
  ELSE IF @P_TYPE = 'U'
    BEGIN
      UPDATE Tbl_Files
      SET Writer=@P_WRITER
      WHERE Name=@P_FILE_NAME AND Type=@P_FILE_TYPE;
    END
END

```

Şekil 4.17. Karşılaştırma işlemi yapılan dosya bilgilerini kaydeden prosedür



#### 4.2.4. Kullanılan Değişken Bilgilerini Kaydeden Prosedür

Aşağıdaki şekilde “P\_INSERT\_PROJECT\_VARIABLE” isimli prosedür yer almaktadır. Bu prosedür kod dosyalarında kullanılan değişkenleri, değişkenlerin tip bilgilerini, kod dosyalarının isim bilgilerini ve bu kod dosyalarının yer aldığı proje bilgilerini kaydetmek amacıyla kullanılmaktadır. Prosedür input olarak varchar tipinde “P\_PROJECT\_NAME”, “P\_PAGE\_NAME”, “P\_VARIABLE\_TYPE” ve “P\_VARIABLE\_NAME” isimli dört tane parametre almaktadır. Prosedür aldığı değerleri kod dosyalarında kullanılan değişken bilgilerinin tutulduğu “Table\_Project\_Variable” isimli tabloya kaydetmektedir.

```
CREATE PROCEDURE [dbo].[P_INSERT_PROJECT_VARIABLE]
(
  @P_PROJECT_NAME      varchar(100),
  @P_PAGE_NAME         varchar(100),
  @P_VARIABLE_TYPE     varchar(100),
  @P_VARIABLE_NAME     varchar(100)
)
AS
BEGIN
  INSERT INTO
  Table_Project_Variable (Project_Name, Page_Name, Variable_Type, Variable_Name)
  values (@P_PROJECT_NAME, @P_PAGE_NAME, @P_VARIABLE_TYPE, @P_VARIABLE_NAME
);
END
```

Şekil 4.18. Değişken bilgilerini kaydeden prosedür

#### 4.2.5. Kullanılan Cümleleri ve Kullanım Sayısı Bilgilerini Kaydeden Prosedür

Aşağıdaki şekilde “P\_INSERT\_SENTENCES” isimli prosedür yer almaktadır. Bu prosedür uygulama notepad ve word tipindeki metin belgelerini karşılaştırmak için çalıştırıldığında kullanılmaktadır. Prosedür metin belgelerinde kullanılan cümleleri ve kullanım sayısı bilgilerini kaydetmek amacıyla kullanılmaktadır. Prosedür input olarak varchar tipinde “P\_DOCUMENT\_NAME”, “P\_DOCUMENT\_TYPE”, “P\_COUNT” ve xml tipinde “P\_SENTENCE” olmak üzere toplam dört tane parametre almaktadır. Prosedür aldığı bu dört değeri kullanılan cümleler ve kullanım sayısı bilgilerinin tutulduğu “Tbl\_Sentences” isimli tabloya kaydetmektedir.

```

CREATE PROCEDURE [dbo].[P_INSERT_SENTENCES]
(
  @P_DOCUMENT_NAME      varchar(100),
  @P_DOCUMENT_TYPE      varchar(100),
  @P_SENTENCE           xml,
  @P_COUNT              varchar(100)
)
AS
BEGIN
  INSERT INTO Tbl_Sentences (DOCUMENT_NAME, DOCUMENT_TYPE, SENTENCE, COUNTS)
  values (@P_DOCUMENT_NAME, @P_DOCUMENT_TYPE, @P_SENTENCE, @P_COUNT);
END

```

Şekil 4.19. Cümleleri ve kullanım sayı bilgilerini kaydeden prosedür

#### 4.2.6. Programlama Diline Ait Özel Terimlerin Kullanım Bilgisini Kaydeden Prosedür

Aşağıdaki şekilde “P\_INSERT\_STRUCTURAL\_VARIABLE” isimli prosedür yer almaktadır. Bu prosedür uygulama kod dosyalarını karşılaştırmak için çalıştırıldığında kullanılmaktadır. Prosedür int, string gibi programlama diline ait terimlerin kullanım sayısı bilgilerini kaydetmek amacıyla kullanılmaktadır. Prosedür input olarak varchar tipinde “P\_PROJECT\_NAME”, “P\_PAGE\_NAME”, “P\_STRUCTURAL\_TYPE” ve “P\_COUNT” olmak üzere toplam dört tane parametre almaktadır. Prosedür aldığı bu dört değeri değişken tipleri ve kullanım sayısı bilgilerinin tutulduğu “Table\_Structural\_Count” isimli tabloya kaydetmektedir.

```

CREATE PROCEDURE [dbo].[P_INSERT_STRUCTURAL_VARIABLE]
(
  @P_PROJECT_NAME      varchar(100),
  @P_PAGE_NAME         varchar(100),
  @P_STRUCTURAL_TYPE   varchar(100),
  @P_COUNT             varchar(100)
)
AS
BEGIN
  INSERT INTO
  Table_Structural_Count (Project_Name, Page_Name, Structural_Type, Structural_Count)
  values (@P_PROJECT_NAME, @P_PAGE_NAME, @P_STRUCTURAL_TYPE, @P_COUNT);
END

```

Şekil 4.20. Özel terimleri ve kullanım sayısı bilgilerini kaydeden prosedür

#### 4.2.7. Metin Belgeleri için Katsayı Ağırlık Bilgilerini Kaydeden Prosedür

Aşağıdaki şekilde “P\_INSUP\_FACTORS” isimli prosedür yer almaktadır. Bu prosedür metin belgeleri için benzerlik oranı hesaplanırken kullanılan ve kullanıcı tarafından belirlenmiş olan katsayı ağırlıklarının veritabanına kaydedilmesi amacı ile kullanılmaktadır. Prosedür input olarak varchar tipinde “P\_WORD\_FACTOR”, “P\_SENTENCE\_FACTOR”, “P\_SIZE\_FACTOR” ve “P\_WRITER\_FACTOR” olmak üzere toplam dört tane parametre almaktadır. Metin belgeleri için katsayı ağırlık bilgilerinin tutulduğu “Tbl\_Factors” isimli tabloda kayıt olmaması durumunda alınan katsayı ağırlık bilgileri tabloya kaydedilmekte, tabloda mevcut ağırlıkların var olması durumunda ise tablodaki eski değerler prosedüre gelen ve kullanıcının girmiş olduğu yeni bilgiler ile değiştirilmektedir.

```

CREATE PROCEDURE [dbo].[P_INSUP_FACTORS]
(
    @P_WORD_FACTOR          varchar(100),
    @P_SENTENCE_FACTOR     varchar(100),
    @P_SIZE_FACTOR         varchar(100),
    @P_WRITER_FACTOR       varchar(100)
)
AS
DECLARE @V_COUNT INTEGER;
BEGIN
SELECT @V_COUNT = COUNT(*) FROM Tbl_Factors;

    IF @V_COUNT=0 OR @V_COUNT IS NULL
    BEGIN
        INSERT INTO
Tbl_Factors (WORDFACTOR, SENTENCEFACTOR, SIZEFACTOR, WRITERFACTOR)
values (@P_WORD_FACTOR, @P_SENTENCE_FACTOR, @P_SIZE_FACTOR, @P_WRITER_FACTOR);
        END

    ELSE
    BEGIN

        IF @P_WRITER_FACTOR IS NULL

            BEGIN

                UPDATE Tbl_Factors
                SET WORDFACTOR          = @P_WORD_FACTOR
                , SENTENCEFACTOR       = @P_SENTENCE_FACTOR
                , SIZEFACTOR           = @P_SIZE_FACTOR

            END

        ELSE
        BEGIN
            UPDATE Tbl_Factors
            SET WORDFACTOR          = @P_WORD_FACTOR
            , SENTENCEFACTOR       = @P_SENTENCE_FACTOR
            , SIZEFACTOR           = @P_SIZE_FACTOR
            , WRITERFACTOR         = @P_WRITER_FACTOR

        END

    END

END
END

```

Şekil 4.21. Metin belgeleri için katsayı ağırlık bilgisini kaydeden prosedür

#### 4.2.8. Kod Dosyaları için Katsayı Ağırlık Bilgilerini Kaydeden Prosedür

Aşağıdaki şekilde “P\_INSUP\_FACTORS4CODE” isimli prosedür yer almaktadır. Bu prosedür kod dosyaları arasında benzerlik oranı hesaplarken kullanılan ve kullanıcının belirlemiş olduğu katsayı ağırlıklarının veritabanına kaydedilmesi amacı ile kullanılmaktadır. Prosedür input olarak varchar tipinde “P\_FOR\_IF\_COUNT”, “P\_VARIABLE\_NAMES”, “P\_VARIABLE\_COUNT”, “P\_BUTON\_LABEL\_NAMES”, “P\_BUTON\_LABEL\_COUNT” ve “P\_SIZE\_FACTOR” olmak üzere toplam altı tane parametre almaktadır. Kod dosyaları için katsayı ağırlık bilgilerinin

tutulduğu “Tbl\_Factors4Code” isimli tabloda kayıt olmaması durumunda alınan ağırlık bilgileri tabloya kaydedilmekte, tabloda mevcut ağırlıkların var olması durumunda ise tablodaki eski değerler prosedüre gelen ve kullanıcının girmiş olduğu yeni ağırlık bilgileri ile değiştirilmektedir.

```

CREATE PROCEDURE [dbo].[P_INSUP_FACTORS4CODE]
(
    @P_FOR_IF_COUNT          varchar(100),
    @P_VARIABLE_NAMES       varchar(100),
    @P_VARIABLE_COUNT       varchar(100),
    @P_BUTON_LABEL_NAMES    varchar(100),
    @P_BUTON_LABEL_COUNT    varchar(100),
    @P_SIZE_FACTOR          varchar(100)
)
AS
DECLARE @V_COUNT INTEGER;
BEGIN
SELECT @V_COUNT = COUNT(*) FROM Tbl_Factors4Code;

IF @V_COUNT=0 OR @V_COUNT IS NULL
BEGIN
    INSERT INTO Tbl_Factors4Code
    (
        FOR_IF_COUNT,
        VARIABLE_NAMES,
        VARIABLE_COUNT,
        BUTON_LABEL_NAMES,
        BUTON_LABEL_COUNT,
        SIZE_FACTOR
    )
    values
    (
        @P_FOR_IF_COUNT,
        @P_VARIABLE_NAMES,
        @P_VARIABLE_COUNT,
        @P_BUTON_LABEL_NAMES,
        @P_BUTON_LABEL_COUNT,
        @P_SIZE_FACTOR
    );
END
ELSE
BEGIN

UPDATE Tbl_Factors4Code
SET FOR_IF_COUNT          =@P_FOR_IF_COUNT
, VARIABLE_NAMES         =@P_VARIABLE_NAMES
, VARIABLE_COUNT         =@P_VARIABLE_COUNT
, BUTON_LABEL_NAMES      =@P_BUTON_LABEL_NAMES
, BUTON_LABEL_COUNT      =@P_BUTON_LABEL_COUNT
, SIZE_FACTOR            =@P_SIZE_FACTOR

END
END

```

Şekil 4.22. Kod dosyaları için katsayı ağırlık bilgisini kaydeden prosedür

#### 4.2.9. Metin Belgeleri için Katsayı Ağırlık Bilgilerini Getiren Prosedür

Aşağıdaki şekilde “P\_SELECT\_FACTORS” isimli prosedür yer almaktadır. Bu prosedür Şekil 3.7’deki katsayı seçim ekranı açıldığında metin belgeleri için daha önceden belirlenmiş olan katsayı ağırlık bilgilerinin ekranda kullanıcıya gösterilmesi amacıyla kullanılmaktadır. Prosedür bilgileri getirmek için metin belgeleri için katsayı ağırlık bilgilerinin tutulduğu “Tbl\_Factors” isimli tabloyu kullanmaktadır.

```
CREATE PROCEDURE [dbo].[P_SELECT_FACTORS]
AS
BEGIN
SELECT * FROM Tbl_Factors;
END
```

Şekil 4.23. Metin belgeleri için katsayı ağırlık bilgisini getiren prosedür

#### 4.2.10. Kod Dosyaları için Katsayı Ağırlık Bilgilerini Getiren Prosedür

Aşağıdaki şekilde “P\_SELECT\_FACTORS4CODE” isimli prosedür yer almaktadır. Bu prosedür Şekil 3.21’deki kod dosyaları için katsayı seçim ekranı açıldığında kod dosyaları için daha önceden belirlenmiş olan katsayı ağırlık bilgilerinin ekranda kullanıcıya gösterilmesi amacıyla kullanılmaktadır. Prosedür bilgileri getirmek için kod dosyaları için katsayı ağırlık bilgilerinin tutulduğu “Tbl\_Factors4Code” isimli tabloyu kullanmaktadır.

```
CREATE PROCEDURE [dbo].[P_SELECT_FACTORS4CODE]
AS
BEGIN
SELECT * FROM Tbl_Factors4Code;
END
```

Şekil 4.24. Kod dosyaları için katsayı ağırlık bilgisini getiren prosedür

#### 4.2.11. Boyut ve Yazar Bilgilerini Getiren Prosedür

Aşağıdaki şekilde “P\_SELECT\_FILES” isimli prosedür yer almaktadır. Bu prosedür karşılaştırma işlemi yapılan metin belgeleri ve kod dosyalarına ait boyut ve yazar bilgilerinin getirilmesi amacıyla kullanılmaktadır. Prosedür input olarak varchar tipinde “P\_FILE\_NAME” ve “P\_FILE\_TYPE” isimli iki parametre almaktadır.

Prosedür bilgileri getirmek için dosyalara ait isim bilgileri, tür bilgileri, boyut ve yazar bilgilerinin tutulduğu “Tbl\_Files” isimli tabloyu kullanmaktadır.

```
CREATE PROCEDURE [dbo].[P_SELECT_FILES]
(
  @P_FILE_NAME    varchar(100)=NULL,
  @P_FILE_TYPE    varchar(100)
)
AS
BEGIN
  SELECT Name, Type, Size, Writer
  FROM   Tbl_Files
  WHERE  (Name=@P_FILE_NAME or @P_FILE_NAME is null)
        AND (Type=@P_FILE_TYPE or @P_FILE_TYPE is null) ;
END
```

Şekil 4.25. Boyut ve yazar bilgilerini getiren prosedür

#### 4.2.12. Kod Dosyalarına Ait Bilgileri Getiren Prosedür

Aşağıdaki şekilde “P\_SELECT\_PROJECT\_VARIABLE” isimli prosedür yer almaktadır. Bu prosedür kod dosyalarına ait proje adı, sayfa adı, değişken tipi ve değişken adı bilgilerinin getirilmesi amacıyla kullanılmaktadır. Prosedür input olarak varchar tipinde “P\_PROJECT\_NAME”, ”P\_PAGE\_NAME” ve “P\_VARIABLE\_TYPE” isimli üç parametre almaktadır. Prosedür bilgileri getirirken kod dosyaları için proje adı, sayfa adı, değişken tipi ve değişken adı bilgilerinin tutulduğu “Table\_Project\_Variable” isimli tablo kullanılmaktadır.

```

CREATE PROCEDURE [dbo].[P_SELECT_PROJECT_VARIABLE]
(
    @P_PROJECT_NAME      varchar(100),
    @P_PAGE_NAME         varchar(100),
    @P_VARIABLE_TYPE     varchar(100)
)
AS
BEGIN
SELECT *
    FROM Table_Project_Variable
    WHERE (Project_Name =@P_PROJECT_NAME or @P_PROJECT_NAME is null)
        AND (Page_Name =@P_PAGE_NAME or @P_PAGE_NAME is null)
        AND (Variable_Type =@P_VARIABLE_TYPE or @P_VARIABLE_TYPE is null) ;
END

```

Şekil 4.26. Kod dosyalarına ait bilgileri getiren prosedür

#### 4.2.13. Metin Belgelerinde Kullanılan Cümle Bilgilerini Getiren Prosedür

Aşağıdaki şekilde “P\_SELECT\_SENTENCES” isimli prosedür yer almaktadır. Bu prosedür metin belgelerinde kullanılan cümleler ve kullanım sayısı bilgilerinin getirilmesi amacıyla kullanılmaktadır. Prosedür input olarak varchar tipinde “P\_DOCUMENT\_NAME” ve “P\_DOCUMENT\_TYPE” isimli iki parametre almaktadır. Prosedür bilgileri getirirken metin belgelerinde kullanılan cümleler ve kullanım sayısı bilgilerinin tutulduğu “Tbl\_Sentences” isimli tablo kullanılmaktadır.

```

CREATE PROCEDURE [dbo].[P_SELECT_SENTENCES]
(
    @P_DOCUMENT_NAME     varchar(100),
    @P_DOCUMENT_TYPE     varchar(100)
)
AS
BEGIN
    SELECT SENTENCE, COUNTS
        FROM Tbl_Sentences
        WHERE DOCUMENT_NAME =@P_DOCUMENT_NAME AND DOCUMENT_TYPE=@P_DOCUMENT_TYPE ;
END

```

Şekil 4.27. Metin belgelerinde kullanılan cümle bilgilerini getiren prosedür

#### 4.2.14. Kod Dosyalarında Kullanılan Değişken Bilgilerini Getiren Prosedür

Aşağıdaki şekilde “P\_SELECT\_STRUCTURAL\_VARIABLE” isimli prosedür yer almaktadır. Bu prosedür kod dosyalarında kullanılan değişken tipleri ve kullanım sayısı bilgilerinin getirilmesi amacıyla kullanılmaktadır. Prosedür input olarak varchar tipinde “P\_PROJECT\_NAME”, “P\_PAGE\_NAME” ve “P\_STRUCTURAL\_TYPE” isimli üç parametre almaktadır. Prosedür bilgileri



getirirken kod dosyalarında kullanılan değişken tipleri ve kullanım sayısı bilgilerinin tutulduğu “Table\_Structural\_Count” isimli tablo kullanılmaktadır.

```
CREATE PROCEDURE [dbo].[P_SELECT_STRUCTURAL_VARIABLE]
(
    @P_PROJECT_NAME      varchar(100),
    @P_PAGE_NAME         varchar(100),
    @P_STRUCTURAL_TYPE   varchar(100)
)
AS
BEGIN
    SELECT *
    FROM Table_Structural_Count
    WHERE      (Project_Name      =@P_PROJECT_NAME      or @P_PROJECT_NAME is null)
              AND (Page_Name      =@P_PAGE_NAME        or @P_PAGE_NAME is null)
              AND (Structural_Type =@P_STRUCTURAL_TYPE or @P_STRUCTURAL_TYPE is null) ;
END
```

Şekil 4.28. Kod dosyalarında kullanılan değişken bilgilerini getiren prosedür

#### 4.2.15. Metin Belgelerinde Kullanılan Kelime Bilgilerini Getiren Prosedür

Aşağıdaki şekilde “P\_SELECT\_WORDS” isimli prosedür yer almaktadır. Bu prosedür metin belgelerinde kullanılan kelimeler ve kullanım sayısı bilgilerinin getirilmesi amacıyla kullanılmaktadır. Prosedür input olarak varchar tipinde “P\_DOCUMENT\_NAME” ve “P\_DOCUMENT\_TYPE” isimli iki parametre almaktadır. Prosedür bilgileri getirirken metin belgelerinde kullanılan kelimeler ve kullanım sayısı bilgilerinin tutulduğu “Tbl\_Words” isimli tablo kullanılmaktadır.

```
CREATE PROCEDURE [dbo].[P_SELECT_WORDS]
(
    @P_DOCUMENT_NAME     varchar(100),
    @P_DOCUMENT_TYPE     varchar(100)
)
AS
BEGIN
    SELECT WORD,COUNTS
    FROM Tbl_Words
    WHERE DOCUMENT_NAME =@P_DOCUMENT_NAME AND DOCUMENT_TYPE=@P_DOCUMENT_TYPE ;
END
```

Şekil 4.29. Metin belgelerinde kullanılan kelime bilgilerini getiren prosedür

#### 4.2.16. Filtreleme Kombosundaki Bilgileri Getiren Prosedür

Aşağıdaki şekilde “P\_SELECT\_FILES4COMBO” isimli prosedür yer almaktadır. Bu prosedür karşılaştırma sonuçlarının gösterildiği Şekil 3.10’da görülen sonuç gösterim ekranında yer alan ve ekranda gösterilen sonuç bilgilerinin filtrelenmesini sağlayan kombonun doldurulması amacıyla kullanılmaktadır. Prosedür input olarak varchar tipinde “P\_TYPE” isimli bir tane parametre almaktadır. Prosedür almış olduğu parametreye göre karşılaştırma işlemi yapılan dosyaların bulunduğu “Tbl\_Files” tablosundan istenilen tipteki dosya isimlerinin getirilmesini sağlar.

```
CREATE PROCEDURE [dbo].[P_SELECT_FILES4COMBO]
(
    @P_TYPE    varchar(100)
)
AS
BEGIN
SELECT Name as value
    FROM Tbl_Files WHERE Type=@P_TYPE;
END
```

Şekil 4.30. Filtreleme Kombosunu Dolduran Prosedür

#### 4.3. Sonuç

Bu bölümde, geliştirmiş olduğumuz ve üçüncü bölümde detayları anlatılmış olan uygulamanın kullandığı veritabanı tabloları, prosedürleri ve bunların kullanım amaçlarından bahsedilmiştir.

## **BÖLÜM 5. SONUÇ VE ÖNERİLER**

Birinci bölümde, konu ile ilgili temel kavramlar verilmiş, metin belgeleri ve kaynak kod dosyalarında yapılan intihal tespitinin öneminden bahsedilmiştir.

Bölüm 2’de, intihale neden olan unsurlar, kaynak kod klonlama yöntemleri ve tespit araçları anlatılmıştır. Ayrıca, bu bölümde metin belgesi ve kaynak kod dosyalarında yapılan intihalleri tespit etmek amacıyla geliştirmiş olduğumuz uygulamanın piyasada benzerlik tespiti yapmak amacıyla kullanılan diğer programlarla karşılaştırılması yapılmıştır.

Üçüncü bölüm çalışmanın ana bölümü olarak sayılabilir. Bu bölümde, uygulamanın çalışma mantığı ve kullanımından bahsedilmiş, uygulama örnek verilerle çalıştırılarak elde edilen sonuçların nasıl yorumlanması gerektiği hakkında detaylı bilgi verilmiştir.

Bölüm dördte, uygulamanın kullandığı veritabanı tablolarının yapısı anlatılmış ve bu tabloların kullanım amaçlarından bahsedilmiştir.

Beşinci bölümde ise uygulamanın kullandığı prosedürlerin detayları, kullanım amaçları ve bu prosedürlerin ne zaman çalıştırıldıkları hakkında bilgi verilmiştir.

Sonuç olarak bu çalışmada, metin belgesi ve kaynak kod dosyaları arasında benzerlik tespiti yapan uygulamalar incelenmiştir. Bu inceleme sonucunda elde edilen bilgiler doğrultusunda metin belgeleri ve kod dosyaları arasında benzerlik oranı hesaplayan, benzer olan kısımları gösteren, kullanıcıya sunduğu katsayı girişi sayesinde benzerlik oranı hesaplamada esneklik sağlayan, klasör tabanlı çalışma imkanı ile çok sayıda kod dosyası içeren klasörlerin birbirleri ile karşılaştırılması imkanını veren, sonuç gösterim ekranı ile kullanıcının sadece incelemek istediği klasörü seçmesine olanak

sağlayan, karşılaştırma sonuçlarının çıktısının alınabildiği ve ayrıca karşılaştırma işlemi yapılan dosyalarda ortak olan kısımların gösterildiği ekranda sonuçların filtrelenmesine imkan sağlayan web tabanlı uygulama geliştirilmiştir.

Geliştirilen uygulamada kaynak kod karşılaştırma işlemi, web tabanlı C# projeleri için incelenmiş olup istenildiği takdirde uygulama diğer diller içinde genişletilebilir.

## KAYNAKLAR

- [1] Özen, Z., Gülseçen, S., "Kaynak Kod Benzerliği ve Klon Kod Tespit Araçları" Akademik Bilişim Konferansları Bildirileri, 1-3 Şubat 2012.
- [2] Roy, C. K., Cordy, J.R., Koschke, R. , "Comparison and evaluation of code clone detection techniques and tools: A qualitative approach" Science of Computer Programming, (7) 74 , 470-495 , 2009.
- [3] Higo, Y., Kamiya, T., Kusumoto, S., Inoue K. , "Method and implementation for investigating code clones in a software system" Information and Software Technology, 49 , 985–998 , 2007.
- [4] Yang, W., "Identifying Syntactic Differences Between Two Programs" Software - Practice and Experience, (7) 21, 739-755, 1991.
- [5] Chilowicz, M., Duris, E., Roussel, G., "Finding Similarities in Source Code Through Factorization" Electronic Notes in Theoretical Computer Science, 238, 47–62 , 2009.
- [6] Bettenburg, N., Shang, W., Ibrahim, W. M., Adams, B., Zou, Y., Hassan, A. E., "An empirical study on inconsistent changes to code clones at the release level" Science of Computer Programming, (6) 77, 760-776, 2012.
- [7] Arwin, C., Tahaghoghi, S.M.M., "Plagiarism Detection across Programming Languages" ACSC '06 Proceedings of the 29th Australasian Computer Science Conference, 48, 277-286, 2006.
- [8] Ducasse, S., Rieger, M., Demeyer, S., "A Language Independent Approach for Detecting Duplicated Code" ICSM '99 Proceedings of the IEEE International Conference on Software Maintenance, 109, 1999.
- [9] Balint, M., Girba, T., Marinescu, R., "How Developers Copy" In Proceedings of International Conference on Program Comprehension, 56–65, 2006.
- [10] Pape, S.J., "Compare word problems: Consistency hypothesis revisited"

- Contemporary Educational Psychology, 28, 396–421, 2003.
- [11] Djuric, Z., Gasevic D., “A Source Code Similarity System for Plagiarism Detection” The Computer Journal, 2012.
- [12] Roy, C.K., Cordy J.R., “Scenario-Based Comparison of Clone Detection Technique” ICPC '08 Proceedings of the 16th IEEE International Conference on Program Comprehension, 153-162, 2008.
- [13] Calefato, F., Lanubile, F., Mallardo, T., “Function Clone Detection in Web Applications: A Semiautomated Approach” Journal of Web Engineering, (1) 3, 3-21, 2004.
- [14] Kamiya, T., “Interoperation Potential: Integration of Code-Clone Detection Methods with Other Analysis Methods” Workshop proceedings of the 13th European Conference on Software Maintenance and Reengineering , 3-6, 2009.
- [15] Roy, C.K., Cordy, J.R., “A Survey on Software Clone Detection Research” School of Computing, Queen's University, Canada, 2007.
- [16] Uçak, N. Ö., Birinci, H. G., “Bilimsel etik ve intihal” Türk Kütüphaneciliği, (2) 22, 187-204, 2008.
- [17] Van De Vanter, M. L., “The documentary structure of source code” Information and Software Technology, 44 , 767–782, 2002.
- [18] Baker, B.S., "On Finding Duplication and Near-Duplication in Large Software Systems" Proceedings of 2nd Working Conference on Reverse Engineering, 86 – 95, 1995.
- [19] <http://theory.stanford.edu/~aiken/moss/>
- [20] <https://jplag.ipd.kit.edu>
- [21] [www.ccfinder.net](http://www.ccfinder.net)
- [22] [sydney.edu.au/engineering/it/~scilect/sherlock](http://sydney.edu.au/engineering/it/~scilect/sherlock)
- [23] [ftp://ftp.cs.vu.nl/pub/dick/similarity\\_tester/](ftp://ftp.cs.vu.nl/pub/dick/similarity_tester/)
- [24] [www.harukizaemon.com/simian/](http://www.harukizaemon.com/simian/)
- [25] <http://pmd.sourceforge.net/cpd.html>
- [26] <http://duplo.sourceforge.net/>
- [27] <http://www.cs.hut.fi/Software/Plaggie/download.html>

## ÖZGEÇMİŞ

Uğur ŞİMŞEK, 30.12.1988'de Sakarya'da doğdu. İlk, orta ve lise eğitimini Sakarya'da tamamladı. 2007 yılında Sakarya Anadolu Lisesi'nden mezun oldu. Aynı yıl başladığı Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümü'nden 2011 yılında bölüm üçüncüsü olarak mezun oldu. 2009 – 2012 yılları arasında Ford Otosan A.Ş.'nde bilgisayar mühendisi olarak görev aldı. Bu süre içerisinde geliştirilen birçok web tabanlı projelerin yanı sıra müşteri ilişkileri yönetimi projesinde aktif rol aldı. 2012 yılında Ford Otosan A.Ş.'den ayrılarak Finansbank-IBTECH A.Ş.'nde çalışmaya başladı. Şu anda aynı iş yerinde bilgisayar mühendisi olarak çalışmalarına devam etmektedir.