

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/228446264>

# Kaynak Kod Benzerliđi Ve Klon Kod Tespit Araçları

## Article

CITATIONS

2

READS

1,073

7 authors, including:



[Zeki Ozen](#)

Istanbul University

44 PUBLICATIONS 31 CITATIONS

[SEE PROFILE](#)



[S. Gulsecen](#)

Istanbul University

55 PUBLICATIONS 111 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Thesis [View project](#)



A proposal of decision support model for Make To Order (MTO) Environments [View project](#)

# Kaynak Kod Benzerliği Ve Klon Kod Tespit Araçları

Zeki ÖZEN<sup>1</sup>, Sevinç GÜLSEÇEN<sup>2</sup>

<sup>1</sup> İstanbul Üniversitesi, Enformatik Bölümü, İstanbul

<sup>2</sup> İstanbul Üniversitesi, Fen Fakültesi Matematik Bölümü, İstanbul

zekiozen@istanbul.edu.tr, gulseecen@istanbul.edu.tr

**Özet:** Kaynak kodların intihali yazılım sektöründe faaliyet gösteren şirketlerde maddi zarara sebep olmakta ve Fikir ve Sanat Eserleri Kanunu'na göre yazılımların eser sahibi olan yazılım geliştiricilerinin manevi haklarına tecavüz oluşturmaktadır. Bu nedenle kaynak kodlarda intihalin tespiti hem hukuki açıdan ve hem de teknik açıdan önemli bir konudur. Bu çalışmada kaynak kodlarda intihal tespiti ve klon kod benzerlik tespiti yapan araçlar tanıtılarak kullanımlarına ilişkin bilgiler verilecektir.

**Anahtar Sözcükler:** Kaynak kod intihali, klon kod, kaynak kod benzerliği, kaynak kod benzerlik tespit araçları

## Source Code Similarity and Clone code Detection Tools

**Abstract:** Source code plagiarism has been causing economical damage for software companies and according to Law on Intellectual and Artistic Works, source code plagiarism has been infringement to moral rights of the software developers who are the owners of the product of the software. Therefore the detection of plagiarism on source codes is an important subject both in terms of legal and technical aspect. In this study, the tools that detect plagiarism on source codes and clone code similarity will be presented and information about the usage of these tools will be given.

**Keywords:** Source Code Plagiarism, Clone Code, Source Code Similarity, Source Code Similarity Detection Tools

### 1. Giriş

Bilgisayar programının programcı tarafından bir programlama diliyle yazılan formuna kaynak kod denir [26]. Kaynak kod (aynı zamanda kaynak, kod ya da kaynak program da denir) aslen bir insan tarafından düz metin şeklinde (okunabilir alfanumerik karakterlerle) ve bir programlama diliyle yazılmış ve o dile ait komutlar, ifadeler ve bunların komut setlerini barındıran yazılımın bir biçimidir [24].

Kaynak kodlar, Fikir ve Sanat Eserleri Kanunu'na göre bilgisayar programlarının korunan unsurları arasındadır [2].

Kanunun diğer eserlere sağladığı koruma kaynak kodlar için de geçerlidir ve Kanunun eser sahiplerine verdiği hakların tamamı kaynak kodun yazarlarına da aynen verilmiştir.

Kaynak kod intihali, kaynak kod dosyalarının veya kaynak kod kısımlarının asıl eser sahibinin izni olmaksızın kopyalanarak veya değiştirilerek başka bir yazılım üretmek amacıyla kullanılmasıdır [18].

Kaynak kodların intihali ve başka bir programda kullanımı asıl eser sahibinin malî ve manevî haklarının ihlâl edilmesine yol açar [7].

Kaynak kod intihalinin diğer intihal türlerinden farkı; 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 [18].

Kaynak kod intihalinin gerçekleşmesine yol açan birçok neden olmakla birlikte, en bilinen yöntem kod klonlama olarak da adlandırılan kaynak kodun dosyalar arasında kopyalanmasıdır [18].

Klon kod, bir kod bölümünün başka bir kod bölümüyle eşdeğer (aynı) veya benzer olması şeklinde tarif edilir ve klon kodlar birbirlerinin kopyası olan iki veya daha fazla kod bölümlerini temsil eder [10, 25].

Daha önceki araştırmalar tipik yazılım sistemlerinin kaynak kodlarının %7 - %23 oranında klon kod içerdiğini göstermiştir [22]. Tablo 1’de bazı yazılımlardaki klon kod yüzdeleri gösterilmiştir.

Program Adı	Satır Sayısı (bin)	Klonlama Oranı (%)
Linux kernel	4.365	15
Java Development Kit (JDK) 1.4.2	2.418	8
JDK 1.3.0	570	9
Process-Control System	400	12
JHotDraw 7.0.7	71	19

**Tablo 1 :** Bazı yazılımlarda klon kod bulunma yüzdesi [25]

Kaynak kodlar arasında intihal olduğunun tespiti için birbirine benzeyen kaynak kodların karşılaştırılması gerekmektedir. Unutulmamalıdır ki, birbirine benzeyen her kod intihal edilmiş kod değildir. Cosma [6] benzerlik ve intihalin birbirine karıştırılmaması gerektiğini, kaynak kod dosyaları arasında benzerlik olsa bile, intihalin ancak benzerliğin dikkatlice incelenmesi sonucunda saptanabileceğini belirtmiştir. İntihalden şüphelenilen durumlarda kaynak kod dosyaları arasında belirgin bir şekilde programın mantığı, yaklaşımı ve işlevselliği benzeşir ve bu tür bir benzerlik intihalin olduğuna yönelik güçlü bir kanıt teşkil eder [6].

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.

Çalışmanın devamında bu amaçla geliştirilmiş yazılımların kullanımı hakkında bilgi verilmiştir.

## 2. Kaynak Kod Benzerlik ve Klon Kod Tespit Araçları

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.

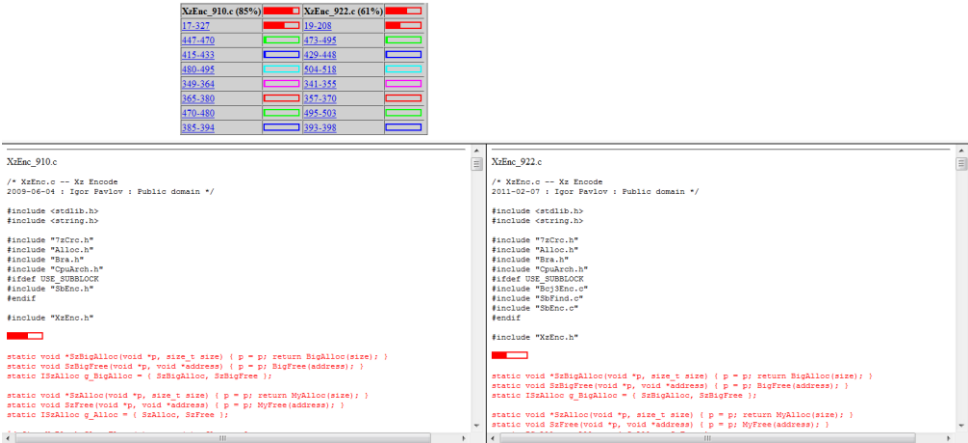
## 2.1 MOSS

Web tabanlı kaynak kod karşılaştırması yapan MOSS (Measure of Software Similarity), kaynak kod benzerliği tespitinde kullanılan en popüler araçlardan biridir [9]. MOSS'un kullanımı üyelik gerektirmekte olup, üyelik işlemi ücretsizdir. 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 yapılabilmektedir.

Örnek kullanımı aşağıdaki gibi olan MOSS benzerlik tespit aracı, karşılaştırma sonucu benzerlik oranını HTML sayfası olarak oluşturmakta ve karşılaştırılan kaynak kodları yan yana listelerek benzerlik olan kısımları farklı renklerle göstermektedir (Şekil 1).

MOSS örnek kullanımı:

*moss.pl -l java dosya1.java dosya2.java*



Şekil 1: MOSS ile karşılaştırılan iki kaynak kodun gösterilmesi

## 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 [14]. JPlag C, C++, Java, C#, Scheme ve doğal dillerde benzerlik tespiti yapılabilmektedir [21].

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 (Şekil 2).

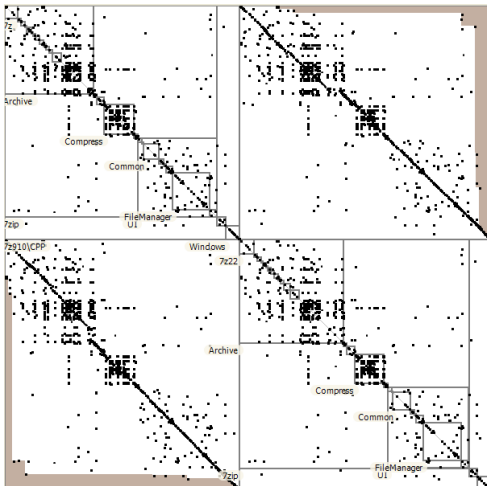


Şekil 2: JPlag ile karşılaştırılan kaynak kodların gösterilmesi

### 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 [13]. Java, C/C++, C#, Visual Basic (VB) ve COBOL dillerinde karşılaştırma yapabilen CCFinder, farklı kod satırlarındaki farklı isimlendirmelere sahip klonları bulabilmektedir [3, 13].

CCFinder benzerlik sonuçlarını çizim üzerinde noktalama yöntemi (*dotplotting*) kullanarak görselleştirir (Şekil 3) [5].



Şekil 3: CCFinder benzerlik sonucu scatterplot üzerinde göstermektedir.

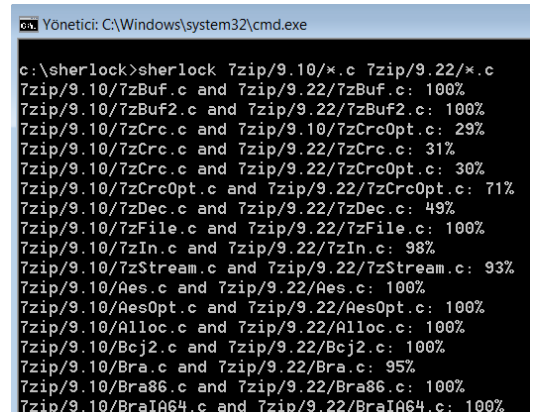
### 2.4 Sherlock

Sherlock, kaynak kodlarda simgeleştirme yöntemini kullanarak metinlerde ise string eşleme yöntemini kullanarak benzerlik tespiti yapan açık kaynak kodlu bir araçtır [12, 8]. Sherlock C, C++ Java, Pascal, HTML ve metin dosyaları arasındaki benzerlik oranlarını vermektedir [8].

Sherlock, kaynak kodda kelimelerin dizilimine bakarak hash algoritması ile dijital sayı dizileri üretmekte ve bunları karşılaştırmaktadır [12].

Temel olarak aşağıdaki yapıda kullanılan Sherlock'un ekran görüntüsü Şekil 4'te verilmiştir:

*sherlock* [-parametre değer]  
karşılaştırılacak\_dosya1  
karşılaştırılacak\_dosya2



Şekil 4: Scherlock örnek kullanımı

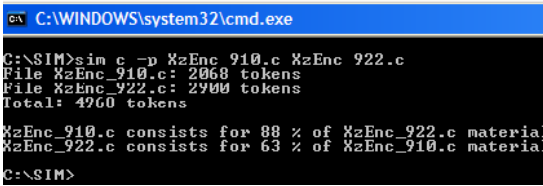
## 2.5 SIM

SIM C, Java, Pascal, Modula-2, Lisp, Miranda ve metin dosyalarında benzerlik tespiti yapan bir araçtır. [ftp://ftp.cs.vu.nl/pub/dick/similarity\\_tester/](ftp://ftp.cs.vu.nl/pub/dick/similarity_tester/) adresinden indirilebilen ve ücretsiz bir yazılım olan SIM isim değişikliklerini ve program bloklarının yer değişimini fark edebilmekte, boşlukları ise dikkate almamaktadır [4].

Temel kullanımı aşağıdaki gibi olan SIM benzerlik tespit aracının örnek kullanım ekranı Şekil 5'te verilmiştir.

Temel kullanım:

*sim\_c* *karşılaştırılacak\_dosya1*  
*karşılaştırılacak\_dosya2*



```
C:\WINDOWS\system32\cmd.exe
C:\SIM>sim c -p XzEnc_910.c XzEnc_922.c
File XzEnc_910.c: 2068 tokens
File XzEnc_922.c: 2700 tokens
Total: 4260 tokens

XzEnc_910.c consists for 88 % of XzEnc_922.c material
XzEnc_922.c consists for 63 % of XzEnc_910.c material
C:\SIM>
```

**Şekil 5:** SIM ile iki C kaynak kodunun karşılaştırılması

## 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 kısmî olarak karşılaştırma yapan Simian (Similarity Analyser) metin tabanlı karşılaştırma yapan kaynak kod benzerlik tespit aracıdır [23, 19].

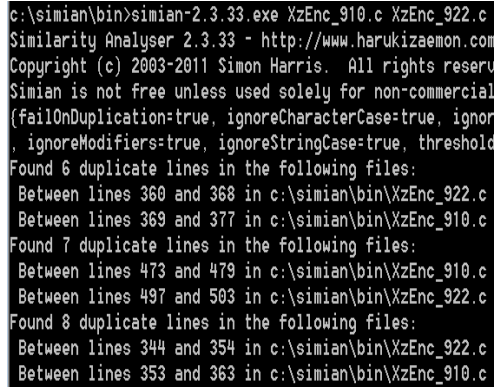
Simian, karşılaştırılan kodların benzerlik oranlarını vermemekte, sadece karşılaştırılan kaynak kod dosyalarında birbirinden kopyalanmış satırları göstermektedir [16]. Simian sadece dosya bazlı değil klasör tabanlı karşılaştırma da yapabilmektedir.

İki C kaynak kod dosyasının karşılaştırılmasına ilişkin örnek Simian

kullanımı ve karşılaştırma sonrası oluşan ekran görüntüsü Şekil 6'da gösterilmiştir.

Temel kullanım:

*simian karşılaştırılacak\_dosya1*  
*karşılaştırılacak\_dosya2*



```
c:\simian\bin>simian-2.3.33.exe XzEnc_910.c XzEnc_922.c
Similarity Analyser 2.3.33 - http://www.harukizaemon.com
Copyright (c) 2003-2011 Simon Harris. All rights reserved.
Simian is not free unless used solely for non-commercial
(failOnDuplication=true, ignoreCharacterCase=true, ignore
, ignoreModifiers=true, ignoreStringCase=true, threshold
Found 6 duplicate lines in the following files:
Between lines 360 and 368 in c:\simian\bin\XzEnc_922.c
Between lines 369 and 377 in c:\simian\bin\XzEnc_910.c
Found 7 duplicate lines in the following files:
Between lines 473 and 479 in c:\simian\bin\XzEnc_910.c
Between lines 497 and 503 in c:\simian\bin\XzEnc_922.c
Found 8 duplicate lines in the following files:
Between lines 344 and 354 in c:\simian\bin\XzEnc_922.c
Between lines 353 and 363 in c:\simian\bin\XzEnc_910.c
```

**Şekil 6:** Simian ile iki C kaynak kodunun karşılaştırılmasına ilişkin ekran görüntüsü

## 2.7 CPD

CPD (Copy/Paste Dedector), <http://pmd.sourceforge.net/cpd.html> adresinden indirilebilen ve PMD projesine bütünlük simge tabanlı klon kod tespiti yapabilen ve son versiyonu “Karp-Rabin” metin eşleme algoritmasını kullanan açık kaynak bir yazılımdır [20, 22]. 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 (Şekil 7) [16]. BSD lisansı ile yayımlanan CPD aralarında Eclipse, NetBeans, JDeveloper, JEdit, BlueJ ve IntelliJ IDEA gibi birçok IDE'ye eklenti olarak da kurulabilmektedir.

Plaggie, karşılaştırma işlemini kurulduğu bilgisayarda yapmakta ve benzerlik sonuçlarını HTML sayfası şeklinde kullanıcıya sunmaktadır (Şekil 9).

## Detection results for /home/zekzen/HoDoKu/HoDoKu2.0/Sudoku.java and /home/zekzen/HoDoKu/HoDoKu1.0/Sudoku.java

Similarity % 79,7  
Similarity A % 82,6  
Similarity B % 96,5

Similarity distribution of /home/zekzen/HoDoKu/HoDoKu2.0/Sudoku.java

Similarity distribution of /home/zekzen/HoDoKu/HoDoKu1.0/Sudoku.java

/home/zekzen/HoDoKu/HoDoKu2.0/Sudoku.java % 82,6 | Tokens /home/zekzen/HoDoKu/HoDoKu1.0/Sudoku.java % 96,5 | Tokens Matched lines

```
9: 0(20(6601)-25110(14224))...0(20(6601)-25110(14224))|left:right:token_left:token_right
10: 441126(51131)-58134(5422))...291107(4251)-401115(6566))|left:right:token_left:token_right
7: 991196(54321)-931159(6642))...451125(4763)-79148(5832))|left:right:token_left:token_right
8: 991143(66261)-121179(7541)...401115(5988)-156167(6543))|left:right:token_left:token_right
15: 1231181(71851)-1271185(72651)...1071108(85191)-1111179(6696))|left:right:token_left:token_right
10: public Sudoku() {
10: // allowed positions
10: for (int i = 0; i <= 9; i++) {
10: allowedPositions[i] = new SudokuSet(true);
10: allowedPositions[i].setAll(i);
10: possiblePositions[i] = new SudokuSet(true);
10: possiblePositions[i].setAll(i);
10: positions[i] = new SudokuSet(i);
10: }
10: setSudoku(null);
10: }
10: // override
10: public Sudoku clone() {
10: try {
10: newSudoku = (Sudoku) super.clone();
10: }
```

```
10: public Sudoku(boolean initTemplates) {
10: // allowed positions
10: for (int i = 0; i <= 9; i++) {
10: allowedPositions[i] = new SudokuSet(true);
10: allowedPositions[i].setAll(i);
10: possiblePositions[i] = new SudokuSet(true);
10: possiblePositions[i].setAll(i);
10: positions[i] = new SudokuSet(i);
10: }
10: // Buddies und Unit-Sets initialisieren
10: initBuddies();
10: // Templates initialisieren
10: if (initTemplates && templates[0] == null) {
10: initTemplates();
10: }
```

Şekil 9: Plagie karşılaştırma sonuçlarını görüntüleme ekranı

### 3. Sonuç ve Öneriler

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 [11, 15]. Böylece yazılımın bakımı kolaylaşmakta ve bakım maliyetleri de düşmektedir. 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 [22].

Hukuksal anlamda ise benzerlik tespit araçları kaynak kodlardaki intihali tespit etmek amacıyla kullanılmaktadır.

Kanun tarafından koruma sağlanan kaynak kodların kopyalanması veya intihal edilmesi

durumunda, eser sahibi olan bilgisayar programcısı diğer eser sahiplerine tanınan hakların tamamına sahiptir. İntihalin varlığı veya intihal şüphesi durumunda eser sahibi mahkemeden intihalin tespitini isteyebilir. Mahkemelerin intihali tespit amacıyla görevlendirdiği bilirkişilerin bu araçları kullanması ile intihalin varlığına ilişkin daha sağlıklı sonuçlar alınabilmektedir.

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

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

### 4. Teşekkür

Bu çalışma İstanbul Üniversitesi Bilimsel Araştırma Projeleri Yürütücü Sekreterliğinin 13472 numaralı projesi ile desteklenmiştir.



## 5. Kaynaklar

- [1] Ahtiainen, A., Surakka, S. ve Rahikainen, M. 2006. Plaggie: GNU-licensed source code plagiarism detection engine for Java exercises. *Proceedings of the 6th Baltic Sea conference on Computing education research: Koli Calling 2006* (New York, NY, USA, 2006), 141–142.
- [2] Aksu, M. 2006. *Bilgisayar programlarının fikrî mülkiyet hukukunda korunması*. Beta yayınları, İstanbul, 1. Basım, ISBN 9789752955127.
- [3] The archive of CCFinder Official Site: 2010. <http://www.ccfinder.net/ccfinderxos.html>. Erişim: 2011-12-06.
- [4] Arwin, C. ve Tahaghoghi, S.M.M. 2006. Plagiarism detection across programming languages. *Proceedings of the 29th Australasian Computer Science Conference - Volume 48* (Darlinghurst, Australia, Australia, 2006), 277–286.
- [5] Burd, E. ve Bailey, J. 2002. Evaluating Clone Detection Tools for Use during Preventative Maintenance. *Proceedings of the Second IEEE International Workshop on Source Code Analysis and Manipulation* (Washington, USA, 2002), 36-43.
- [6] Cosma, G. 2008. *An Approach to Source-Code Plagiarism Detection and Investigation Using Latent Semantic Analysis*. University of Warwick.
- [7] Dalyan, Ş. 2008. *Bilgisayar Programlarının Fikrî Hukukta Korunması*. Ankara Üniversitesi Sosyal Bilimler Enstitüsü.
- [8] Goel, S. ve Rao, D. 2005. *Plagiarism and its Detection in Programming Languages*. Department of Computer Science and Information Technology.
- [9] Hage, J., Rademaker, P. ve Vugt, N. van 2010. *A comparison of plagiarism detection tools*, Technical Report, #UU-CS-2010-015. Department of Information and Computing Sciences, Utrecht University.
- [10] Higo, Y. 2006. *Code Clone Analysis Methods for Efficient Software Maintenance*. Osaka University.
- [11] Jafar, Y.M. 2007. *Clone Detection Using Pictorial Similarity in Slice Traces*. Kings College London.
- [12] Joy, M. ve Luck, M. 1999. Plagiarism in programming assignments. *IEEE Transactions on Education*. 42, 2 (May. 1999), 129-133.
- [13] Kamiya, T., Kusumoto, S. ve Inoue, K. 2002. CCFinder: a multilinguistic token-based code clone detection system for large scale source code. *IEEE Transactions on Software Engineering*. 28, 7 (Tem. 2002), 654- 670.
- [14] Kustanto, C. ve Liem, I. 2009. Automatic Source Code Plagiarism Detection. *Proceedings of the 2009 10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing* (Washington, 2009), 481–486.
- [15] Lozano, A., Wermelinger, M. ve Nuseibeh, B. 2007. Evaluating the Harmfulness of Cloning: A Change Based Experiment. *Proceedings of the Fourth International Workshop on Mining Software Repositories* (Washington, DC, USA, 2007).
- [16] Mishne, G. ve De Rijke, M. 2004. Source Code Retrieval using Conceptual Similarity. *PROC. 2004 CONF. COMPUTER ASSISTED INFORMATION RETRIEVAL (RIAO '04*. 1984 (2004), 539--554.
- [17] Mozgovoy, M. 2007. *Enhancing Computer-Aided Plagiarism Detection*. University Of Joensuu.
- [18] Özen, Z. 2012. *Bilişim Hukukunda Kaynak Kod İntihali*. İstanbul Üniversitesi Fen Bilimleri Enstitüsü.
- [19] Pate, J.R., Tairas, R. ve Kraft, N.A. *Clone evolution: a systematic review*. Technical Report #SERG-2010-01.R2.
- [20] PMD - Finding copied and pasted code: 2011. <http://pmd.sourceforge.net/cpd.html>. Accessed: 2011-12-08.

- [21] Prechelt, L., Malpohl, G. ve Philippsen, M. 2000. *JPlag: Finding plagiarisms among a set of programs*. University of Karlsruhe, Department of Informatics.
- [22] Roy, C.K., Cordy, J.R. ve Koschke, R. 2009. Comparison and evaluation of code clone detection techniques and tools: A qualitative approach. *Science of Computer Programming*. 74, 7 (2009), 470-495.
- [23] Simian - Similarity Analyser | Duplicate Code Detection for the Enterprise | Features: 2011. <http://www.harukizaemon.com/simian/features.html>. Accessed: 2011-12-05.
- [24] Source Code Definition: 2006. [http://www.linfo.org/source\\_code.html](http://www.linfo.org/source_code.html). Accessed: 2011-05-14.
- [25] TAIRAS, R.A. 2010. *REPRESENTATION, ANALYSIS, AND REFACTORING TECHNIQUES TO SUPPORT CODE CLONE MAINTENANCE*. The University of Alabama at Birmingham.
- [26] Yadav, A. 2008. *Microprocessor 8085, 8086*. University Science Press/Laxmi Publications Pvt. Ltd.