

## SKRIPSI

PEMBANGKIT TIMELAPSE PENGEMBANGAN PROYEK  
PERANGKAT LUNAK BERBASIS WEB



Billy Adiwijaya

NPM: 2015730053

PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS  
UNIVERSITAS KATOLIK PARAHYANGAN  
2019



**UNDERGRADUATE THESIS**

**TIMELAPSE GENERATOR FOR WEB-BASED SOFTWARE  
PROJECT DEVELOPMENT**



**Billy Adiwijaya**

**NPM: 2015730053**

**DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES  
PARAHYANGAN CATHOLIC UNIVERSITY  
2019**



## **ABSTRAK**

«Tuliskan abstrak anda di sini, dalam bahasa Indonesia»

**Kata-kata kunci:** «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Indonesia»



## **ABSTRACT**

«Tuliskan abstrak anda di sini, dalam bahasa Inggris»

**Keywords:** «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Inggris»



## DAFTAR ISI

<b>KATA PENGANTAR</b>	<b>xv</b>
<b>DAFTAR ISI</b>	<b>xvii</b>
<b>DAFTAR GAMBAR</b>	<b>xix</b>
<b>DAFTAR TABEL</b>	<b>xxi</b>
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Rumusan Masalah . . . . .	1
1.3 Tujuan . . . . .	2
1.4 Batasan Masalah . . . . .	2
1.5 Metodologi . . . . .	2
1.6 Sistematika Pembahasan . . . . .	2
<b>2 LANDASAN TEORI</b>	<b>5</b>
2.1 Git . . . . .	5
2.1.1 Version Control Systems . . . . .	5
2.1.2 Cara Kerja Git . . . . .	7
2.1.3 Operasi Dasar pada Git . . . . .	10
2.1.4 Git Checkout . . . . .	12
2.2 JGit . . . . .	12
2.2.1 Kelas Repository . . . . .	12
2.2.2 Kelas Ref . . . . .	13
2.2.3 Kelas FileRepository . . . . .	13
2.2.4 Kelas Git . . . . .	13
2.2.5 Kelas CheckoutCommand . . . . .	15
2.2.6 Kelas ListBranchCommand . . . . .	15
2.2.7 Enum ListBranchCommand.ListMode . . . . .	15
2.2.8 Kelas LogCommand . . . . .	15
2.2.9 Kelas ResetCommand . . . . .	15
2.2.10 Kelas ResetCommand.ResetType . . . . .	16
2.2.11 Kelas RevCommit . . . . .	16
2.2.12 Kelas PersonIdent . . . . .	17
2.3 Selenium WebDriver . . . . .	17
2.3.1 Interface WebDriver . . . . .	17
2.3.2 Interface WebDriver.Window . . . . .	18
2.3.3 Kelas ChromeDriver . . . . .	18
2.3.4 Kelas FirefoxDriver . . . . .	18
2.3.5 Kelas OperaDriver . . . . .	19
2.3.6 Kelas SafariDriver . . . . .	19
2.3.7 Kelas InternetExplorerDriver . . . . .	19

2.3.8	Kelas EdgeDriver . . . . .	19
2.3.9	Kelas HtmlUnitDriver . . . . .	19
2.3.10	Interface OutputType . . . . .	19
2.3.11	Interface TakesScreenshot . . . . .	20
2.4	Apache Commons CLI . . . . .	20
2.4.1	CommandLineParser . . . . .	20
2.4.2	DefaultParser . . . . .	20
2.4.3	CommandLine . . . . .	21
2.4.4	Options . . . . .	21
2.4.5	Option . . . . .	21
2.4.6	Option.Builder . . . . .	22
<b>3</b>	<b>ANALISIS</b>	<b>25</b>
3.1	Analisis Aplikasi Sejenis . . . . .	25
3.2	Analisis Penggunaan JGit dan Selenium WebDriver . . . . .	27
3.2.1	Analisis Penggunaan JGit . . . . .	27
3.2.2	Analisis Penggunaan Selenium WebDriver . . . . .	28
3.3	Analisis Fitur Aplikasi yang Dibangun . . . . .	29
3.4	Prapengujian Website Piktora . . . . .	32
3.4.1	Perbedaan Letak <i>File</i> . . . . .	34
3.4.2	Permasalahan Konfigurasi <i>Database</i> . . . . .	35
3.4.3	Permasalahan Migrasi <i>Database</i> . . . . .	35
<b>4</b>	<b>PERANCANGAN</b>	<b>37</b>
4.1	Perancangan Kelas . . . . .	37
4.2	Perancangan Antarmuka . . . . .	39
<b>5</b>	<b>IMPLEMENTASI DAN PENGUJIAN</b>	<b>43</b>
5.1	Implementasi . . . . .	43
5.1.1	Lingkungan Implementasi . . . . .	43
5.1.2	Hasil Implementasi . . . . .	43
5.2	Pengujian . . . . .	45
5.2.1	Pengujian Fungsional . . . . .	45
5.2.2	Pengujian Eksperimental . . . . .	49
<b>6</b>	<b>KESIMPULAN DAN SARAN</b>	<b>55</b>
6.1	Kesimpulan . . . . .	55
6.2	Saran . . . . .	55
<b>DAFTAR REFERENSI</b>		<b>57</b>
<b>A KODE PROGRAM</b>		<b>59</b>
<b>B LIBRARY UNTUK MEMBUAT FILE GIF</b>		<b>67</b>
<b>C KODE PROGRAM PROYEK PIKTORA PADA COMMIT 89000BE (12 JANUARI 2018)</b>		<b>71</b>

## DAFTAR GAMBAR

2.1 Local version control[1]. . . . .	5
2.2 Centralized version control[1]. . . . .	6
2.3 Distributed version control[1]. . . . .	7
2.4 Menyimpan data sebagai <i>snapshots</i> dari <i>project</i> [1]. . . . .	8
2.5 Menyimpan data sebagai perubahan terhadap versi dasar dari setiap <i>file</i> [1]. . . . .	8
2.6 <i>Working tree</i> , <i>Staging area</i> , dan Git direktori[1]. . . . .	9
2.7 <i>Checkout</i> pada <i>commit</i> . . . . .	12
3.1 Visualisasi proyek perangkat lunak menggunakan Gource. . . . .	25
3.2 Histori <i>commit</i> direpresentasikan sebagai Directed Acyclic Graph. . . . .	28
3.3 <i>Browser</i> yang dikontrol oleh ChromeDriver. . . . .	29
3.4 <i>Use case diagram</i> perangkat lunak. . . . .	31
4.1 Diagram kelas. . . . .	37
4.2 Rancangan <i>output</i> jika terdapat satu halaman <i>web</i> . . . . .	41
4.3 Rancangan <i>output</i> jika terdapat dua halaman <i>web</i> . . . . .	41
4.4 Rancangan <i>output</i> jika terdapat tiga halaman <i>web</i> . . . . .	42
4.5 Rancangan <i>output</i> jika terdapat empat halaman <i>web</i> . . . . .	42
5.1 <i>Screenshot</i> proyek Piktora pada <i>commit</i> 315d374 (31 Oktober 2016) - <i>commit</i> 89000be (12 Januari 2018). . . . .	44
5.2 Salah satu <i>commit</i> yang terdapat pada <i>file</i> hasil animasi. Terdapat judul dibagian pojok kiri bawah. . . . .	45
5.3 Salah satu <i>commit</i> yang terdapat pada <i>file</i> hasil animasi. Terdapat logo dibagian pojok kanan bawah. . . . .	45
5.4 Salah satu <i>commit</i> pada <i>file</i> hasil animasi jika terdapat satu argumen <i>-capture-url</i> . . . . .	48
5.5 Salah satu <i>commit</i> pada <i>file</i> hasil animasi jika terdapat dua argumen <i>-capture-url</i> . . . . .	48
5.6 Salah satu <i>commit</i> pada <i>file</i> hasil animasi jika terdapat tiga argumen <i>-capture-url</i> . . . . .	48
5.7 Salah satu <i>commit</i> pada <i>file</i> hasil animasi jika terdapat empat argumen <i>-capture-url</i> . . . . .	49
5.8 Tampilan pada Firefox <i>browser</i> saat dikontrol oleh FirefoxDriver. . . . .	50
5.9 Tampilan pada Opera <i>browser</i> saat dikontrol oleh OperaDriver. . . . .	51
5.10 Tampilan halaman <i>web</i> pada <i>browser</i> saat dikontrol oleh EdgeDriver. . . . .	52
5.11 Tampilan halaman <i>web</i> pada <i>browser</i> saat tidak dikontrol oleh EdgeDriver. . . . .	52
5.12 Keadaan awal dari <i>browser</i> saat dikontrol oleh InternetExplorerDriver. . . . .	53
5.13 Tampilan pada <i>browser</i> saat dikontrol oleh InternetExplorerDriver. . . . .	53



## **DAFTAR TABEL**

3.1	<i>Scenario case</i> membangkitkan animasi <i>timelapse</i> . . . . .	31
5.1	Tabel pengujian fungsional . . . . .	46
5.2	Tabel pengujian fungsional . . . . .	47



1

## BAB 1

2

### PENDAHULUAN

#### 3 1.1 Latar Belakang

4 Git merupakan perangkat lunak *Version Control Systems*[1]. *Version control* adalah sistem yang  
5 merekam perubahan pada *file* atau sekumpulan *file* dari waktu ke waktu. Perubahan yang terjadi  
6 pada *repository* dicatat oleh Git dalam bentuk histori *commit*. Setiap *commit* mengandung informasi  
7 mengenai perubahan yang terjadi pada *repository*, waktu perubahan, dan orang yang melakukan  
8 perubahan. *Database* pada *git* tidak bersifat terpusat, melainkan terdistribusi. Setiap orang yang  
9 terlibat mempunyai *database* lokal pada masing-masing komputer, sehingga pengelolaan perangkat  
10 lunak dapat dilakukan secara *online* dan *offline*.

11 JGit adalah *library* Java murni yang mengimplementasikan Git *version control systems*[2]. JGit  
12 dikembangkan oleh Eclipse Foundation. JGit bersifat *open source*. Dengan menggunakan JGit,  
13 fitur-fitur dalam Git dapat diakses melalui program Java.

14 Selenium adalah seperangkat alat yang secara khusus digunakan untuk mengotomatisasi *web*  
15 *browsers*[3]. Dengan menggunakan Selenium WebDriver, pengguna dapat memasukkan *script* bahasa  
16 pemrograman tertentu untuk melakukan pengujian. Bahasa pemrograman yang didukung yaitu C#,  
17 Java, Perl, PHP, Python, Ruby, dan JavaScript. Selenium WebDriver dapat melakukan pengujian  
18 pada Google Chrome *browser*, Firefox *browser*, Opera *browser*, Safari *browser*, Internet Explorer  
19 *browser*, dan Microsoft Edge *browser*.

20 Pada skripsi ini, akan dibuat sebuah perangkat lunak yang dapat membangun animasi *timelapse*  
21 dari pengembangan proyek perangkat lunak berbasis *web*. Yang akan dibuat animasinya adalah  
22 halaman *web* dari perangkat lunak. Jumlah halaman web bisa lebih dari satu, tergantung pada  
23 masukan. Tujuan dari dibuatnya animasi atau visualisasi adalah melihat progres dari perangkat  
24 lunak. Sehingga *programmer* bisa belajar dari proses perkembangan perangkat lunak sebelumnya  
25 dan bisa membuat perkembangan perangkat lunak menjadi lebih efisien. Perangkat lunak ini  
26 dibangun menggunakan bahasa Java. Perangkat lunak ini menggunakan tampilan terminal/konsol.  
27 Dalam pembuatan animasi timelapse, dibutuhkan perangkat lunak Selenium WebDriver dan JGit.

#### 28 1.2 Rumusan Masalah

29 Rumusan masalah dari skripsi ini adalah sebagai berikut:

30 1. Bagaimana cara membangkitkan animasi *timelapse* pada pengembangan proyek perangkat  
31 lunak berbasis web?

- 32        2. Bagaimana cara mengimplementasikan aplikasi untuk membangkitkan *timelapse* pada pe-  
33        ngembangan proyek perangkat lunak berbasis web?
- 34        3. Kesimpulan apa yang dapat diambil dari animasi *timelapse* pada beberapa situs *web* Open  
35        Source?

36 **1.3 Tujuan**

37 Tujuan dari skripsi ini adalah sebagai berikut:

- 38        1. Mengetahui cara untuk membangkitkan animasi *timelapse* pada pengembangan proyek per-  
39        angkat lunak berbasis web.
- 40        2. Mengetahui cara untuk mengimplementasikan aplikasi untuk membangkitkan *timelapse* pada  
41        pengembangan proyek perangkat lunak berbasis web.
- 42        3. Mendapatkan kesimpulan dari animasi *timelapse* pada beberapa situs *web* Open Source.

43 **1.4 Batasan Masalah**

44 Batasan masalah pada skripsi ini adalah sebagai berikut:

- 45        1. Perangkat lunak ini hanya membangkitkan animasi *timelapse* untuk perangkat lunak berbasis  
46        *web*.
- 47        2. Masukan perangkat lunak berupa alamat direktori proyek perangkat lunak yang terekam oleh  
48        Git.
- 49        3. Jumlah maksimal halaman *web* pada hasil animasi adalah empat halaman.
- 50        4. *Setup* perangkat lunak dilakukan secara otomatis melalui *script* PHP.

51 **1.5 Metodologi**

52 Metodologi penelitian yang digunakan dalam skripsi ini adalah sebagai berikut:

- 53        1. Melakukan studi literatur tentang Git, Selenium WebDriver, Git, dan JGit.
- 54        2. Melakukan analisis penggunaan Selenium WebDriver dan JGit untuk membangkitkan animasi  
55        *timelapse*.
- 56        3. Merancang perangkat lunak.
- 57        4. Membangun perangkat lunak.
- 58        5. Melakukan eksperimen dan pengujian pada perangkat lunak.

## 59    1.6 Sistematika Pembahasan

- 60    Setiap bab dalam penelitian ini memiliki sistematika penulisan yang dijelaskan ke dalam poin-poin  
61    sebagai berikut:
- 62    1. Bab 1: Pendahuluan, yaitu membahas mengenai gambaran umum penelitian ini. Berisi  
63    tentang latar belakang, rumusan masalah, tujuan, batasan masalah, metode penelitian, dan  
64    sistematika penulisan.
- 65    2. Bab 2: Dasar Teori, yaitu membahas mengenai teori-teori yang mendukung berjalannya  
66    penelitian ini. Berisi tentang teori Git, JGit, Selenium WebDriver, dan Apache Commons  
67    CLI.
- 68    3. Bab 3: Analisis, yaitu membahas mengenai analisa masalah. Berisi tentang analisis aplikasi  
69    sejenis, analisis penggunaan JGit dan Selenium WebDriver untuk membangkitkan animasi  
70    timelapse, prapengujian, dan analisis fitur aplikasi yang dibangun.
- 71    4. Bab 4: Perancangan, yaitu membahas mengenai perancangan yang dilakukan sebelum me-  
72    lakukan tahapan implementasi. Berisi tentang perancangan perangkat lunak pembangkit  
73    *timelapse* proyek pengembangan perangkat lunak.
- 74    5. Bab 5: Implementasi dan Pengujian, yaitu membahas mengenai implementasi dan pengujian  
75    aplikasi yang telah dilakukan. Berisi tentang implementasi dan hasil pengujian aplikasi.
- 76    6. Bab 6: Kesimpulan dan Saran, yaitu membahas hasil kesimpulan dari keseluruhan penelitian  
77    ini dan saran-saran yang dapat diberikan untuk penelitian berikutnya.



## BAB 2

### LANDASAN TEORI

80 Pada bab ini dibahas dasar teori yang mendukung berjalannya skripsi ini. Dasar teori yang dibahas  
 81 yaitu Git, JGit, Selenium WebDriver, dan Apache Commons CLI.

82 **2.1 Git**

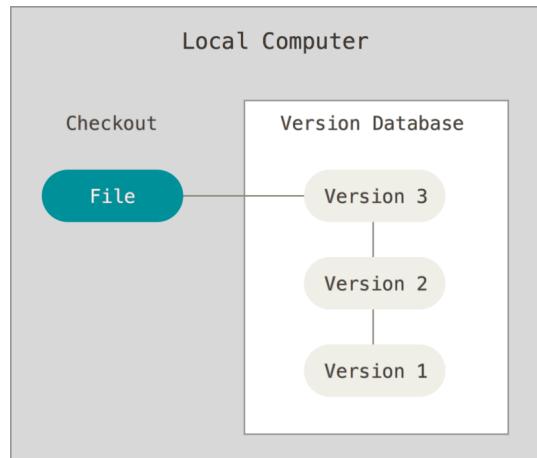
83 Git merupakan perangkat lunak *Version Control Systems*. Pada subbab ini, dijelaskan mengenai  
 84 *Version Control Systems*, cara kerja Git, Git *checkout*, dan operasi-operasi dasar pada Git. Subbab  
 85 ini mengacu pada [1].

86 **2.1.1 Version Control Systems**

87 *Version Control Systems* adalah sistem yang merekam perubahan pada *file* atau sekumpulan *file*  
 88 dari waktu ke waktu. *Version Control Systems* biasanya digunakan untuk merekam file yang berisi  
 89 *source code program*, tetapi pada kenyataannya *Version Control Systems* dapat merekam hampir  
 90 semua jenis file dalam komputer. Terdapat tiga jenis *Version Control Systems*, yaitu: *Local Version*  
 91 *Control Systems*, *Centralized Version Control Systems*, dan *Distributed Version Control Systems*.

92 **Local Version Control Systems**

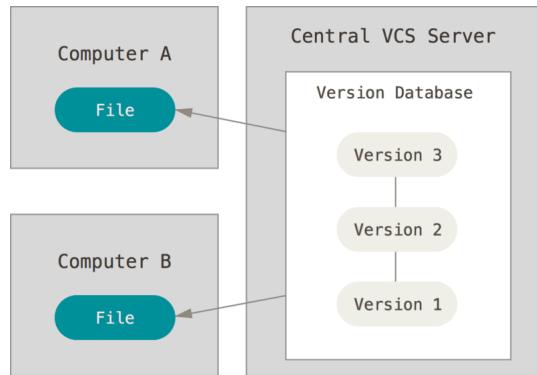
93 Metode *version-controlled* yang banyak digunakan orang adalah dengan cara menyalin sekumpulan  
 94 *file* ke direktori lain. Namun cara tersebut rentan terhadap *error*. Misalnya, terdapat direktori A  
 95 dan B, pengguna ingin mengubah *file* yang terdapat pada direktori B, tetapi pengguna lupa kalau  
 96 dia sedang berada di direktori A, maka pengguna mengubah *file* pada direktori yang salah. Untuk  
 97 mengatasi masalah tersebut, *programmer* mengembangkan *Local Version Control Systems*.



Gambar 2.1: Local version control[1].

98 Gambar 2.1 merupakan struktur dari *Local Version Control Systems*. *Database local Version*  
 99 *Control Systems* ini tersimpan pada *local directory* di komputer. *Database* ini menyimpan perubahan  
 100 file ke dalam beberapa versi atau *state*.

## 101 Centralized Version Control Systems

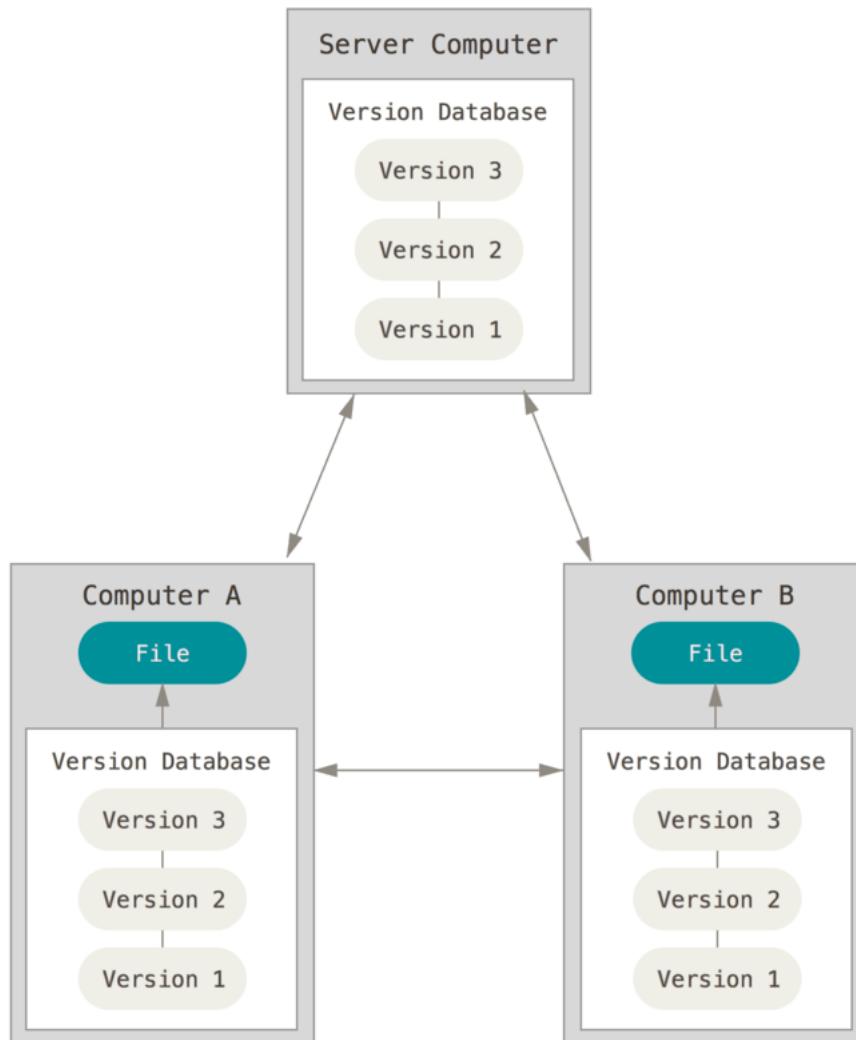


Gambar 2.2: Centralized version control[1].

102 *Local Version Control* hanya menyimpan *file* pada satu komputer saja. Muncul masalah baru ketika  
 103 *user* ingin berkolaborasi dengan *user* lain. Untuk mengatasi masalah ini dikembangkan *Centralized*  
 104 *version control*. Gambar 2.2 merupakan struktur dari *Centralized Version Control Systems*. Dalam  
 105 *Centralized Control Version Systems* terdapat sebuah *server* yang menyimpan setiap versi *file*, dan  
 106 klien yang dapat melakukan *checkout file*.

107 Sistem *Centralized Version Control Systems* memiliki beberapa kelebihan. Setiap *user* dapat  
 108 mengetahui pekerjaan yang dilakukan oleh *user* lain. Administrator dapat lebih mudah mengontrol  
 109 *database Centralized Version Control Systems* dibandingkan dengan *database Local Version Control*  
 110 *Systems* dari setiap klien.

111 Sistem *Centralized Version Control Systems* memiliki kelemahan. Jika *server* pusat *Centralized*  
 112 *Version Control Systems* mati, maka perubahan pada *file* tidak bisa disimpan. Klien juga tidak  
 113 dapat melakukan kolaborasi dengan klien lain. Jika *harddisk* pada server rusak, maka semua versi  
 114 *file* akan hilang.

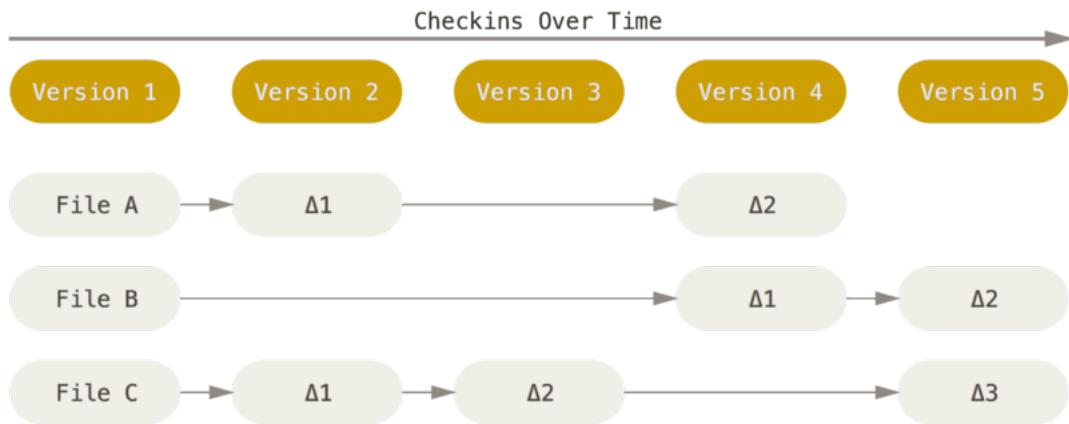
115 **Distributed Version Control Systems**

Gambar 2.3: Distributed version control[1].

116 Gambar 2.3 merupakan struktur dari *Distributed Version Control Systems*. Dalam sebuah DVCS  
 117 (seperti Git, Mercurial, Bazaar atau Darcs), klien tidak hanya melakukan *checkout* untuk *snapshot*  
 118 terakhir setiap *file*, namun klien juga memiliki salinan dari repositori tersebut. Dengan kata lain  
 119 setiap klien memiliki *version database local* pada komputernya. Jika server pusat mati, klien masih  
 120 bisa melakukan kolaborasi dan klien manapun dapat mengirimkan kembali salinan repositori ke  
 121 *server*.

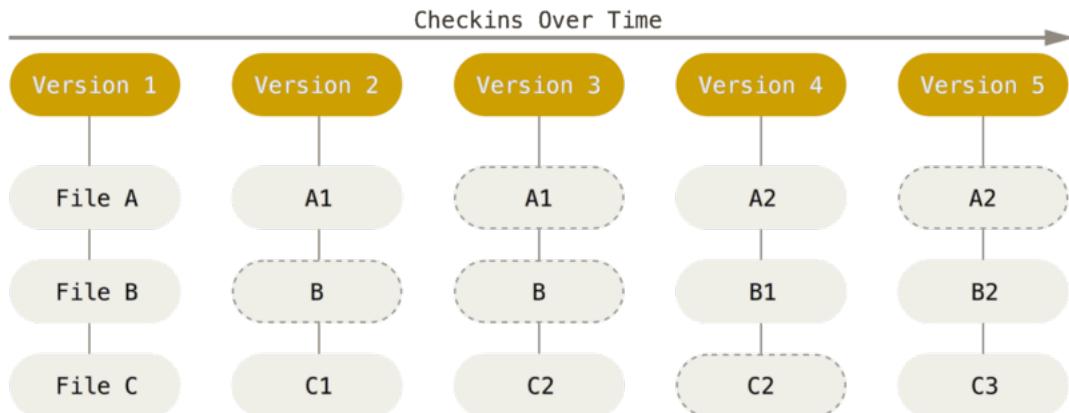
122 **2.1.2 Cara Kerja Git**

123 Salah satu perbedaan antara Git dengan VCS lainnya adalah dalam cara Git memperlakukan  
 124 datanya. Kebanyakan sistem *Version Control Systems* lain menyimpan informasi sebagai daftar  
 125 perubahan *file*. Pada Gambar 2.4, terdapat tiga *file*. *Version Control Systems* menyimpan *file* A, B,  
 126 dan C pada versi pertama saja. Untuk versi kedua dan seterusnya yang disimpan adalah perubahan  
 127 pada setiap *file*. Sistem ini disebut juga sebagai *delta-based Version Control Systems*.



Gambar 2.4: Menyimpan data sebagai *snapshots* dari *project*[1].

128 Berbeda dengan *Version Control Systems* lainnya, Git memperlakukan datanya sebagai sebuah  
 129 kumpulan *snapshot* dari sebuah miniatur *file system*. Setiap kali dilakukan *commit*, git merekam  
 130 *state* dari sekumpulan *file* dan menyimpannya sebagai *snapshot*. Gambar 2.5, menunjukkan *snapshots*  
 131 dari *file* A, B, dan C. Pada versi kedua, *file* B tidak mengalami perubahan, sehingga yang disimpan  
 132 adalah *reference* dari *file* B pada versi sebelumnya.



Gambar 2.5: Menyimpan data sebagai perubahan terhadap versi dasar dari setiap *file*[1].

### 133 State pada Git

134 Terdapat tiga *state* pada Git yaitu *committed*, *modified*, and *staged*. *Committed* adalah *state* dimana  
 135 data sudah disimpan di *local database*. *Modified* adalah *state* dimana terdapat perubahan pada *file*,  
 136 namun *file* tersebut belum di *commit* ke *database*. *Staged* adalah *state* dimana *file* telah ditandai  
 137 untuk kemudian dilakukan *commit*.

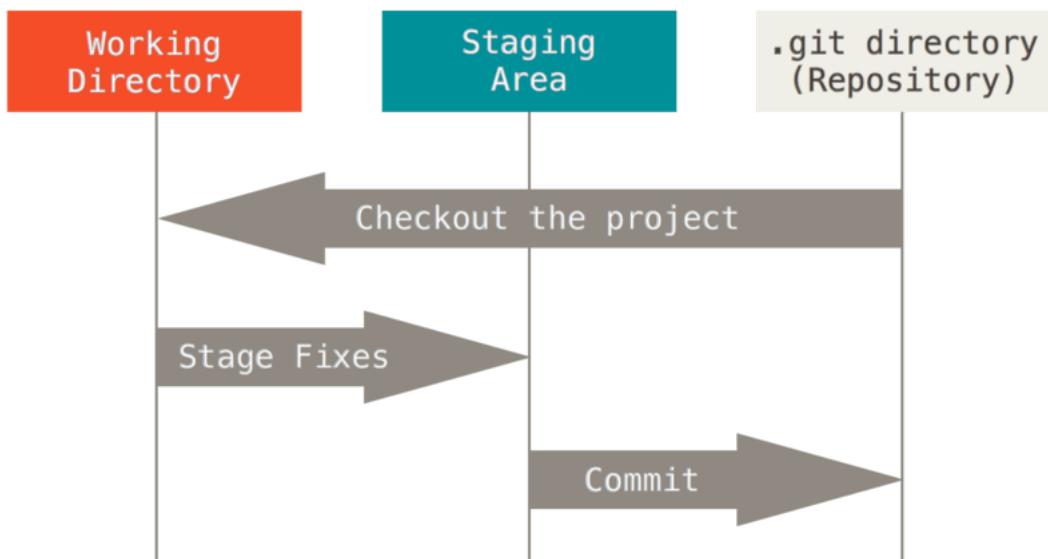
138 Terdapat tiga bagian utama dari sebuah *project* Git yaitu direktori Git, *working directory*, and  
 139 *staging area*. Direktori Git merupakan tempat dimana Git menyimpan *metadata* dan *object database*  
 140 dari *project*. *Working tree* adalah suatu *snapshot* dari *project*. Sekumpulan *file* ini diambil dari  
 141 *database* di direktori Git dan ditempatkan pada *disk* untuk digunakan dan dimodifikasi. *Staging*  
 142 area adalah suatu *file*, dimana *file* ini menyimpan daftar *file* yang telah ditandai untuk kemudian  
 143 dilakukan *commit*. *File staging area* terdapat pada direktori Git. Untuk lebih jelasnya, lihat Gambar  
 144 2.6.

<sup>145</sup> Alur kerja dari Git adalah sebagai berikut:

<sup>146</sup> 1. Melakukan modifikasi pada *file*.

<sup>147</sup> 2. Menandai perubahan pada *file* dan memindahkannya ke *staging area*.

<sup>148</sup> 3. Mengambil *file* dari *staging area* dan menyimpan *snapshot* ke direktori Git. Proses ini disebut dengan *commit*.



Gambar 2.6: *Working tree*, *Staging area*, dan Git direktori<sup>[1]</sup>.

## <sup>150</sup> Commit

<sup>151</sup> Commit merupakan sebuah *snapshot* dari suatu *file* atau direktori. Dengan kata lain yang disimpan oleh Git adalah *state* dari *file* atau direktori pada suatu waktu. *File* disimpan oleh Git dalam bentuk *blob*, sedangkan direktori disimpan dalam bentuk *tree*. *Commit*, *blob*, dan *tree* merupakan Git Object. Setiap Git Object memiliki ID berupa SHA-1 *hash value* dengan panjang empat puluh karakter heksadesimal.

Listing 2.1: Contoh histori commit dalam pengembangan perangkat lunak

```

156 commit 89000be7ce7d16f006813cddefb4ec6d70d15ed6 (HEAD -> master, origin/master,
157           origin/HEAD)
158 Author: Hizkia Steven <xvii.hs@gmail.com>
159 Date:   Fri Jan 12 12:25:30 2018 +0700
160
161     Update new company address
162
163 commit 6a085c1c37949e6308cfe06a117302e528388e54
164 Author: Hizkia Steven <xvii.hs@gmail.com>
165 Date:   Tue Dec 12 14:38:38 2017 +0700
166
167     Update company address
168
  
```

```

13 commit 9f041ef239bfe236ab4d679ad698d773a8ba6f56
14 Author: TommyAdhityaThe <toms.warior@gmail.com>
15 Date: Mon May 15 10:40:16 2017 +0700
16
17     set insta url to https://www.instagram.com/piktorastudio/
18
19 commit 38711f0cc8f487aac62babac10c1185f5ee14d33
20 Author: Tommy Adhitya The <toms.warior@gmail.com>
21 Date: Mon Apr 17 15:15:03 2017 +0700
22
23     fix bug ugly display when projects too high

```

Listing 2.1 menunjukkan sebagian histori *commit* pada *branch master*. Baris pertama menunjukkan *commit ID* berupa *SHA-1 hash value*, dengan panjang empat puluh karakter heksadesimal. Baris kedua menunjukkan orang yang melakukan *commit* dan alamat emailnya. Baris ketiga menunjukkan waktu *commit*. Baris terakhir berisi deskripsi dari *commit* tersebut.

HEAD dan *master* pada Listing 2.1, disebut juga dengan Git Ref / Reference. Ref merupakan suatu variabel yang menyimpan ID (SHA-1 hash value) dari suatu Git Object. Pada Listing 2.1, Ref *master* menyimpan ID dari *commit* yang isinya adalah 89000be7ce7d16f006813cddefb4ec6d70d15ed6. Ref HEAD tidak secara langsung menyimpan ID dari *commit*, melainkan *pointer* ke Ref *master*. Ref yang menyimpan *pointer* ke Ref lain disebut juga dengan *symbolic Ref*.

Ref origin/master merupakan Ref *master* yang berada pada *remote repository*. Ref origin/HEAD merupakan ref HEAD yang berada pada *remote repository*. *Remote repository* merupakan repositori yang disimpan pada suatu *server*, misalnya *server Github*.

### 2.1.3 Operasi Dasar pada Git

Pada subbab ini dijelaskan mengenai operasi dasar dalam Git dan sintaks-sintaksnya. Sintaks-sintaksnya ini dimasukkan pada Git *command line*. Berikut ini adalah operasi-operasi dasar dalam Git:

#### 1. Init

Operasi ini digunakan untuk membuat repositori lokal baru dengan nama tertentu. Bisa juga digunakan untuk merekam direktori yang sudah ada. Berikut adalah sintaks untuk melakukan operasi *init*:

```
$ git init [project-name]
```

#### 2. Add

Operasi ini digunakan untuk menandai perubahan pada *file* dan memindahkan *file* tersebut ke *staging area*. Operasi ini juga digunakan untuk menambahkan *file* yang dipantau perubahannya. Berikut adalah sintaks untuk melakukan operasi add:

```
$ git add [file]
```

#### 3. Commit

Operasi ini digunakan untuk merekam *snapshot* atau *state file* atau sekumpulan *file*. Operasi ini juga digunakan untuk memindahkan *file* yang berada di *staging area* ke direktori Git. Berikut adalah sintaks untuk melakukan operasi *commit*:

```
$ git commit -m "[descriptive message]"
```

211 4. Branch

212 Operasi ini digunakan untuk menampilkan semua *branch* baik itu yang terdapat di *local*  
213 *repository* maupun *remote repository*, menampilkan *branch* yang terdapat pada *remote reposi-*  
214 *tory*, membuat *branch* baru, dan menghapus *branch*. Berikut adalah sintaks-sintaks untuk  
215 melakukan operasi *branch*:

```
216 $ git branch -a  
217 $ git branch -r  
218 $ git branch [branch-name]  
219 $ git branch -d [branch-name]  
220 $ git branch -D [branch-name]
```

221 5. Diff

222 Operasi ini digunakan untuk menampilkan perbedaan pada *file* yang belum masuk *staging*  
223 *area*, menampilkan perbedaan pada *file* yang berada di *staging area* dengan *file* di *commit*  
224 sebelumnya, dan perbedaan *file* antara dua *branch*. Berikut adalah sintaks-sintaks untuk  
225 melakukan operasi *diff*:

```
226 $ git diff  
227 $ git diff -staged  
228 $ git diff [first-branch]...[second-branch]
```

229 6. Clone

230 Operasi ini digunakan untuk menyalin repositori Git yang berada di komputer lain atau suatu  
231 *server*. Berikut adalah sintaks untuk melakukan operasi *clone*:

```
232 $ git clone [url]
```

233 7. Fetch

234 Operasi ini digunakan untuk mengambil data dari *remote* repositori ke repositori lokal. Berikut  
235 adalah sintaks untuk melakukan operasi *fetch*:

```
236 $ git fetch [bookmark]
```

237 8. Merge

238 Operasi ini digunakan untuk menggabungkan *branch* tertentu dengan *branch* yang sedang  
239 aktif. Operasi ini juga digunakan untuk menggabungkan data yang diambil dari *remote*  
240 repositori dengan data pada *working directory*. Berikut adalah sintaks untuk melakukan  
241 operasi *merge*:

```
242 $ git merge [branch]/[bookmark]
```

243 9. Pull

244 Operasi ini adalah gabungan dari operasi *fetch* dan *merge*. Berikut adalah sintaks untuk  
245 melakukan operasi *pull*:

```
246 $ git pull
```

247 10. Push

248 Operasi ini digunakan untuk mengirim data pada reposipori Git lokal ke *remote repository*.  
249 Berikut adalah sintaks untuk melakukan operasi *push*:

```
250 $ git push [bookmark] [branch]
```

251 11. Checkout

252 Operasi ini digunakan untuk berpindah ke *branch* atau *commit* tertentu, setelah itu mem-  
253 perbarui *file* pada *working directory* berdasarkan *branch* atau *commit* tersebut. Berikut ini  
254 adalah sintaks-sintaks untuk operasi *checkout*:

255 \$ git checkout [commit ID]  
256 \$ git checkout [branch-name]

257 12. Log

258 Operasi ini digunakan untuk menampilkan semua histori *commit* pada *branch* yang sedang  
259 aktif. Berikut ini adalah sintaks untuk melakukan operasi *log*:

260 \$ git log

261 13. Reset

262 Operasi ini digunakan untuk memindahkan posisi *HEAD* ke *commit* tertentu, selain itu secara  
263 opsional melakukan *reset* pada *staging area* dan *working tree* berdasarkan tipe *reset*. Terdapat  
264 tiga tipe reset yaitu:

- 265 • Soft

266 Pada tipe ini, tidak dilakukan *reset* pada *staging area* dan *working directory*.

- 267 • Hard

268 Pada tipe ini, dilakukan *reset* pada *staging area* dan *working tree*. Selain itu perubahan  
269 di *working directory* setelah *commit* tertentu akan hilang.

- 270 • Mixed

271 Pada tipe ini, dilakukan *reset* pada *staging area*.

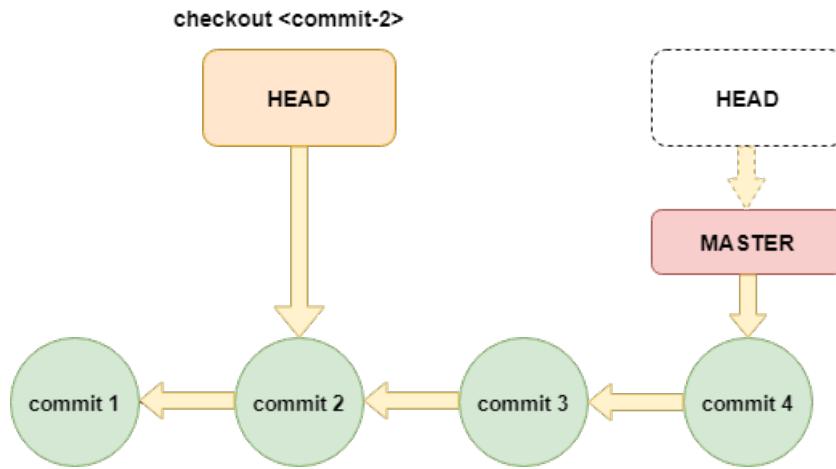
272 Berikut ini adalah sintaks untuk melakukan operasi *reset*:

273 \$ git reset -hard [commit]  
274 \$ git reset -mixed [commit]  
275 \$ git reset -soft [commit]

276 2.1.4 Git Checkout

277 Seperti yang sudah dijelaskan pada subbab 2.1.3, *checkout* dapat digunakan untuk berpindah  
278 ke *branch* atau *commit* tertentu. Operasi *checkout* dapat dilakukan menggunakan sintaks \$ git  
279 *checkout* diikuti dengan nama *branch* atau *commit ID*. Gambar 2.7 menunjukkan contoh *checkout*  
280 pada *commit*. Posisi awal *HEAD* menunjuk pada *branch master*, setelah dilakukan *checkout* ke  
281 *commit 2*, posisi *HEAD* menunjuk pada *commit 2*. *Working directory* diperbarui berdasarkan *state*  
282 pada *commit 2*.

283 *HEAD* yang menunjuk langsung ke suatu *commit* disebut dengan *detached HEAD*. Perubahan  
284 yang terjadi pada *detached HEAD* tidak akan terekam oleh Git. Jika terdapat perubahan, kemudian  
285 dilakukan *checkout commit* atau *branch*, perubahan tersebut akan hilang. *HEAD* dapat dipindahkan  
286 ke posisi semula(menunjuk pada *branch master*) dengan menggunakan sintaks \$ git *checkout*  
287 *master*.



Gambar 2.7: *Checkout* pada *commit*

## 2.2 JGit

JGit adalah *library* Java murni yang mengimplementasikan Git *version control systems*[2]. Dengan menggunakan JGit, operasi-operasi dalam Git bisa dilakukan melalui program Java. Pada subbab berikut dijelaskan beberapa kelas dari *library* JGit. Subbab ini mengacu pada [4].

### 2.2.1 Kelas Repository

Kelas ini merepresentasikan repositori Git. Berikut ini adalah beberapa *method* dalam kelas ini:

- public void create()  
Berfungsi untuk membuat repositori Git baru.
- public String getBranch()  
Berfungsi untuk mendapatkan nama *branch* yang ditunjuk oleh *HEAD*.  
Kembalian: nama dari *branch* yang sedang aktif, contohnya *master*.
- public Ref getRef(String name)  
Berfungsi untuk mendapatkan *reference* berdasarkan nama yang diberikan.  
Parameter: nama dari *reference*.  
Kembalian: *object* bertipe Ref.

### 2.2.2 Kelas Ref

Ref di dalam Git merupakan sebuah variabel yang menyimpan ID dari Git Object. Berikut adalah beberapa *method* yang dimiliki oleh kelas ini:

- String getName()  
Berfungsi untuk mengembalikan nama dari suatu Ref.  
Kembalian: nama dari Ref.
- ObjectId getObjectId()  
Berfungsi untuk mengembalikan ID dari suatu Git Object.

311 Kembalian: *object* bertipe ObjectId. Di dalam kelas ObjectId terdapat *method* getName()  
312 yang berfungsi untuk mengembalikan ID dari Git Object berupa SHA-1 *hash value*.

313 **2.2.3 Kelas FileRepository**

314 Kelas ini merupakan turunan dari kelas *Repository*. Berikut ini adalah *constructor* dari kelas ini:

- 315 • public FileRepository(String gitDir) throws IOException  
316 *Constructor* ini membuat representasi dari repositori Git. *Constructor* ini melempar IOException jika repositori tidak bisa diakses.  
318 Parameter: *path* dari suatu repositori Git.

319 **2.2.4 Kelas Git**

320 Kelas ini menyediakan API yang mirip Git Command Line untuk berinteraksi dengan repositori  
321 Git. Berikut ini adalah *constructor* dan beberapa *method* dalam kelas ini:

- 322 • public Git(Repository repo)  
323 *Constructor* ini membuat objek Git yang digunakan untuk berinteraksi dengan repositori Git.  
324 Parameter: objek *Repository* yang digunakan untuk berinteraksi. Parameter tidak boleh  
325 bernilai *null*.
- 326 • public static InitCommand init()  
327 *Method* ini mengembalikan objek *command* untuk mengeksekusi operasi *init*.  
328 Kembalian: objek *InitCommand* yang berfungsi untuk mengumpulkan parameter opsional  
329 dan akhirnya mengeksekusi operasi *init*.
- 330 • public AddCommand add()  
331 *Method* ini mengembalikan objek *command* untuk mengeksekusi operasi *add*.  
332 Kembalian: objek *AddCommand* yang berfungsi untuk mengumpulkan parameter opsional  
333 dan akhirnya mengeksekusi operasi *add*.
- 334 • public LogCommand log()  
335 *Method* ini mengembalikan objek *command* untuk mengeksekusi operasi *log*.  
336 Kembalian: objek *LogCommand* yang berfungsi untuk mengumpulkan parameter opsional  
337 dan akhirnya mengeksekusi operasi *log*.
- 338 • public ListBranchCommand branchList()  
339 *Method* ini mengembalikan objek *command* yang digunakan untuk mengeksekusi operasi  
340 *branch*. Operasi ini setara dengan sintaks *git branch -r/-a*.  
341 Kembalian: objek dengan tipe *ListBranchCommand*.
- 342 • public CheckoutCommand checkout()  
343 *Method* ini mengembalikan objek *command* untuk mengeksekusi operasi *checkout*.  
344 Kembalian: objek *ChecoutCommand* yang berfungsi untuk mengumpulkan parameter opsional  
345 dan akhirnya mengeksekusi operasi *checkout*.

- 346     ● public CommitCommand commit()  
347         *Method* ini mengembalikan objek *command* untuk mengeksekusi operasi *commit*.  
348         Kembalian: objek *CommitCommand* yang berfungsi untuk mengumpulkan parameter opsional  
349         dan akhirnya mengeksekusi operasi *commit*.
  
- 350     ● public FetchCommand fetch()  
351         *Method* ini mengembalikan objek *command* untuk mengeksekusi operasi *fetch*.  
352         Kembalian: objek *FetchCommand* yang berfungsi untuk mengumpulkan parameter opsional  
353         dan akhirnya mengeksekusi operasi *fetch*.
  
- 354     ● public PushCommand push()  
355         *Method* ini mengembalikan objek *command* untuk mengeksekusi operasi *push*.  
356         Kembalian: objek *PushCommand* yang berfungsi untuk mengumpulkan parameter opsional  
357         dan akhirnya mengeksekusi operasi *push*.
  
- 358     ● public DiffCommand diff()  
359         *Method* ini mengembalikan objek *command* untuk mengeksekusi operasi *diff*.  
360         Kembalian: objek *DiffCommand* yang berfungsi untuk mengumpulkan parameter opsional  
361         dan akhirnya mengeksekusi operasi *diff*.
  
- 362     ● public static CloneCommand cloneRepository()  
363         *Method* ini mengembalikan objek *command* untuk mengeksekusi operasi *clone*.  
364         Kembalian: objek *CloneCommand* yang berfungsi untuk mengumpulkan parameter opsional  
365         dan akhirnya mengeksekusi operasi *clone*.
  
- 366     ● public MergeCommand merge()  
367         *Method* ini mengembalikan objek *command* untuk mengeksekusi operasi *merge*.  
368         Kembalian: objek *MergeCommand* yang berfungsi untuk mengumpulkan parameter opsional  
369         dan akhirnya mengeksekusi operasi *merge*.
  
- 370     ● public PullCommand pull()  
371         *Method* ini mengembalikan objek *command* untuk mengeksekusi operasi *pull*.  
372         Kembalian: objek *PullCommand*.
  
- 373     ● public CreateBranchCommand branchCreate()  
374         *Method* ini mengembalikan objek *command* untuk membuat *branch* baru.  
375         Kembalian: objek *CreateBranchCommand*.
  
- 376     ● public ListBranchCommand branchList()  
377         *Method* ini mengembalikan objek *command* untuk menampilkan daftar *branch*.  
378         Kembalian: objek *ListBranchCommand*.
  
- 379     ● public DeleteBranchCommand branchDelete()  
380         *Method* ini mengembalikan objek *command* untuk menghapus *branch*.  
381         Kembalian: objek *DeleteBranchCommand*.

- 382 • public ResetCommand reset()  
383     *Method ini mengembalikan objek *command* untuk mengeksekusi operasi *reset*.*  
384     *Kembalian: objek *ResetCommand*.*

385 **2.2.5 Kelas CheckoutCommand**

386 Kelas yang digunakan untuk melakukan operasi *checkout*. Berikut adalah beberapa *method* yang  
387 terdapat pada kelas ini:

- 388 • public CheckoutCommand setName(String name)  
389     Menentukan nama *branch* atau *commit* untuk melakukan checkout.  
390     Parameter: nama dari *branch* atau *commit*.  
391     Kembalian: *object CheckoutCommand*.  
392 • public Ref call()  
393     Berfungsi untuk menjalankan operasi *checkout*.

394 **2.2.6 Kelas ListBranchCommand**

395 Kelas yang digunakan untuk melakukan operasi *branch* dan mendapatkan daftar *branch*. Berikut  
396 adalah beberapa *method* yang terdapat pada kelas ini:

- 397 • public ListBranchCommand setListMode(ListBranchCommand.ListMode listMode)  
398     Menentukan mode yang digunakan untuk mendapatkan *branch*. Jika parameter tidak dima-  
399 sukan, hanya daftar *branch* lokal yang didapatkan.  
400     Parameter: mode yang digunakan untuk mendapatkan *branch* berupa konstanta.  
401     Kembalian: *object CheckoutCommand*.  
402 • public List<Ref> call()  
403     Berfungsi untuk menjalankan operasi *branch*.  
404     Kembalian: daftar *branch* dalam bentuk List<Ref>.

405 **2.2.7 Enum ListBranchCommand.ListMode**

406 Merupakan sebuah Enum yang berisi mode yang digunakan untuk mendapatkan *branch*. Berikut  
407 ini adalah konstanta yang terdapat pada Enum ini:

- 408 • public static final ListBranchCommand.ListMode ALL  
409     Mode ini digunakan untuk mendapatkan semua daftar *branch* baik lokal maupun *remote*.  
410     Mode ini setara dengan *Option -a* pada sintaks git branch -a.  
411 • public static final ListBranchCommand.ListMode REMOTE  
412     Mode ini digunakan untuk mendapatkan daftar *remote branch*. Mode ini setara dengan *Option -r*  
413     pada sintaks git branch -r.

**414 2.2.8 Kelas LogCommand**

415 Kelas yang digunakan untuk melakukan operasi *log*. Berikut adalah *method* yang terdapat pada  
416 kelas ini:

- 417 • public Iterable<RevCommit> call()

Berfungsi untuk menjalankan operasi *log*.

419 Kembalian: histori *commit* pada *branch* yang sedang aktif.

**420 2.2.9 Kelas ResetCommand**

421 Kelas yang digunakan untuk melakukan operasi *reset*. Berikut adalah beberapa *method* yang  
422 terdapat pada kelas ini:

- 423 • public ResetCommand setMode(ResetCommand.ResetType mode)

424 Menentukan tipe *reset*.

425 Parameter: tipe *reset*.

426 Kembalian: *object* ResetCommand.

- 427 • public ResetCommand setRef(String ref)

428 Mengatur nama *reference* tujuan dalam operasi *reset*. Secara default, *reference* yang digunakan  
429 adalah *HEAD*.

430 Parameter: nama dari *reference*.

431 Kembalian: *object* ResetCommand.

- 432 • public Ref call()

433 Berfungsi untuk menjalankan operasi *reset*.

434

**435 2.2.10 Kelas ResetCommand.ResetType**

436 Merupakan *enumeration* yang menentukan tipe *reset* yang digunakan pada operasi *reset*. *Enumeration*  
437 tersebut adalah sebagai berikut:

- 438 • public static final ResetCommand.ResetType SOFT

439 Hanya mengubah posisi *HEAD*.

- 440 • public static final ResetCommand.ResetType HARD

441 Mengubah posisi *HEAD* ke *reference* tujuan. Selain itu, mengubah *staging area* dan *working*  
442 *directory* berdasarkan *reference* tersebut.

- 443 • public static final ResetCommand.ResetType MIXED

444 Mengubah posisi *HEAD* ke *reference* tujuan dan mengubah *staging area* berdasarkan *reference*  
445 tersebut.

**446 2.2.11 Kelas RevCommit**

447 Kelas ini merupakan *reference* ke *commit* yang ada di *Directed Acyclic Graph*. Berikut ini adalah  
448 beberapa *method* dari kelas ini:

- 449 • public final String getFullMessage()  
450 Berfungsi untuk mendapatkan deskripsi dari suatu *commit*.  
451 Kembalian: deskripsi dari suatu *commit* dalam bentuk String.
- 452 • public final String getShortMessage()  
453 Berfungsi untuk mendapatkan deskripsi dari suatu *commit*. Jika panjang deskripsi *commit*  
454 lebih dari satu baris, maka hanya baris pertama yang dikembalikan.  
455 Kembalian: baris pertama dari deskripsi *commit* dalam bentuk String.
- 456 • public final String getName()  
457 *Method* ini mengembalikan *commit ID*.  
458 Kembalian: *commit ID* berupa SHA-1 hash value dalam bentuk String.
- 459 • public final PersonIdent getAuthorIdent()  
460 Berfungsi untuk mendapatkan informasi mengenai *author* yang melakukan *commit*.  
461 Kembalian: objek *PersonIdent* yang memuat informasi tentang *author*(nama dan *email*) dan  
462 waktu dilakukannya *commit*.

#### 463 2.2.12 Kelas PersonIdent

464 Kelas ini memberikan informasi mengenai *author* dari suatu *commit*. Berikut ini adalah beberapa  
465 *method* dari kelas ini:

- 466 • public String getName()  
467 Berfungsi untuk mengembalikan nama dari *author* yang melakukan *commit*.  
468 Kembalian: nama dari *author*.
- 469 • public String getEmailAddress()  
470 Berfungsi untuk mengembalikan alamat *email* dari *author* yang melakukan *commit*.  
471 Kembalian: alamat *email* dari *author*.
- 472 • public Date getWhen()  
473 Berfungsi mengembalikan waktu dilakukannya suatu *commit* oleh *author*.  
474 Kembalian: waktu dilakukannya suatu *commit* berupa objek Date.

### 475 2.3 Selenium WebDriver

476 Selenium adalah seperangkat alat yang secara khusus digunakan untuk mengotomatisasi *web*  
477 *browsers*[3]. Selenium mendukung bahasa pemrograman C#, Java, Perl, PHP, Python, Ruby,  
478 dan JavaScript. Selenium terdiri dari beberapa bagian, yaitu Selenium 1(Selenium RC), Selenium  
479 2(Selenium WebDriver), dan Selenium IDE. Selenium RC merupakan proyek utama *Selenium* untuk  
480 waktu yang lama, sebelum akhirnya bergabung dengan WebDriver menjadi Selenium 2. Selenium  
481 RC melakukan *automation test* dengan cara menginjeksi kode JavaScript ke *browser*. Selenium RC  
482 sekarang sudah *deprecated* dan tidak digunakan lagi. Selenium Webdriver merupakan gabungan  
483 dari Selenium RC dan WebDriver. Selenium IDE merupakan *extension* dari *browser* Chrome dan  
484 Firefox yang digunakan untuk melakukan *record and playback test* pada *browser*.

485 WebDriver merupakan kakas untuk mengotomatisasi pengujian pada aplikasi *web*[5]. WebDriver  
486 dapat berkomunikasi secara langsung dengan *browser* menggunakan *native support* pada *browser*  
487 tanpa melakukan injeksi kode JavaScript. WebDriver yang terdapat pada Selenium WebDriver antara  
488 lain ChromeDriver, FirefoxDriver, OperaDriver, SafariDriver, InternetExplorerDriver, EdgeDriver,  
489 dan HtmlUnitDriver.

490 ChromeDriver merupakan implementasi WebDriver yang mengontrol Chrome *browser*. Fire-  
491 foxDriver merupakan implementasi WebDriver yang mengontrol Firefox *browser*. OperaDriver  
492 merupakan implementasi WebDriver yang mengontrol Opera *browser*. SafariDriver merupakan  
493 implementasi WebDriver yang mengontrol Safari *browser*. InternetExplorerDriver merupakan  
494 implementasi WebDriver yang mengontrol Internet Explorer *browser*. EdgeDriver merupakan  
495 implementasi WebDriver yang mengontrol Microsoft Edge *browser*. HtmlUnitDriver merupakan im-  
496 plementasi WebDriver yang mengontrol HTMLUnit *browser* yang bersifat *headless*(tidak mempunyai  
497 GUI).

498 Pada skripsi ini *tools* Selenium yang digunakan hanya Selenium WebDriver. Bahasa pemro-  
499 graman yang digunakan adalah java. WebDriver yang digunakan adalah ChromeDriver. Pemilihan  
500 WebDriver ini hanya berdasarkan preferensi penulis saja. Penulis tidak memilih InternetExplore-  
501 rDriver, EdgeDriver, dan SafariDriver karena InternetExplorerDriver dan EdgeDriver hanya bisa  
502 berjalan pada sistem operasi Windows, selain itu SafariDriver hanya bisa berjalan pada sistem  
503 operasi macOS. HtmlUnitDriver tidak dipilih karena tidak mempunyai GUI sehingga tidak bisa  
504 dilakukan pengambilan *screenshot*. Pada subbab berikut dijelaskan beberapa kelas dan *interface*  
505 dari *library* Selenium WebDriver. Subbab ini mengacu pada [6].

### 506 2.3.1 Interface WebDriver

507 Merupakan *interface* utama yang digunakan untuk pengujian. Berikut ini adalah beberapa *method*  
508 dalam *interface* ini:

- 509 • void close()

510 Berfungsi untuk menutup *window* pada *browser*, jika *window* yang sekarang merupakan  
511 satu-satunya *window* yang terbuka maka *browser* akan ditutup.

- 512 • void quit()

513 Berfungsi untuk menutup *browser* dan semua *window* yang sedang terbuka.

- 514 • void get(String url)

515 Berfungsi untuk memuat halaman *web* pada *window* yang sedang aktif. *Method* ini mengirim  
516 *HTPP GET Request* untuk memuat halaman, dan *method* ini akan melakukan *blocking* sampai  
517 halaman *web* selesai dimuat.

518 Parameter: alamat *url* untuk memuat halaman *web*.

- 519 • String getTitle()

520 Berfungsi untuk mengembalikan judul dari halaman *web* yang sedang aktif.

521 Kembalian: judul dari halaman *web*.

- 522 • String getCurrentUrl()

523 Berfungsi untuk mendapatkan URL yang sedang aktif di *browser*.

524 Kembalian: URL dari halaman *web* yang sedang dimuat di *browser*.  
525 • WebDriver.Options manage()  
526 Berfungsi untuk mendapatkan interface WebDriver.Options.  
527 Kembalian: interface dengan tipe WebDriver.Options. Di dalam kelas ini terdapat *method*  
528 window() yang mengembalikan *interface* dengan tipe WebDriver.Window.

### 529 2.3.2 Interface WebDriver.Window

530 *Interface* yang digunakan untuk mengatur *window* pada *browser*. Berikut adalah *method* yang  
531 dimiliki oleh *interface* ini:

532 • void maximize()  
533 Berfungsi untuk membuat ukuran *window* pada *browser* menjadi maksimal.  
534 • public setSize(Dimension targetSize)  
535 Berfungsi untuk mengatur ukuran *window* pada *browser* sesuai parameter.  
536 Parameter: ukuran dari *browser* berupa objek Dimension. Objek Dimension mempunyai  
537 atribut height dan weight yang menyatakan ukuran panjang dan lebar dalam satuan piksel.

### 538 2.3.3 Kelas ChromeDriver

539 Merupakan merupakan implementasi WebDriver yang mengontrol Chrome *browser*. Berikut adalah  
540 *constructor* yang terdapat pada kelas ini:

541 • public ChromeDriver()  
542 *Constructor* ini membuat instans dengan tipe ChromeDriver.

### 543 2.3.4 Kelas FirefoxDriver

544 Merupakan merupakan implementasi WebDriver yang mengontrol Firefox *browser*. Berikut adalah  
545 *constructor* yang terdapat pada kelas ini:

546 • public FirefoxDriver()  
547 *Constructor* ini membuat instans dengan tipe FirefoxDriver.

### 548 2.3.5 Kelas OperaDriver

549 Merupakan merupakan implementasi WebDriver yang mengontrol Opera *browser*. Berikut adalah  
550 *constructor* yang terdapat pada kelas ini:

551 • public OperaDriver()  
552 *Constructor* ini membuat instans dengan tipe OperaDriver.

### 553 2.3.6 Kelas SafariDriver

554 Merupakan merupakan implementasi WebDriver yang mengontrol Safari *browser*. Berikut adalah  
555 *constructor* yang terdapat pada kelas ini:

556 • public SafariDriver()  
557 *Constructor* ini membuat instans dengan tipe SafariDriver.

558 **2.3.7 Kelas InternetExplorerDriver**

559 Merupakan merupakan implementasi WebDriver yang mengontrol Internet Explorer *browser*. Berikut  
560 adalah *constructor* yang terdapat pada kelas ini:

- 561 • public InternetExplorerDriver()

562 *Constructor* ini membuat instans dengan tipe InternetExplorerDriver.

563 **2.3.8 Kelas EdgeDriver**

564 Merupakan merupakan implementasi WebDriver yang mengontrol Microsoft Edge *browser*. Berikut  
565 adalah *constructor* yang terdapat pada kelas ini:

- 566 • public EdgeDriver()

567 *Constructor* ini membuat instans dengan tipe EdgeDriver.

568 **2.3.9 Kelas HtmlUnitDriver**

569 Merupakan merupakan implementasi WebDriver yang mengontrol HtmlUnit *browser*. Berikut  
570 adalah *constructor* yang terdapat pada kelas ini:

- 571 • public HtmlUnitDriver()

572 *Constructor* ini membuat instans dengan tipe HtmlUnitDriver.

573 **2.3.10 Interface OutputType**

574 Merupakan *interface* yang menentukan tipe *output* pada *screenshot*. Terdapat tiga konstanta untuk  
575 menentukan tipe *output* pada *screenshot*. Konstanta tersebut adalah sebagai berikut:

- 576 • static final OutputType<String> BASE64

577 Berfungsi untuk mendapatkan *screenshot* dalam bentuk *base64 data*.

- 578 • static final OutputType<byte[]> BYTES

579 Berfungsi untuk mendapatkan *screenshot* dalam bentuk *raw bytes*.

- 580 • static final OutputType<java.io.File> FILE

581 Berfungsi untuk mendapatkan *screenshot* dalam bentuk *temporary file* yang akan dihapus  
582 setelah program keluar dari *Java Virtual Machine*.

583 **2.3.11 Interface TakesScreenshot**

584 Merupakan *interface* yang digunakan untuk mengambil *screenshot*. WebDriver yang mengimplementasikan  
585 *interface* ini yaitu ChromeDriver, FirefoxDriver, OperaDriver, InternetExplorerDriver,  
586 dan EdgeDriver. HtmlUnitDriver tidak mengimplementasikan *interface* ini. Berikut adalah *method*  
587 yang terdapat dalam *interface* ini:

- 588 • <X> X getScreenshotAs(OutputType<X> target) throws WebDriverException

589 *Method* ini berfungsi untuk mengambil *screenshot* dan mengembalikannya.

590 Kembalian: objek hasil *screenshot*

591 Parameter: tipe *output* yang diinginkan(lihat subbab [2.3.10](#)).

## 592 2.4 Apache Commons CLI

593 Library Apache Commons CLI menyediakan API untuk melakukan *parsing* argumen Command  
594 Line yang dikirimkan ke program[7]. Apache Commons CLI termasuk ke dalam salah satu *project*  
595 Apache Commons. Tujuan utama dari *project* Apache Commons adalah membuat dan melakukan  
596 *maintain* pada komponen Java yang *reusable*. Pada subbab berikut dijelaskan beberapa kelas dan  
597 *interface* dari *library* Apache Commons CLI. Subbab ini mengacu pada [8].

### 598 2.4.1 CommandLineParser

599 Merupakan sebuah *interface*. Kelas yang mengimplementasikan *interface* ini dapat melakukan  
600 *parsing* argumen CommandLine berdasarkan pada *option* yang telah ditentukan. Berikut ini adalah  
601 beberapa *method* yang dimiliki *interface* ini:

- 602 • CommandLine parse(Options options, String[] arguments) throws ParseException  
603 Berfungsi untuk melakukan *parsing* argumen Command Line berdasarkan pada *option* yang  
604 telah ditentukan. *Method* ini melempar *ParseException* jika terjadi masalah saat melakukan  
605 *parsing*.  
606 Parameter: *option* yang telah ditentukan, argumen Command Line.  
607 Kembalian: objek *CommandLine*.
- 608 • CommandLine parse(Options options, String[] arguments, boolean stopAtNonOption) throws  
609 ParseException  
610 Berfungsi untuk melakukan *parsing* argumen Command Line berdasarkan pada *option* yang  
611 telah ditentukan.  
612 Parameter: *option* yang telah ditentukan, argumen Command Line, dan suatu *boolean* yang  
613 menentukan apakah *parsing* dihentikan jika terdapat argumen yang tidak valid. Jika bernilai  
614 *true*, *parsing* akan dihentikan dan semua argumen yang sudah diuraikan akan ditambahkan ke  
615 objek *CommandLine*. Jika bernilai *false*, akan dilempar *ParseException* bila terdapat Option  
616 yang tidak valid.  
617 Kembalian: objek *CommandLine*.

### 618 2.4.2 DefaultParser

619 Merupakan kelas yang mengimplementasikan *interface* CommandLineParser. Berikut adalah *method*  
620 yang dimiliki kelas ini:

- 621 • CommandLine parse(Options options, String[] arguments) throws ParseException  
622 Berfungsi untuk melakukan *parsing* argumen Command Line berdasarkan pada *option* yang  
623 telah ditentukan. *Method* ini melempar *ParseException* jika terjadi masalah saat melakukan  
624 *parsing*.  
625 Parameter: *option* yang telah ditentukan, argumen Command Line.  
626 Kembalian: objek *CommandLine*.
- 627 • CommandLine parse(Options options, String[] arguments, boolean stopAtNonOption) throws  
628 ParseException

629 Berfungsi untuk melakukan *parsing* argumen Command Line berdasarkan pada *option* yang  
630 telah ditentukan.

631 Parameter: *option* yang telah ditentukan, argumen Command Line, dan suatu *boolean* yang  
632 menentukan apakah *parsing* dihentikan jika terdapat argumen yang tidak valid. Jika bernilai  
633 *true*, *parsing* akan dihentikan dan semua argumen yang sudah diuraikan akan ditambahkan ke  
634 objek *CommandLine*. Jika bernilai *false*, akan dilempar *ParseException* bila terdapat Option  
635 yang tidak valid.

636 Kembalian: objek *CommandLine*.

#### 637 2.4.3 CommandLine

638 Kelas ini merepresentasikan kumpulan Option hasil *parsing*. Berikut ini adalah beberapa *method*  
639 yang dimiliki kelas ini:

- public String getOptionValue(String opt)

Berfungsi untuk mendapatkan nilai dari suatu Option sesuai dengan namanya.

Parameter: nama dari Option.

Kembalian: nilai dari suatu Option. Jika Option tidak ditemukan, akan mengembalikan *null*.

- public String[] getOptionValues(Option option)

Berfungsi untuk mendapatkan kumpulan nilai dari suatu Option. Parameter: nama dari  
645 Option. Kembalian: kumpulan nilai dari suatu Option dengan tipe *array of String*. Jika  
646 Option tidak ditemukan, akan mengembalikan *null*.

- public Option[] getOptions()

Berfungsi untuk mengembalikan Option-option hasil *parsing*.

Kembalian: *array* dari Option hasil *parsing*.

#### 651 2.4.4 Options

652 Kelas ini merepresentasikan kumpulan dari objek *Option*, yang mendeskripsikan kemungkinan  
653 *option* pada *command line*. Berikut ini adalah *method* yang dimiliki kelas ini:

- public Options addOption(Option opt)

Berfungsi untuk menambahkan Option.

Parameter: Option yang akan ditambahkan.

Kembalian: hasil dari Option yang ditambahkan.

#### 658 2.4.5 Option

659 Kelas ini merepresentasikan sebuah Command Line Option. Berikut ini adalah beberapa *method*  
660 yang dimiliki kelas ini:

- public String getLongOpt()

Berfungsi untuk mendapatkan nama panjang dari suatu *option*.

Kembalian: nama panjang dari suatu *option*.

- 664 • public String getValue()  
Berfungsi untuk mendapatkan nilai dari Option.  
Kembalian: nilai dari Option. Jika terdapat lebih dari satu kembalian, hanya nilai pertama  
yang dikembalikan. Jika tidak ada nilai, akan mengembalikan *null*.
- 668 • public String[] getValues()  
Berfungsi untuk mendapatkan nilai-nilai dari Option.  
Kembalian: nilai-nilai dari Option berupa *array of String*. Jika tidak ada nilai, akan mengem-  
balikan *null*.
- 672 • public boolean hasArg()  
Berfungsi untuk mengetahui apakah suatu *option* membutuhkan argumen.  
Kembalian: *true* jika *option* ini membutuhkan argumen , *false* jika *option* ini tidak membu-  
tuhkan argumen.
- 676 • public boolean hasArg()  
Berfungsi untuk mengetahui apakah suatu *option* dapat menerima lebih dari satu argumen.  
Kembalian: *true* jika *option* ini dapat menerima lebih dari satu argumen , *false* jika tidak.
- 679 • public String getDescription()  
Berfungsi untuk mendapatkan deskripsi dari suatu *option*.  
Kembalian: deskripsi dari *option* ini.
- 682 • public String getArgName()  
Berfungsi untuk mendapatkan nama dari suatu *option*.  
Kembalian: nama dari argumen suatu *option*

#### 685 2.4.6 Option.Builder

686 Kelas ini merupakan *nested class* dari kelas *Option*. Kelas ini digunakan untuk membuat objek  
687 Option berdasarkan parameter yang diberikan. Berikut ini adalah beberapa *method* yang dimiliki  
688 kelas ini:

- 689 • public Option.Builder desc(String description)  
Berfungsi untuk memberikan deskripsi pada *option*.  
Parameter: deskripsi dari *option*.  
Kembalian: objek *Option.Builder* yang bisa digunakan untuk *method chaining*.
- 693 • public Option.Builder longOpt(String longOpt)  
Berfungsi untuk memberikan nama panjang pada *option*.  
Parameter: nama panjang *option*.  
Kembalian: objek *Option.Builder* yang bisa digunakan untuk *method chaining*.
- 697 • public Option.Builder hasArg()  
Berfungsi untuk menyatakan bahwa *option* ini membutuhkan argumen.  
Kembalian: objek *Option.Builder* yang bisa digunakan untuk *method chaining*.

- 700     • public Option.Builder hasArgs()

701       Bergfungsi untuk menyatakan bahwa *option* ini membutuhkan argumen, dimana jumlah  
702       argumen bisa lebih dari satu.

703       Kembalian: objek *Option.Builder* yang bisa digunakan untuk *method chaining*.

- 704     • public Option.Builder argName(String argName)

705       Bergfungsi untuk memberi nama pada argumen.

706       Parameter: nama argumen.

707       Kembalian: objek *Option.Builder* yang bisa digunakan untuk *method chaining*.

- 708     • public Option.Builder required(boolean required)

709       Bergfungsi untuk menyatakan bahwa *option* ini wajib ada.

710       Parameter: variabel bertipe *boolean* yang menentukan apakah *option* ini wajib ada.

711       Kembalian: objek *Option.Builder* yang bisa digunakan untuk *method chaining*.

- 712     • public Option build()

713       Berfungi untuk membuat objek *Option* berdasarkan nilai pada *Option.Builder*.

714       Kembalian: objek *Option*.



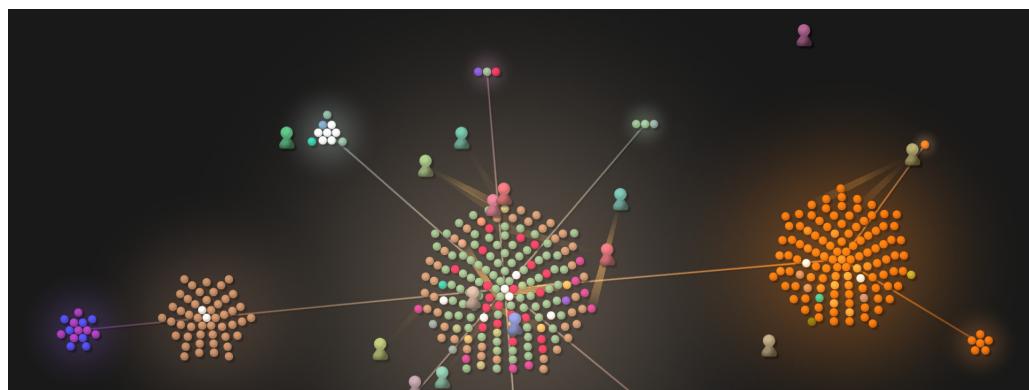
## BAB 3

## ANALISIS

717 Pada bab ini dijelaskan mengenai analisis aplikasi sejenis, analisis penggunaan JGit dan Selenium  
 718 WebDriver untuk membangkitkan animasi timelapse, prapengujian, dan analisis fitur aplikasi yang  
 719 dibangun.

720 **3.1 Analisis Aplikasi Sejenis**

721 Saat skripsi ini dibuat, aplikasi sejenis yang digunakan untuk membangkitkan animasi adalah Gource.  
 722 Proyek perangkat lunak ditampilkan oleh Gource sebagai animasi pohon, dimana pusatnya adalah  
 723 *root directory* dari proyek perangkat lunak[9]. Direktori ditampilkan sebagai *branch*, sedangkan *file*  
 724 ditampilkan sebagai *leaf*. Developer dapat terlihat di *working tree* pada saat mereka berkontribusi  
 725 untuk proyek.



Gambar 3.1: Visualisasi proyek perangkat lunak menggunakan Gource.

726 Gambar 3.1 menunjukkan contoh visualisasi proyek perangkat lunak menggunakan Gource. Efek  
 727 cahaya yang terdapat pada Gambar 3.1 disebut dengan *bloom*. Pada awalnya ukuran *working tree*  
 728 tidak terlalu besar. Setiap kali ditambahkan *file* dan *folder* baru, akan dibuat *branch* dan *leaf* baru  
 729 pada *working tree*.

731 Gource memiliki beberapa fitur. Fitur-fitur tersebut dapat diatur melalui *command line option*.  
 732 Berikut ini adalah beberapa *command line option* yang terdapat pada Gource:

733 1. `gource -[WIDTH]x[HEIGHT]`

734 Opsi ini berfungsi untuk mengatur resolusi layar dari animasi. Parameter dari opsi ini adalah  
 735 lebar dan panjang layar dalam satuan piksel.

736 2. **gource -camera-mode [MODE]**

737 Opsi ini berfungsi untuk mengatur mode kamera pada Gource. Parameter dari opsi ini adalah  
738 mode dari kamera. Terdapat dua mode yaitu *overview* dan *track*. Dalam mode *track*, kamera  
739 bergerak mengikuti *user* yang sedang aktif. Dalam mode *overview*, kamera menampilkan  
740 seluruh repositori.

741 3. **gource -path [PATH]**

742 Opsi ini berfungsi untuk mengatur *path* dari direktori yang akan dibuat animasinya. Opsi  
743 dari parameter ini adalah *path* dari direktori.

744 4. **gource -start-date [YYYY-MM-DD hh:mm:ss +tz] -stop-date [YYYY-MM-DD hh:mm:ss +tz]**

745 Opsi ini berfungsi untuk mengatur periode waktu dalam menampilkan animasi. Parameter  
746 dari opsi ini adalah waktu mulai dan waktu akhir dalam format "YYYY-MM-DD hh:mm:ss  
747 +tz". Dimana YYYY adalah tahun, MM adalah bulan, DD adalah tanggal, hh adalah jam,  
748 mm adalah menit, ss adalah detik, dan +tz adalah zona waktu. Parameter jam, menit, detik,  
749 dan zona waktu bersifat opsional.

750 5. **gource -bloom-multiplier [FLOAT]**

751 Opsi ini berfungsi untuk mengatur radius dari efek *bloom*. Parameter dari opsi ini adalah  
752 radius dalam format bilangan riil.

753 6. **gource -bloom-intensity [FLOAT]**

754 Opsi ini berfungsi untuk mengatur intensitas dari efek *bloom*. Parameter dari opsi ini adalah  
755 intensitas *bloom* dalam format bilangan riil.

756 7. **gource -disable-bloom**

757 Opsi ini berfungsi untuk menonaktifkan animasi *bloom*.

758 8. **gource -date-format [FORMAT]**

759 Opsi untuk mengatur format waktu yang ditampilkan pada bagian tengah atas. Opsi dari  
760 parameter ini adalah format waktu dalam bentuk *string*.

761 9. **gource -background [FFFFFF]**

762 Opsi ini berfungsi untuk mengatur warna *background*. Parameter dari opsi ini adalah warna  
763 dalam format heksadesimal.

764 10. **gource -background-image [IMAGE]**

765 Opsi ini berfungsi untuk mengatur gambar *background*. Parameter dari opsi ini adalah nama  
766 file dari gambar.

767 11. **gource -font-size [SIZE]**

768 Opsi ini digunakan untuk mengatur ukuran *font* pada tulisan *title* dan tanggal. Parameter  
769 dari opsi ini adalah ukuran *font*.

770 12. **gource -font-colour [FFFFFF]**

771 Opsi ini digunakan untuk mengatur warna *font* pada tulisan *title* dan tanggal. Parameter  
772 dari opsi ini adalah warna *font* dalam format heksadesimal.

773 13. `gource -logo [IMAGE]`

774 Opsi ini berfungsi untuk memasukkan logo. Parameter dari opsi ini adalah nama *file* dari  
775 gambar.

776 14. `gource -logo-offset [X]x[Y]`

777 Opsi ini berfungsi untuk mengatur posisi dari logo. Parameter dari opsi ini adalah posisi x  
778 dan posisi y dari logo.

779 15. `gource -title [TITLE]`

780 Opsi ini berfungsi untuk memberi judul. Dimana judul tersebut ditampilkan pada pojok kiri  
781 bawah layar.

782 16. `gource -output-framerate [FPS]`

783 Opsi ini berfungsi untuk mengatur jumlah *frame* per detik pada video animasi. Parameter  
784 dari opsi ini adalah jumlah *frame* per detik.

785 17. `gource -seconds-per-day [SECONDS]`

786 Opsi ini berfungsi untuk mengatur kecepatan visualisasi. Parameter dari opsi ini adalah waktu  
787 yang dibutuhkan untuk menampilkan satu hari.

788 18. `gource -hide [DISPLAY-ELEMENT]`

789 Opsi ini berfungsi untuk menyembunyikan satu atau lebih *display element*. Parameter dari  
790 opsi ini adalah elemen yang akan disembunyikan. *Display element* yang dapat disembunyikan  
791 yaitu:

- 792 • *bloom*: efek *bloom*.
- 793 • *date*: waktu.
- 794 • *dirnames*: nama direktori.
- 795 • *files*: ikon dari berkas.
- 796 • *filenames*: nama berkas.
- 797 • *root*: *root directory*.
- 798 • *users*: ikon dari *user*.
- 799 • *usernames*: nama dari *user*.

800 Parameter yang berjumlah lebih dari satu dipisahkan dengan koma, contoh: *bloom,root,users*.

801 Gource dapat digunakan untuk berbagai macam proyek perangkat lunak. Program pada skripsi

802 ini hanya akan berfokus untuk proyek perangkat lunak berbasis *web*. Tidak seperti Gource yang  
803 menampilkan direktori dan *file* pada animasi, program pada skripsi ini menampilkan *screenshot*  
804 dari halaman suatu *website*.

## 805 3.2 Analisis Penggunaan JGit dan Selenium WebDriver

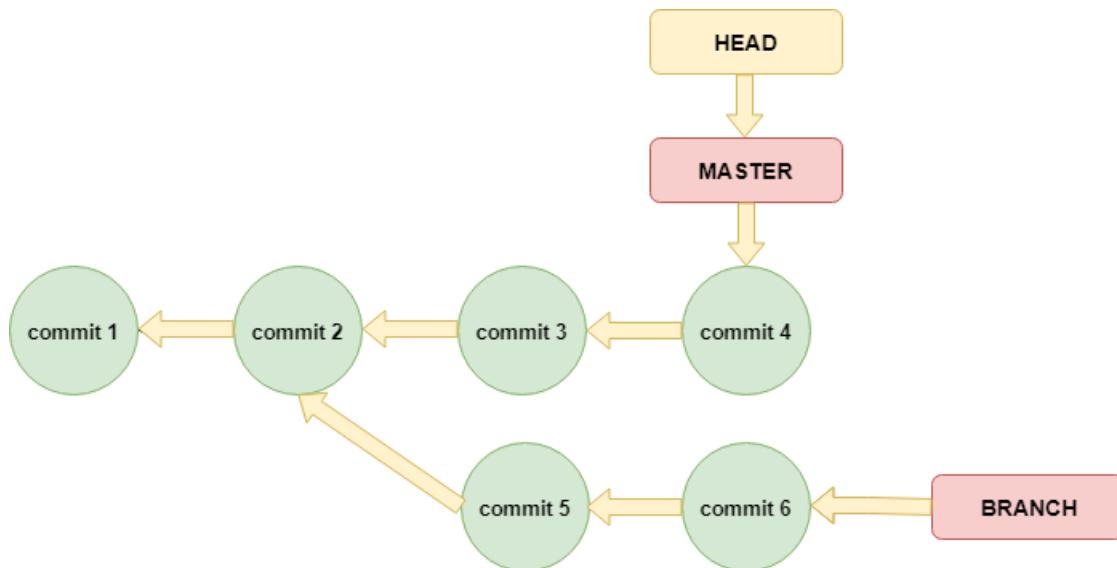
806 Pada subbab ini dijelaskan mengenai analisis penggunaan JGit dan Selenium WebDriver.

### 3.2.1 Analisis Penggunaan JGit

JGit dapat digunakan untuk berinteraksi dengan repositori yang terekam oleh Git. Dimana interaksi ini dapat dilakukan melalui program java. Pada analisis ini dibahas kelas-kelas pada *library* JGit yang digunakan pada program dalam skripsi ini. Kelas-kelas yang dipakai yaitu Git, Repository, FileRepository, dan RevCommit.

Untuk dapat berinteraksi dengan suatu repositori Git diperlukan kelas Repository dan Git. Kelas Repository dan FileRepository merepresentasikan suatu repositori Git. Kelas FileRepository merupakan turunan dari kelas Repository. Operasi-operasi pada Git dapat dilakukan dengan menggunakan kelas Git. *Constructor* dari kelas Git menerima parameter bertipe Repository. Repository bersifat abstrak, karena itu tidak bisa diinisialisasi secara langsung. Repository dapat diinisialisasi menggunakan *object* yang bertipe FileRepository. Dimana *constructor* FileRepository menerima parameter berupa alamat dari direktori Git.

Dibutuhkan beberapa langkah untuk melakukan operasi Git Log. Dari object bertipe Git, dapatkan object bertipe LogCommand dengan memanggil *method* log(). Setelah itu, panggil *method* call() untuk melakukan operasi Git Log. Setelah operasi Git Log dijalankan, akan didapatkan seluruh *commit* pada *branch* yang sedang aktif, dimana seluruh *commit* tersebut berupa *object* yang bertipe Iterable<RevCommit>. Jika histori seperti pada Gambar 3.2 dan saat ini HEAD berada di Master, Iterable<RevCommit> akan berisi commit4, commit3, commit2, dan commit1. Jika saat ini HEAD berada di BRANCH maka Iterable<RevCommit> akan berisi commit6, commit5, commit2, dan commit1. Operasi Git Log ini nantinya dipakai oleh program dalam skripsi ini untuk mendapatkan seluruh histori *commit* dari perangkat lunak berbasis *web*.



Gambar 3.2: Histori *commit* direpresentasikan sebagai Directed Acyclic Graph.

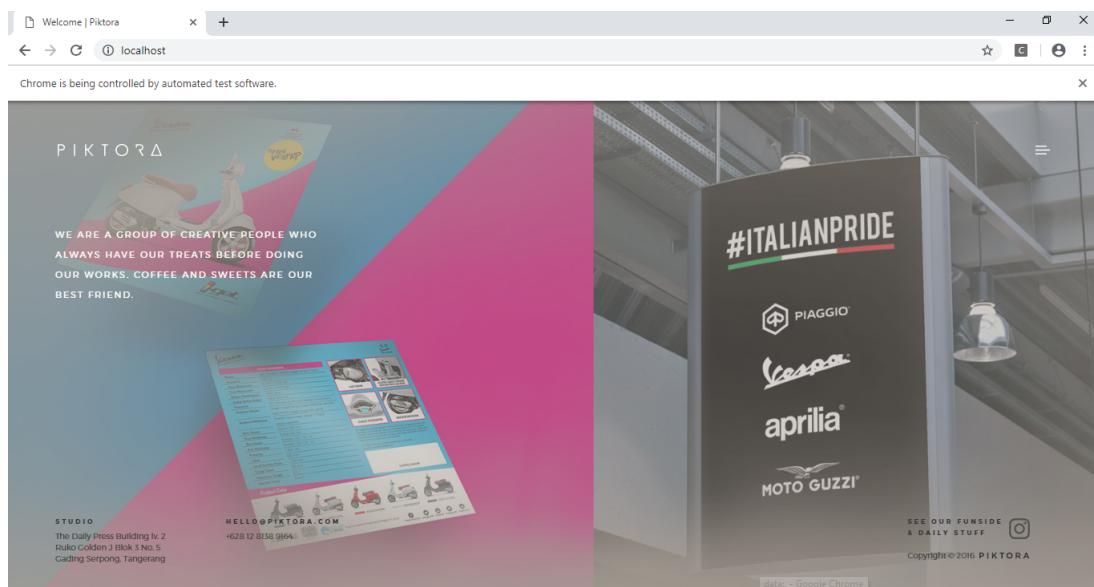
Dibutuhkan beberapa langkah untuk melakukan operasi Git Checkout. Dari object bertipe Git, dapatkan object bertipe CheckoutCommand dengan memanggil *method* checkout(). Setelah itu, panggil *method* setName() diikuti dengan parameter berupa *commit ID*. Kemudian panggil *method* call() untuk menjalankan operasi Git Checkout. Setelah operasi Git Checkout dijalankan, *working directory* akan diperbarui sesuai dengan *state* pada *commit* tertentu. Operasi *checkout* ini nantinya

833 akan digunakan oleh program dalam skripsi ini untuk menelusuri *commit* dan mendapatkan halaman  
 834 *web* dari perangkat lunak berbasis *web*.

### 835 3.2.2 Analisis Penggunaan Selenium WebDriver

836 Selenium WebDriver dapat digunakan untuk mengotomatisasi *web browser*. Pada analisis ini dibahas  
 837 kelas dan *interface* pada *library* Selenium WebDriver yang digunakan dalam program pada skripsi  
 838 ini. Kelas yang dipakai yaitu ChromeDriver, sedangkan *interface* yang dipakai adalah WebDriver  
 839 dan TakeScreenshot.

840 WebDriver merupakan *interface* utama yang digunakan untuk pengujian. WebDriver dapat  
 841 diinisialisasi menggunakan *object* dengan tipe ChromeDriver. Setelah melakukan inisialisasi pada  
 842 WebDriver, *browser* akan dijalankan. Untuk membuka suatu halaman *web*, digunakan *method*  
 843 `get()` dengan parameter alamat URL. Gambar 3.3 menunjukkan Chrome *browser* yang dikontrol  
 844 oleh ChromeDriver. WebDriver ini nantinya akan digunakan oleh program dalam skripsi ini untuk  
 845 membuka halaman *web* pada perangkat lunak berbasis *web*.



Gambar 3.3: *Browser* yang dikontrol oleh ChromeDriver.

846 TakesScreenshot merupakan *interface* yang digunakan untuk menangkap *screenshot* halaman  
 847 *web* pada WebDriver. Kelas ChromeDriver mengimplementasikan *interface* ini. *Method* yang  
 848 digunakan untuk menangkap *screenshot* adalah `getScreenshotAs()`, diikuti dengan parameter bertipe  
 849 *OutputType*. *OutputType* ini merupakan *interface* yang digunakan untuk menentukan tipe *output*  
 850 dari hasil *screenshot*. *OutputType* yang digunakan pada program dalam skripsi ini adalah *File*.  
 851 Tipe ini dipilih karena memudahkan kelas *ImageIO*<sup>1</sup> dalam membaca *image*. Pada *OutputType*  
 852 *BYTES*, byte harus dibaca dulu oleh kelas *ByteArrayInputStream*<sup>2</sup>, kemudian dibaca oleh kelas  
 853 *ImageIO*. Pada *OutputType* *BASE64*, *base64 data* dikonversi menjadi *array of bytes*, setelah itu  
 854 dibaca oleh *ByteArrayInputStream*, kemudian dibaca oleh kelas *ImageIO*. Pada *OutputType* *FILE*,  
 855 *file* dapat langsung dibaca oleh kelas *ImageIO*. TakesScreenshot ini nantinya akan digunakan oleh

<sup>1</sup><https://docs.oracle.com/javase/7/docs/api/javax/imageio/ImageIO.html>

<sup>2</sup><https://docs.oracle.com/javase/7/docs/api/java/io/ByteArrayInputStream.html>

856 program dalam skripsi ini untuk mengambil *screenshot* dari halaman *web* pada perangkat lunak  
857 berbasis *web*.

858 **3.3 Analisis Fitur Aplikasi yang Dibangun**

859 Pada skripsi ini, dibuat sebuah perangkat lunak yang dapat membangun animasi *timelapse* dari  
860 pengembangan proyek perangkat lunak berbasis web. Yang akan dibuat animasinya adalah halaman  
861 web dari perangkat lunak. Jumlah halaman berkisar antara satu sampai empat halaman, tergantung  
862 pada masukan dari *user*. *Output* dari program adalah *file* hasil animasi yang bertipe GIF. Program  
863 dapat menerima masukan dan konfigurasi dari Command Line Option. Setiap *option* harus memiliki  
864 argumen.

865 Berikut ini adalah *command line option* yang akan diimplementasikan pada skripsi ini:

866 1. **-project-path <PATH>**

867 Opsi ini berfungsi untuk mengatur *path* dari direktori yang akan dibuat animasinya. Parameter  
868 dari opsi ini adalah *path* dari proyek perangkat lunak web yang terekam oleh Git. Opsi ini  
869 wajib ada.

870 2. **-before-capture <TERMINAL-COMMAND>**

871 Opsi ini berfungsi untuk menjalankan *terminal command*. *Terminal command* ini dijalankan  
872 sebelum melakukan *screenshot*. Parameter dari opsi ini adalah *terminal command* dalam  
873 bentuk *string* dan harus diapit dengan tanda petik ganda. Opsi ini bersifat opsional. Jika  
874 opsi ini tidak dimasukkan, program tidak akan menjalankan *terminal command*.

875 3. **-capture-url <URL 1>[, URL 2[, URL 3[, URL 4]]]**

876 Opsi ini berfungsi untuk mengatur alamat *url* dari halaman *web*, dimana dilakukan pengambil-  
877 an *screenshot* pada halaman ini. Jumlah halaman yang dicapture bisa lebih dari satu, dengan  
878 jumlah maksimal empat halaman. Parameter dari opsi ini adalah alamat *url* dari halaman  
879 *web*, dengan jumlah maksimal empat alamat *url*. Opsi ini wajib ada.

880 4. **-seconds-per-commit <SECONDS>**

881 Opsi ini berfungsi untuk mengatur durasi munculnya satu *commit* pada animasi. Parameter  
882 dari opsi ini adalah durasi munculnya satu *commit* dalam satuan detik. Parameter harus  
883 berupa bilangan bulat atau riil, dimana nilainya lebih besar dari nol dan kurang dari 656.  
884 Opsi ini bersifat opsional. Jika opsi ini tidak dimasukkan, nilai defaultnya adalah 1 detik.

885 5. **-title <TITLE>**

886 Opsi ini berfungsi untuk memberi judul. Dimana judul tersebut ditampilkan pada pojok kiri  
887 bawah layar. Opsi ini bersifat opsional. Jika opsi ini tidak dimasukkan, maka judul tidak  
888 akan ditampilkan.

889 6. **-logo <IMAGE>**

890 Opsi ini berfungsi untuk memasukkan logo. Dimana logo tersebut ditampilkan pada pojok  
891 kanan bawah layar. Parameternya adalah *path* dari *file* gambar. Format gambar yang  
892 diterima antara lain JPG, PNG, GIF dan, BMP. Opsi ini bersifat opsional. Jika opsi ini tidak  
893 dimasukkan, maka judul tidak akan ditampilkan.

894 7. **-start-commit <COMMIT-ID>**

895 Opsi ini berfungsi untuk mengatur *commit* awal untuk membangkitkan animasi. Parameter  
896 dari opsi ini adalah *commit ID* awal , dimana panjang *commit ID* adalah tujuh sampai  
897 sepuluh karakter. Opsi ini bersifat opsional. Jika opsi ini tidak dimasukkan, program akan  
898 membangkitkan animasi dimulai dari *commit* paling awal.

899 8. **-stop-commit <COMMIT-ID>**

900 Opsi ini berfungsi untuk mengatur *commit* akhir untuk membangkitkan animasi. Parameter  
901 dari opsi ini adalah *commit ID* akhir , dimana panjang *commit ID* adalah tujuh sampai  
902 sepuluh karakter. Opsi ini bersifat opsional. Jika opsi ini tidak dimasukkan, program akan  
903 membangkitkan animasi sampai dengan *commit* paling akhir.

904 9. **-branch <BRANCH>**

905 Opsi ini berfungsi untuk mengatur *branch* dari repositori yang akan dibangkitkan animasinya.  
906 Parameter dari opsi ini adalah nama dari *branch*. Opsi ini bersifat opsional. Jika opsi ini  
907 tidak dimasukkan, program akan membangkitkan animasi pada *branch* master.

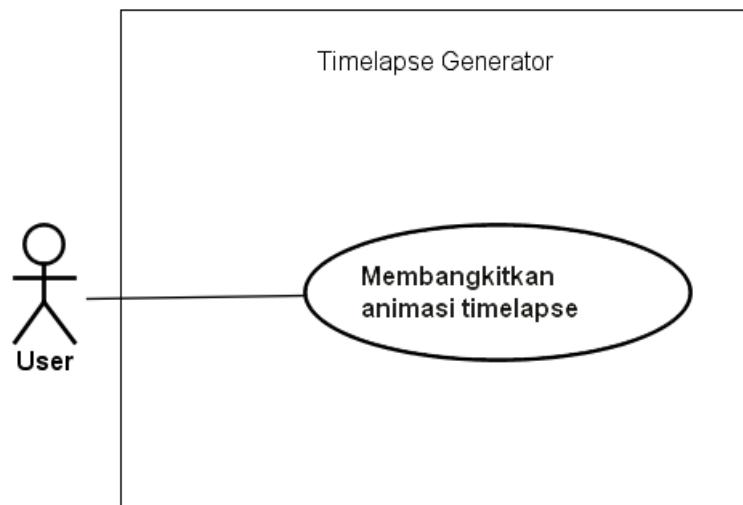
908

909 Opsi **-path**, **-title**, dan **-image** mengacu pada opsi yang terdapat pada Gource. Opsi **-start-commit**  
910 **-stop-commit** dan **-seconds-per-commit** mengacu pada Gource, dengan sedikit penyesuaian.  
911 Opsi **-seconds-per-day** pada Gource menyatakan durasi munculnya satu hari, sedangkan Opsi  
912 **-seconds-per-commit** menyatakan durasi munculnya satu *commit*. Opsi **-start-date** **-stop-date**  
913 pada Gource dan opsi **-start-commit** **-stop-commit** mengatur periode dalam menampilkan ani-  
914 masi. Pada Gource, periode yang digunakan adalah rentang waktu berupa tanggal, bulan, dan  
915 tahun. Sedangkan pada opsi **-start-commit** **-stop-commit**, rentangnya berupa *commit ID*. Opsi  
916 **-before-capture**, **-capture-url**, dan **-branch** tidak mengacu pada Gource. Ketiga opsi tersebut  
917 secara khusus dibuat karena program dalam skripsi ini membuat animasi *timelapse* dari kumpulan  
918 halaman *web*, dimana halaman *web* tersebut diambil dari proyek perangkat lunak berbasis *web* yang  
919 terekam oleh Git. Sedangkan program Gource membuat animasi dari struktur *folder* atau *file*.

920 Program pada skripsi ini memiliki satu fitur. Fitur tersebut adalah membangkitkan animasi  
921 *timelapse*. Penjelasan fitur dapat dilihat pada Gambar 3.4 dan Tabel 3.1.

Nama	Membangkitkan animasi <i>timelapse</i>
Deskripsi	Fitur untuk membangkitkan animasi timelapse dari pengembangan proyek perangkat lunak berbasis web
Aktor	<i>User</i>
Pos-kondisi	File animasi bertipe GIF berhasil dibuat
Skenario	<ol style="list-style-type: none"> <li>1. Sistem membaca masukan <i>input</i> dari Command Line Option.</li> <li>2. Sistem menelusuri histori perkembangan perangkat lunak berbasis <i>web</i> dengan fitur Git. Saat menelusuri histori perkembangan perangkat lunak, sistem mengambil <i>screenshot</i> dari halaman web menggunakan SeleniumWebDriver.</li> <li>3. Sistem menggabungkan file <i>screenshot</i> menjadi satu file bertipe GIF.</li> </ol>

Tabel 3.1: *Scenario case* membangkitkan animasi *timelapse*



Gambar 3.4: *Use case diagram* perangkat lunak.

922

923

924 Langkah-langkah untuk membangkitkan animasi *timelapse* adalah sebagai berikut:

925

- 926 1. Program membaca argumen Command Line menggunakan Apache Commons CLI.
- 927 2. Program mendapatkan seluruh *commit* histori dari proyek perangkat lunak berbasis web  
928 menggunakan JGit.

- 929 3. Program membuka semua *browser* menggunakan Selenium WebDriver. Jumlah *browser*  
930 bergantung pada jumlah argumen dari Option **-capture-url**.
- 931 4. Program melakukan *checkout* dalam suatu rentang *commit* tertentu. Jika tidak terdapat  
932 Option **-start-commit** dan Option **-stop-commit**, akan dilakukan *checkout* ke semua *commit*.
- 933 5. Program melakukan *checkout* ke suatu *commit*.
- 934 6. Program menjalankan *terminal Command* jika terdapat Option **-before-capture**.
- 935 7. Program membuka setiap URL yang didapatkan dari Option **-capture-url** menggunakan  
936 Selenium WebDriver. Setiap *browser* membuka URL yang berbeda.
- 937 8. Selenium WebDriver kemudian mengambil *screenshot* pada semua *browser*.
- 938 9. Jika saat ini sedang berada pada *commit* terakhir, lanjut ke langkah berikutnya. Jika tidak,  
939 ulangi langkah 5-8 untuk *commit* selanjutnya.
- 940 10. Jika terdapat lebih dari satu *browser*, hasil *screenshot* dari setiap *browser* digabungkan menjadi  
941 satu gambar baru.
- 942 11. Program menambahkan judul di pojok kiri bawah jika terdapat Option **-title**.
- 943 12. Program menambahkan logo di pojok kanan bawah jika terdapat Option **-logo**.
- 944 13. Menggabungkan semua *file* gambar menjadi satu *file* bertipe GIF.

945 Penulis menggunakan *library* yang didapatkan dari internet<sup>3</sup> untuk menggabungkan gambar  
946 menjadi *file* GIF.

## 947 3.4 Prapengujian Website Piktora

948 Pengujian dilakukan dengan proyek Piktora sebagai input dari program. Piktora<sup>4</sup> merupakan situs  
949 *web* yang menawarkan layanan *creative design* dan *branding*. Layanan yang ditawarkan berupa  
950 *graphic design* untuk poster, *banner*, *website*, dan aplikasi *mobile*. Repozitori dari situs *web* ini  
951 disimpan pada Gitlab<sup>5</sup>, dibutuhkan akses khusus untuk membuka repositori tersebut. Piktora  
952 dibangun dengan php menggunakan framework Code Igniter. Piktora menggunakan MySQL sebagai  
953 basis datanya.

954 Prapengujian ini hanya menggunakan dua parameter, yaitu project-path dan capture-url(lihat  
955 subbab 3.3). Pengujian dilakukan menggunakan prototipe program yg akan dilengkapi pada Bab 5.  
956 Berikut adalah spesifikasi perangkat keras dan perangkat lunak yang digunakan untuk prapengujian:

- 957 • Processor: Intel Core i3 4030U
- 958 • RAM: 6GB
- 959 • Sistem Operasi: Windows 10 pro 64-bit

<sup>3</sup><http://elliot.kroo.net/software/java/GifSequenceWriter>

<sup>4</sup><http://www.piktora.com>

<sup>5</sup><https://gitlab.com/PNDevworks/Piktora>

- 960     • Versi Apache HTTP Server: 2.4.29  
 961     • Versi MySQL Server: 5.5.5  
 962     • Versi Netbeans: 8.1  
 963     • Versi Google Chrome: 73.0.3683.86

964   Apache HTTP Server digunakan sebagai *web server* lokal. MySQL Server digunakan untuk  
 965   menyimpan basis data secara lokal dan juga digunakan sebagai *database server* lokal.

966   Pada proyek Piktora terdapat 58 *commit*. Listing 3.1 menunjukkan histori *commit* pada proyek  
 967   Piktora, histori didapatkan dengan menjalankan operasi Git Log pada terminal. Setelah dilakukan  
 968   pengujian terdapat beberapa masalah. Masalah tersebut yaitu perbedaan letak *file*, migrasi *database*,  
 969   dan konfigurasi *database*. Masalah-masalah tersebut dibahas pada subbab 3.4.1 sampai 3.4.3.

Listing 3.1: Histori *commit* pada proyek Piktora

```
97b 315d374 - Oct 31 16:52:46 2016 : Basic CI files + htaccess & webconfig + database .
971   php ignore
972
973 27ce3d4 - Nov 5 13:12:43 2016 : setup environment for piktora
974 65f0c9c - Nov 5 19:22:58 2016 : * create structure for all pages * add dummy
975   images
976
977 bffbae1 - Nov 8 18:00:32 2016 : - basic structure (navbar semi complete) - add
978   fonts
979
980 5c59916 - Nov 8 19:51:18 2016 : implement navbar, footer, and projects/ page
981 7738380 - Nov 8 20:05:27 2016 : fix pc and ipad navbar fontsize
982 26bdbbe - Nov 8 20:16:33 2016 : fix position image for desktop /projects
983 3db3ce8 - Nov 9 00:28:27 2016 : - implements project details page - fix some minor
984   issue - add some project image
985
986 5ef34fa - Nov 13 13:01:06 2016 : implement about us (raw version)
987 3caf535 - Nov 15 11:55:15 2016 : fix minor issues view/about: - background-image
988   position. Make it to the center position - slick.js img need to set to inline
989
990 c5eb3b6 - Nov 15 13:02:42 2016 : implement /welcome page
991 ade9216 - Nov 15 13:12:08 2016 : fix minor issues: - move style footer to global -
992   add space before PIKTORA
993
994 c77b5b3 - Nov 18 18:18:25 2016 : implement /contact
995 b42b819 - Nov 18 21:22:22 2016 : change a href to style cursor: pointer
996 3eb7af8 - Nov 21 16:09:40 2016 : .htaccess to be compatible with cloud kilat
997 e87e84b - Nov 22 14:53:45 2016 : - change vw to 100% - add captcha
998 ff8d829 - Nov 22 15:22:53 2016 : Solved captcha font load: use otf instead of ttf
999   Also: create directory assets/img/captcha and ignore everything inside
1000
1001 dc87342 - Nov 22 15:49:03 2016 : - implement captcha code - remove wrong css
1002 9ebe433 - Nov 22 16:12:44 2016 : add scroll feature in project/
1003 f0f7270 - Nov 23 15:17:42 2016 : Added Google PHP Client v2 See https://github.com
1004   /google/google-api-php-client
1005
```

180 | 57a239b – Nov 23 15:19:57 2016 : Merge origin/master  
181 | e2dfebe – Nov 27 14:42:24 2016 : – remove blue outline when click with slick –  
1008 | change background image in about to newest one  
183b  
182 | a19e7f2 – Nov 27 15:24:42 2016 : Detailing dari Edina: 1. Hal. Project Detail ,  
1011 | font coba diperkecil saja mungkin ya. 3. Beberapa ukuran font dan spacing ada  
1012 | yang kurang pas sedikit , terlampir detail revisinya ya (file pdf) 4. Footer  
1013 | dibuat selalu stay terus di bagian bawah dengan posisi yang selalu sama. Di home  
1014 | & contact sudah sama, namun di hal. product posisinya agak lebih naik.  
183  
184 | 3d79d0a – Nov 27 15:29:58 2016 : fix minor issue  
185 | 0fcfd958 – Nov 28 10:11:06 2016 : add raw admin contents  
186 | add3974 – Nov 28 12:03:21 2016 : add summernote , implement read project  
187 | 0680488 – Nov 29 12:38:23 2016 : update admin for projects  
188 | fbe7639 – Nov 29 13:10:30 2016 : implement admin for home  
189 | db0cedd – Nov 29 13:38:29 2016 : – implementasi database bagian user – upload 9  
1022 | gambar contoh project  
140  
141 | 0fe9aaaf – Nov 29 14:17:20 2016 : ubah warna garis captcha  
142 | f2326dd – Nov 29 14:44:51 2016 : – lewati proses otentikasi sementara  
143 | f78cdb4 – Dec 2 12:10:47 2016 : (trying to) fix issue #2  
144 | ef9b62b – Dec 2 17:09:58 2016 : revisi dari edina ke-2  
145 | c689aa8 – Dec 2 17:11:13 2016 : perbaikan admin sedikit  
146 | c4e9576 – Dec 2 17:14:06 2016 : perbaikan di /contact , kelewat  
147 | 02d04f1 – Dec 5 14:55:20 2016 : tambah wording  
148 | a4e4858 – Dec 5 15:08:59 2016 : perbaikan kata2 sedikit  
149 | bbd82c2 – Dec 6 10:41:40 2016 : implementasi email  
150 | f8c64fc – Dec 6 11:03:08 2016 : change to httpdocs  
151 | eb49c2b – Dec 6 11:35:20 2016 : hapus migrasi script di admin  
152 | ace1988 – Dec 6 11:39:00 2016 : change overflow to auto  
153 | 627e65b – Dec 6 11:45:26 2016 : modify database back to local  
154 | 0896f81 – Dec 7 16:08:30 2016 : update home versi mobile jadi baru (revisi dari  
1038 | Edina)  
155  
156 | 5cf1292 – Dec 7 16:21:01 2016 : ubah background di about menjadi tidak pecah  
157 | c83f4aa – Thu Dec 15 15:04:30 2016 : remove piktoria secrets  
158 | 57f5ea4 – Thu Dec 15 15:09:43 2016 : remove unimportant data  
159 | 7931c21 – Dec 24 18:40:41 2016 : edit wording  
160 | 9b0a302 – Dec 25 06:03:50 2016 : Another wording fix  
161 | f1ea410 – Thu Jan 5 15:23:32 2017 : fix instagram link  
162 | 1880a88 – Thu Jan 5 15:24:12 2017 : Merge branch 'master' of https://github.com/  
1047 | pascalalfadian/Piktora  
163  
164 | 286aa78 – Jan 16 12:48:45 2017 : Perbaikan wording di admin edit project  
165 | 33702c2 – Feb 21 13:31:08 2017 : change email sender to piktoria@mailgun.dnartworks.  
1051 | com.au  
166  
167 | 18c39ef – Thu Apr 13 15:21:49 2017 : Test commit (in gitlab). Nothing much important  
168 | 9bfde3c – Apr 17 15:09:54 2017 : add ignore sftp-config.json  
169 | 38711f0 – Apr 17 15:15:03 2017 : fix bug ugly display when projects too high  
170 | 9f041ef – May 15 10:40:16 2017 : set insta url to https://www.instagram.com/  
1057 | piktorastudio/

```

105b
106a 6a085c1 - Dec 12 14:38:38 2017 : Update company address
106b 89000be - Jan 12 12:25:30 2018 : Update new company address

```

### 1061 3.4.1 Perbedaan Letak File

1062 Pada *commit* 315d374 (31 Oktober 2016) s.d. bbd82c2 (6 Desember 2016) halaman *web* proyek  
 1063 Piktora tidak bisa dibuka. Hal ini disebabkan oleh perbedaan letak *file* "index.php". Pada *commit*  
 1064 315d374 (31 Oktober 2016) s.d. bbd82c2 (6 Desember 2016), *file* "index.php" berada pada *folder*  
 1065 "www", sedangkan pada *commit* f8c64fc (6 Desember 2016) s.d. 89000be (12 Januari 2018) *file*  
 1066 "index.php" berada pada *folder* "httpdocs". Akibat adanya perbedaan letak *file* tersebut, maka  
 1067 konfigurasi dari apache harus diubah.

1068 Solusi untuk mengatasi masalah ini adalah dengan menambahkan *command line option* yang akan  
 1069 menjalankan *terminal command*. Dimana *terminal command* ini berisi perintah untuk menjalankan  
 1070 *script* PHP. *Script* PHP ini akan mengecek letak *file* "index.php" pada *folder* "www" dan "httpdocs".  
 1071 *Script* kemudian akan mengecek *directory root* apache pada *file* "httpd.conf". Jika *directory root*  
 1072 sudah mengarah ke *folder* tempat "index.php" berada, maka *script* tidak akan mengubah isi *file*  
 1073 "httpd.conf". Jika *directory root* tidak mengarah ke *folder* tempat "index.php" berada, maka *script*  
 1074 akan mengubah *directory root* pada *file* "httpd.conf" dan melakukan *restart* pada apache. Setelah  
 1075 itu *script* akan melakukan migrasi *database*.

### 1076 3.4.2 Permasalahan Konfigurasi Database

1077 Pada *commit* f8c64fc (6 Desember 2016) s.d. ace1988 (6 Desember 2016), halaman *web* tidak bisa  
 1078 dibuka. Hal ini disebabkan karena perbedaan konfigurasi pada *file* "database.php". Pada *commit*  
 1079 f8c64fc (6 Desember 2016) s.d. 57f5ea4 (15 Desember 2016) dan *commit* f1ea410 (5 Januari 2017),  
 1080 di dalam *file* "database.php" terdapat *password*. *Commit* lainnya tidak terdapat *password* pada *file*  
 1081 "database.php". Penulis menggunakan *password* "piktora" pada konfigurasi *database* di MySQL  
 1082 Server.

Listing 3.2: Isi *file* "database.php" pada *commit* f8c64fc(6 Desember 2016)

```

108b $active_group = 'default';
108a $query_builder = TRUE;
108b
108a $db[ 'default' ] = array(
108b   'dsn' => '',
108a   'hostname' => 'localhost',
108b   'username' => 'piktora',
108a   'password' => 'dmHx64%6',
108a   'database' => 'piktora',
108a   'dbdriver' => 'mysqli',
108b   'dbprefix' => '',
108a   'pconnect' => FALSE,
108a   'db_debug' => (ENVIRONMENT !== 'production'),
108a   'cache_on' => FALSE,
108b   'cachedir' => '',
108a   'char_set' => 'utf8',

```

```

1d97 | 'dbcollat' => 'utf8_general_ci',
1d98 | 'swap_pre' => '',
1d99 | 'encrypt' => FALSE,
1d100 | 'compress' => FALSE,
1d101 | 'stricton' => FALSE,
1d102 | 'failover' => array(),
1d103 | 'save_queries' => TRUE
1d104 );

```

Listing 3.2 merupakan isi dari file "database.php" pada commit f8c64fc (6 Desember 2016). Dapat dilihat bahwa *password* yang terdapat pada file "database.php" adalah dmHx64%6. Commit f8c64fc (6 Desember 2016) s.d. ace1988 (6 Desember 2016) menggunakan *password* dmHx64%6, sedangkan commit 627e65b (6 Desember 2016) s.d. 57f5ea4 (15 Desember 2016) dan f1ea410 (5 Januari 2017) menggunakan *password* piktora. Karena konfigurasi *password* pada file "database.php" dan phpMyAdmin berbeda, halaman website pada commit f8c64fc (6 Desember 2016) s.d. ace1988 (6 Desember 2016) tidak bisa dibuka.

Sama seperti subbab 3.4.1, solusi untuk mengatasi masalah ini adalah dengan menambahkan *command line option* yang akan menjalankan *terminal command*. Dimana *terminal command* ini berisi perintah untuk menjalankan *script PHP*. *Script* ini akan mengecek *password* yang terdapat pada file "database.php". Jika tidak ditemukan *password* atau ditemukan *password* berupa piktora, maka *script* tidak akan mengubah isi file "database.php". Jika ditemukan *password* berupa dmHx64%6, maka *script* akan mengubah *password* menjadi piktora.

### 3.4.3 Permasalahan Migrasi Database

Pada commit 3d79d0a (27 Nov 2016), terjadi *error* saat melakukan migrasi database. Pada commit a19e7f2 (27 November 2016), terdapat satu file untuk melakukan migrasi yaitu "20161122150000\_Structure.php". Pada commit 3d79d0a (27 Nov 2016) terdapat dua file untuk melakukan migrasi yaitu "20161122150000\_Structure.php" dan "20161122150001\_InitialData.php". Pada commit a19e7f2(27 November 2016) file "20161122150001\_InitialData.php" dijalankan saat melakukan migrasi. Versi migrasi *database* menjadi "20161122150000". Pada commit 3d79d0a (27 Nov 2016), file "20161122150000\_Structure.php" tidak dijalankan karena dianggap sama dengan versi migrasi *database* saat ini. Hanya file "20161122150001\_InitialData.php" yang dijalankan pada commit 3d79d0a (27 Nov 2016). Isi file "20161122150001\_InitialData.php" pada commit a19e7f2 (27 November 2016) dan 3d79d0a (27 Nov 2016) berbeda. Hal ini yang menyebabkan terjadinya *error* saat melakukan migrasi database.

Sama seperti subbab 3.4.1, solusi untuk mengatasi masalah ini adalah dengan menambahkan *command line option* yang akan menjalankan *terminal command*. Dimana *terminal command* ini berisi perintah untuk menjalankan *script PHP*. *Script* ini akan melakukan dua pekerjaan. Pertama, *script* akan menghapus *database* piktora. Setelah itu *script* akan membuat *database* piktora, kemudian melakukan migrasi *database*.

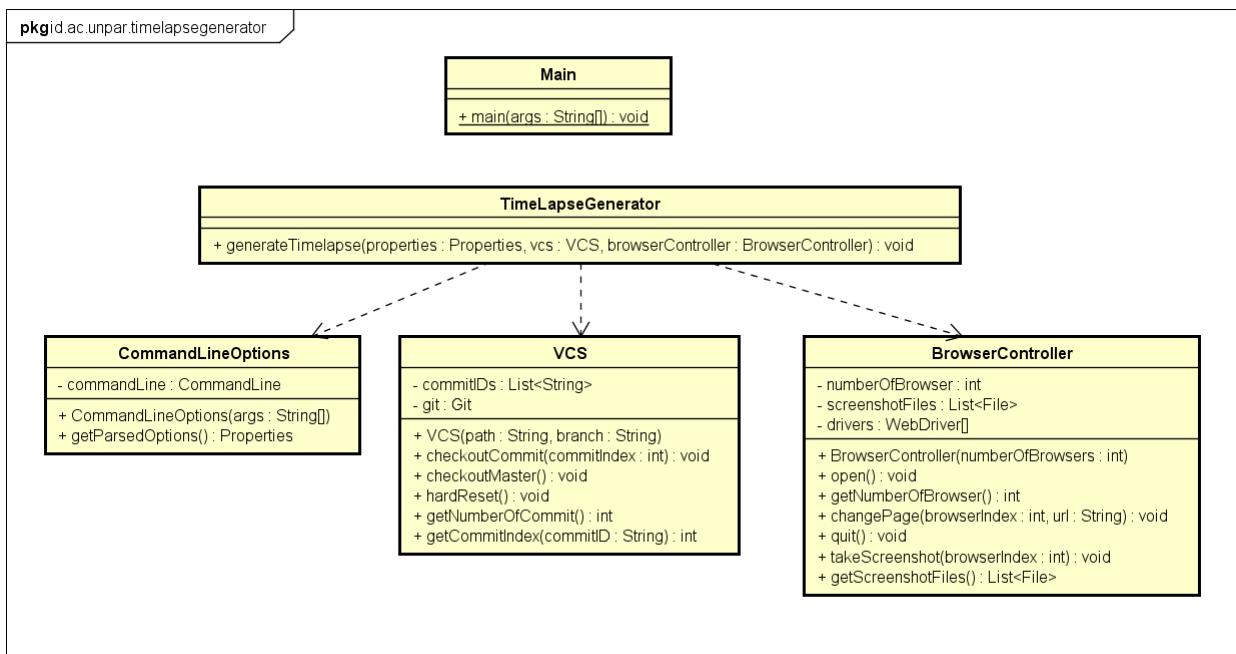


## BAB 4

## PERANCANGAN

1139 Pada bab ini dijelaskan mengenai perancangan perangkat lunak yang dibangun, meliputi perancangan kelas dan perancangan antarmuka.

1141 **4.1 Perancangan Kelas**



Gambar 4.1: Diagram kelas.

1142 Program pada skripsi ini memiliki lima kelas. Diagram kelas pada program ini dapat dilihat pada  
1143 Gambar 4.1. Berikut adalah rincian kelas yang terdapat pada program ini:

1144 • BrowserController

1145 Kelas ini digunakan untuk mengatur *browser*. Operasi-operasi yang dilakukan terhadap  
1146 *browser* yaitu membuka *browser*, mengambil *screenshot*, membuat *browser window* menjadi  
1147 maksimal, dan menutup *browser*. Berikut adalah atribut yang terdapat pada kelas ini:

1148 – private final WebDriver[] drivers

1149 Atribut ini adalah kumpulan *browser* yang digunakan untuk keperluan *automation testing*.

- private final List<File> screenshotFiles  
Atribut ini berfungsi untuk menyimpan *file* hasil *screenshot*.
- private final int numberOfBrowser  
Atribut ini menyatakan jumlah *browser* yang dimiliki oleh kelas ini. Jumlah maksimal dari *browser* adalah empat.

Berikut adalah *method* yang terdapat pada kelas ini:

- public BrowserController(int numberOfBrowsers)  
Constructor dari kelas ini. Berfungsi untuk menginisialisasi atribut yang dimiliki oleh kelas ini. Parameternya adalah jumlah *browser* yang dapat dimiliki oleh kelas ini.
- public void open()  
Berfungsi untuk membuka semua *browser*, kemudian mengatur ukuran *browser window* menjadi maksimal.
- public int getNumberOfBrowser()  
Berfungsi untuk mengembalikan jumlah browser yang dimiliki kelas ini.
- public void changePage(int browserIndex, String url)  
Berfungsi untuk berpindah halaman pada *browser* tertentu. Parameternya adalah alamat URL untuk berpindah halaman dan indeks *browser* yang akan diubah halamannya.
- public void quit()  
Berfungsi untuk menutup semua *browser*.
- public void takeScreenshot(int browserIndex)  
Berfungsi untuk mengambil *screenshot* pada *browser* tertentu dan menyimpannya ke atribut *screenshotFiles*. Parameternya adalah indeks *browser* yang akan diambil *screenshotnya*.

- **CommandLineOptions**

Kelas ini berfungsi untuk menyimpan semua Option yang terdapat dalam program ini, dan melakukan *parsing* argumen Command Line Options yang dimasukkan oleh *user*.

Berikut adalah atribut yang terdapat pada kelas ini:

- private final CommandLine commandLine  
Atribut ini berfungsi untuk melakukan *parsing* argumen Command Line Options dan menampung hasilnya.

Berikut adalah *method* yang terdapat pada kelas ini:

- public CommandLineOptions(String[] args)  
Merupakan Constructor dari kelas ini. Berfungsi untuk menentukan Option yang terdapat pada program dan melakukan parsing argumen Command Line. Parameternya adalah argumen Command Line Option yang didapatkan dari kelas Main.
- public Properties getParsedOptions()  
Berfungsi untuk mengembalikan Command Line Option yang sudah diparsing.

1188 • VCS

1189 Kelas ini digunakan untuk berinteraksi pada proyek perangkat lunak yang terekam oleh Git.  
1190 Berikut adalah atribut yang terdapat pada kelas ini:

- 1191 – private final Git git

1192 Atribut ini digunakan untuk melakukan interaksi pada proyek perangkat lunak yang  
1193 terekam oleh Git.

- 1194 – private final List<String> commitIDs

1195 Atribut ini digunakan untuk menampung seluruh *commit* ID dari hasil penelusuran  
1196 histori.

1197 Berikut ini adalah *method* yang terdapat dalam kelas ini:

- 1198 – public VCS(String path, String branch)

1199 Constructor dari kelas ini. Berfungsi untuk menginisialisasi variabel git dan mendapatkan  
1200 seluruh histori commit proyek perangkat lunak berbasis web pada branch tertentu.  
1201 Dimana branch tersebut diambil dari parameter constructor. Parameternya adalah *path*  
1202 dari proyek perangkat lunak berbasis web dan nama *branch* yang digunakan untuk  
1203 membangkitkan animasi.

- 1204 – public void checkoutCommit(int commitIndex)

1205 Berfungsi untuk melakukan *checkout* ke *commit* tertentu. Parameter dari *method* ini  
1206 adalah indeks dari variabel commitIDs.

- 1207 – public void checkoutMaster()

1208 Berfungsi untuk melakukan *checkout* ke *commit* terakhir.

- 1209 – public void hardReset()

1210 Berfungsi untuk melakukan operasi Git Reset dengan tipe *hard reset*. Operasi ini  
1211 menghapus perubahan pada *working tree* di *commit* tertentu.

- 1212 – public int getNumberOfCommit()

1213 Berfungsi untuk mendapatkan jumlah *commit*.

- 1214 – public int getCommitIndex(String commitID)

1215 Berfungsi untuk mendapatkan indeks dari variabel commitIDs. Parameternya adalah  
1216 *Commit* ID yang akan dicari indeksnya.

1217 • TimeLapseGenerator

1218 Kelas ini digunakan untuk membangkitkan animasi *timelapse*. Berikut adalah *method* yang  
1219 dimiliki oleh kelas ini:

- 1220 – public void generateTimelapse(Properties properties, VCS vcs, BrowserController bro-  
1221 wserController)

1222 Berfungsi untuk membangkitkan animasi *timelapse* berdasarkan langkah-langkah pada  
1223 subbab 3.3. Hasil animasi berupa *file* bertipe GIF. Parameternya adalah objek yang  
1224 bertipe VCS, BrowserController, dan Properties. Parameter properties menampung key  
1225 dan value dari Option yang sudah diparsing.

## 4.2 Perancangan Antarmuka

Program dalam skripsi ini menggunakan dengan terminal sebagai antarmuka, dengan kata lain menggunakan Command Line Interface. *Input* dari program ini dimasukkan melalui argumen Command Line. *Option* yang dapat dimasukkan ke program ini dapat dilihat pada subbab 3.3. *Output* dari program ini berupa status pada terminal dan *file* hasil animasi bertipe GIF.

Listing 4.1: Status pesan yang akan muncul pada terminal saat program berhasil membangkitkan animasi *timelapse*.

Animasi timelapse berhasil dibuat

Listing 4.2: Status pesan yang akan muncul pada terminal saat program gagal membangkitkan animasi *timelapse*.

Animasi timelapse gagal dibuat  
<ERROR MESSAGE>

Listing 4.1 menunjukkan status yang akan ditampilkan pada terminal saat program berhasil membangkitkan animasi *timelapse*. Listing 4.2 menunjukkan status yang akan ditampilkan pada terminal saat program gagal membangkitkan animasi *timelapse*. Pada Listing 4.2, baris pertama menunjukkan bahwa program gagal membangkitkan animasi *timelapse*. Pada Listing 4.2, baris kedua menyatakan *error message* dari program. Berikut ini adalah *error message* yang akan ditampilkan saat *user* memasukkan *input* yang tidak valid:

- Path proyek tidak valid

Pesan ini muncul jika *path* pada argumen Option *project-path* tidak valid.

- Jumlah url yang akan dicapture maksimal 4

Pesan ini muncul jika jumlah argumen pada Option *capture-url* lebih dari 4.

- Seconds per commit harus besar dari 0

Pesan ini muncul jika argumen dari *seconds-per-commit* bernilai lebih kecil dari 0.

- Seconds per commit harus kurang dari sama dengan 655

Pesan ini muncul jika argumen dari *seconds-per-commit* bernilai lebih besar dari 655.

- Seconds per commit harus berupa bilangan riil atau bilangan bulat

Pesan ini muncul jika argumen dari *seconds-per-commit* bukan bertipe bilangan riil atau bilangan bulat.

- Terminal Command tidak valid

Pesan ini muncul jika *terminal command* pada argumen Option *before-capture* tidak valid.

- Path gambar tidak valid

Pesan ini muncul jika *path* gambar pada argumen Option *logo* tidak ditemukan atau *path* gambar bukan merupakan *file* gambar.

- Panjang commit ID awal harus berada di antara 7-40 karakter

Pesan ini muncul jika panjang Commit ID pada argumen Option *start-commit* kurang dari tujuh karakter atau lebih besar dari empat sepuluh karakter.

- 1259     • Panjang commit ID akhir harus berada di antara 7-40 karakter

1260       Pesan ini muncul jika panjang Commit ID pada argumen Option **stop-commit** kurang dari  
1261       tujuh karakter atau lebih besar dari empat sepuluh karakter.

- 1262     • Commit ID awal tidak ditemukan

1263       Pesan ini muncul jika Commit ID pada argumen Option **start-commit** tidak ditemukan.

- 1264     • Commit ID akhir tidak ditemukan

1265       Pesan ini muncul jika Commit ID pada argumen Option **stop-commit** tidak ditemukan.

- 1266     • Commit ID awal dan akhir tidak boleh sama

1267       Pesan ini muncul jika argumen pada Option **start-commit** dan Option **stop-commit** bernilai  
1268       sama.

- 1269     • Commit ID awal dan akhir terbalik

1270       Pesan ini muncul jika nilai argumen pada Option **start-commit** dan Option **stop-commit**  
1271       tertukar.

- 1272     • Branch tidak valid

1273       Pesan ini muncul jika nama branch pada argumen Option **branch** tidak valid.

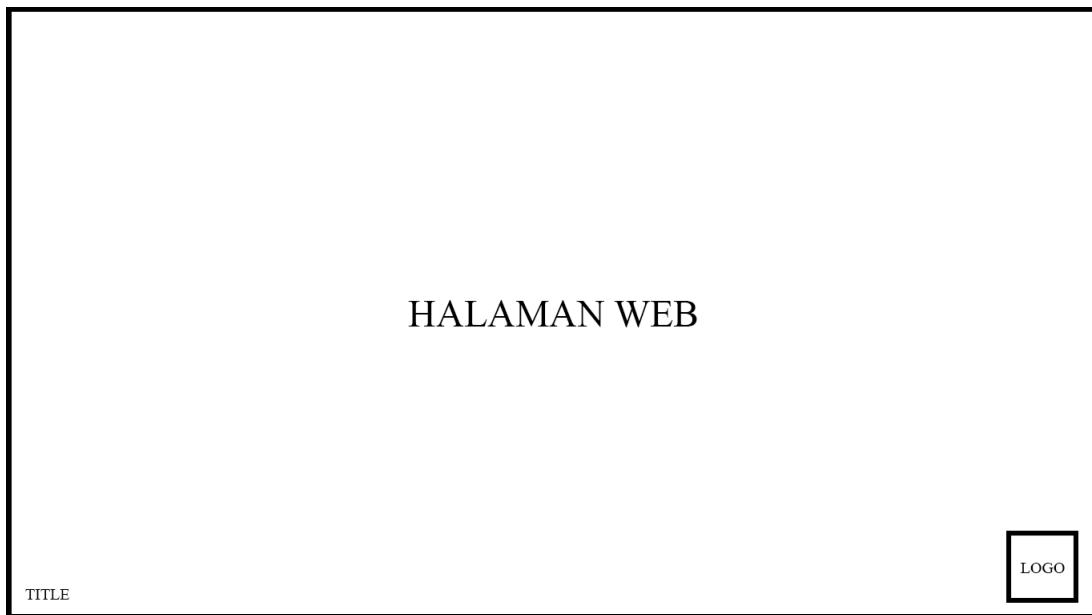
1274       Rancangan *output* dari hasil file animasi *timelapse* dapat dilihat pada Gambar 4.2 sampai dengan

1275       Gambar 4.5. Gambar 4.2 menunjukkan rancangan *output* jika terdapat satu halaman *web*. Gambar

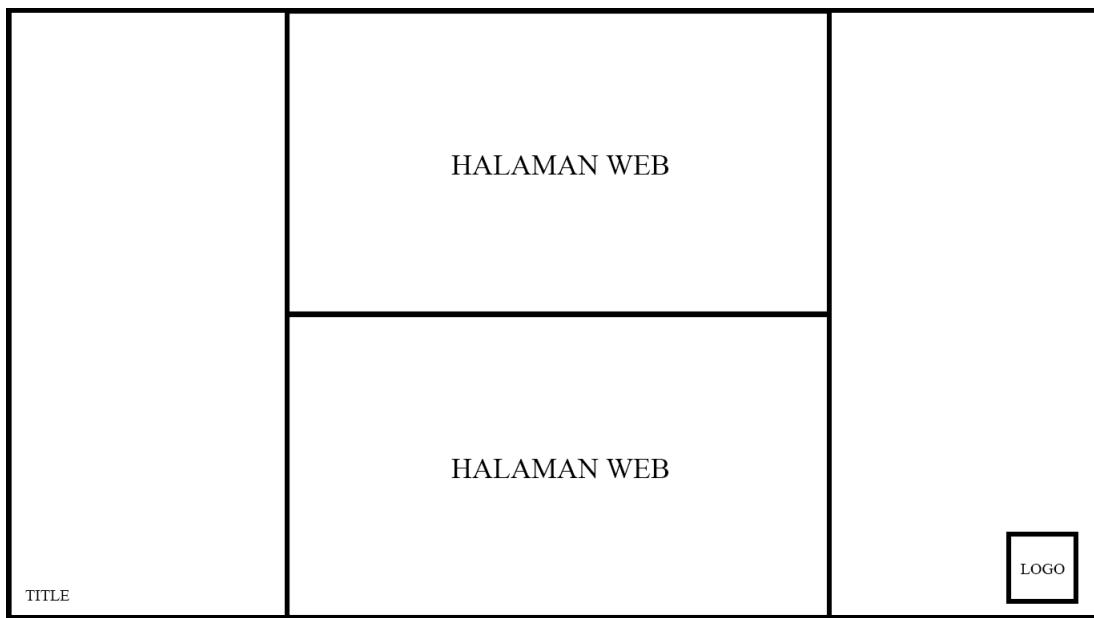
1276       4.3 menunjukkan rancangan *output* jika terdapat dua halaman *web*. Gambar 4.4 menunjukkan

1277       rancangan *output* jika terdapat tiga halaman *web*. Gambar 4.5 menunjukkan rancangan *output* jika

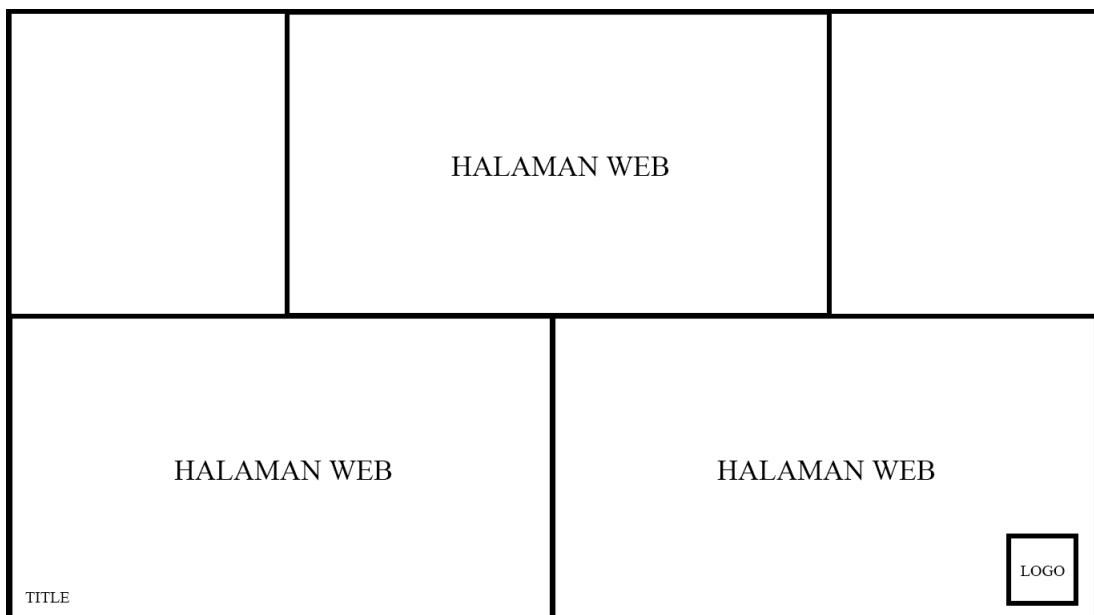
1278       terdapat empat halaman *web*.



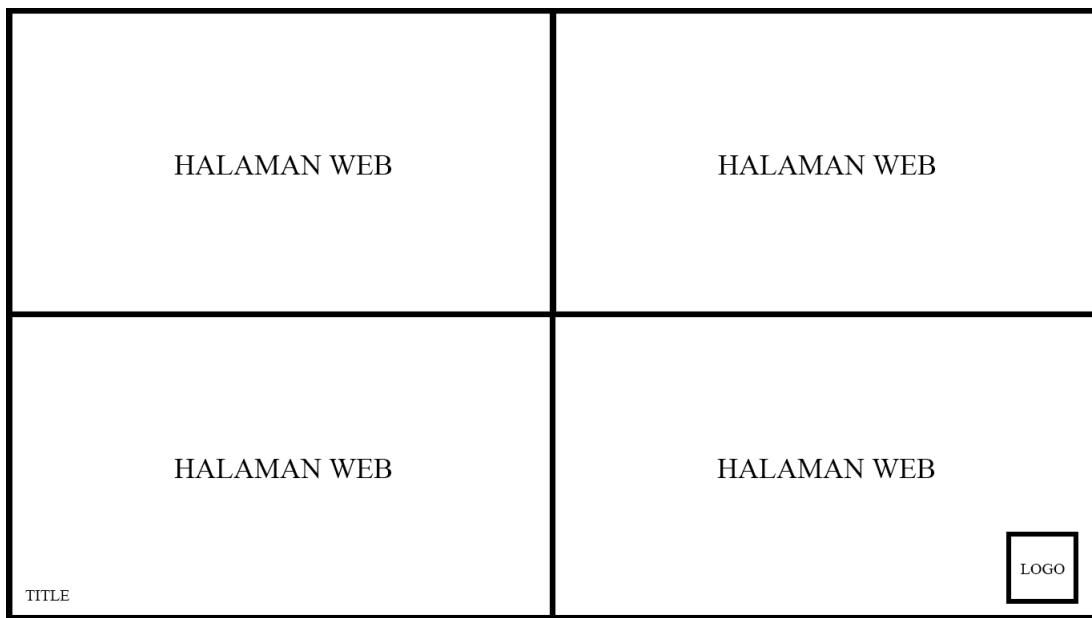
Gambar 4.2: Rancangan *output* jika terdapat satu halaman *web*.



Gambar 4.3: Rancangan *output* jika terdapat dua halaman *web*.



Gambar 4.4: Rancangan *output* jika terdapat tiga halaman *web*.



Gambar 4.5: Rancangan *output* jika terdapat empat halaman *web*.



1279

## BAB 5

1280

### IMPLEMENTASI DAN PENGUJIAN

1281 Pada bab dijelaskan mengenai implementasi perangkat lunak dan pengujian perangkat lunak.  
1282 Bagian implementasi berisi tentang lingkungan implementasi dan hasil implementasi. Bagian  
1283 pengujian berisi tentang pengujian fungsional dan pengujian eksperimental.

1284 **5.1 Implementasi**

1285 **5.1.1 Lingkungan Implementasi**

1286 Implementasi dari perangkat lunak dilakukan pada sebuah laptop. Berikut adalah spesifikasi laptop  
1287 dan perangkat lunak yang digunakan untuk prapengujian:

- 1288 • Processor: Intel Core i3 4030U
- 1289 • RAM: 6GB
- 1290 • Sistem Operasi: Windows 10 pro 64-bit
- 1291 • Versi Apache HTTP Server: 2.4.29
- 1292 • Versi MySQL Server: 5.5.5
- 1293 • Versi Netbeans: 8.1
- 1294 • Versi Google Chrome: 73.0.3683.86

1295 **5.1.2 Hasil Implementasi**

1296 Hasil dari implementasi adalah sebuah perangkat berbasis terminal yang dapat membangkitkan  
1297 animasi *timelapse* pada pengembangan proyek perangkat lunak berbasis *web*. Kode program  
1298 dari perangkat lunak dapat dilihat pada Lampiran A. Setelah dijalankan, perangkat lunak akan  
1299 menghasilkan dua *output* yaitu, status pada terminal dan *file* hasil animasi bertipe GIF.

1300 **1. Status pada Terminal**

1301 Setelah berhasil membangkitkan animasi *timelapse*, perangkat lunak menampilkan status pada  
1302 terminal seperti yang diperlihatkan pada Listing 5.1. Baris 5 menunjukkan bahwa animasi  
1303 *timelapse* berhasil dibangkitkan. Pesan pada baris 1-4 muncul saat ChromeDriver membuka  
1304 dan mulai mengontrol Chrome *browser*.

Listing 5.1: Status pesan pada terminal saat program berhasil membangkitkan animasi *timelapse*.

```
1305 1 Starting ChromeDriver 2.42.591088 (7b2b2dca23cca0862f674758c9a3933e685c27d5) on
1306   port 16446
1307 2 Only local connections are allowed.
1308 3 Feb 24, 2019 3:26:25 PM org.openqa.selenium.remote.ProtocolHandshake
1309   createSession
1310 4 INFO: Detected dialect: OSS
1311 5 Animasi timelapse berhasil dibuat
```

## 1312 2. **File GIF Hasil Animasi**

1313 Selain menghasilkan status pada terminal, program juga akan menghasilkan sebuah *file* GIF  
1314 hasil animasi. Gambar 5.1 menunjukkan *screenshot* setiap *commit* yang terdapat pada *file*  
1315 GIF hasil animasi dari proyek Piktora. Piktora memiliki 58 *commit*, sehingga terdapat 58  
1316 *screenshot*.



Gambar 5.1: *Screenshot* proyek Piktora pada *commit* 315d374 (31 Oktober 2016) - *commit* 89000be (12 Januari 2018).

## <sup>1317</sup> 5.2 Pengujian

### <sup>1318</sup> 5.2.1 Pengujian Fungsional

<sup>1319</sup> Pengujian ini dilakukan dengan tujuan untuk mengetahui apakah Command Line Option yang  
<sup>1320</sup> terdapat pada program sudah berjalan dengan baik. Option yang terdapat pada program dapat  
<sup>1321</sup> dilihat pada subbab 3.3. Lingkungan pengujian fungsional sama dengan lingkungan implementasi

yang terdapat pada subbab 5.1.1. Pengujian dilakukan pada proyek Piktora dengan menggunakan berbagai macam tes kasus. Histori *commit* proyek Piktora dapat dilihat pada subbab 3.4. Hasil pengujian fungsional dapat dilihat pada Tabel 5.1 dan Tabel 5.2.

1325

1326

1327



Gambar 5.2: Salah satu *commit* yang terdapat pada *file* hasil animasi. Terdapat judul dibagian pojok kiri bawah.



Gambar 5.3: Salah satu *commit* yang terdapat pada *file* hasil animasi. Terdapat logo dibagian pojok kanan bawah.



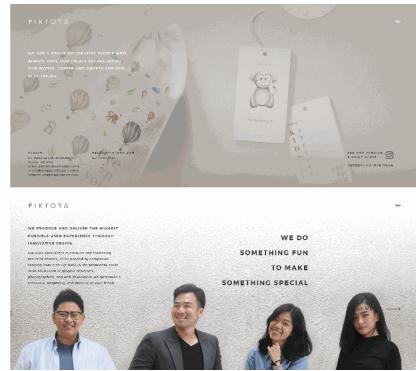
Gambar 5.4: Salah satu *commit* pada *file* hasil animasi jika terdapat satu argumen `-capture-url`.

Tabel 5.1: Tabel pengujian fungsional

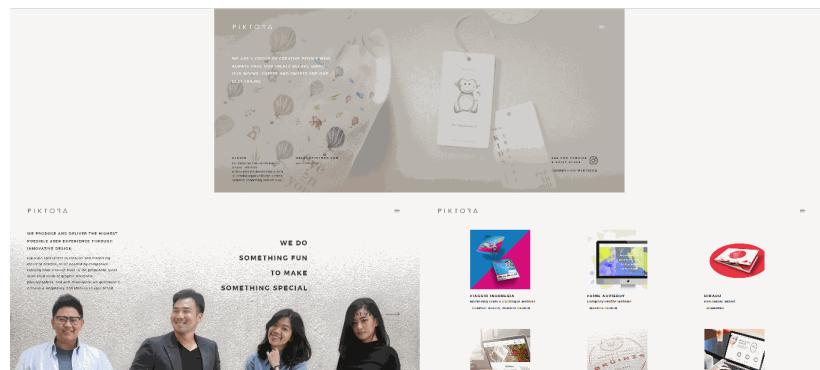
No	Tes Kasus	<i>Output</i> yang diharapkan	<i>Output</i> program
1.	-project-path C:/xampp/htdocs/Piktora/.gi	Program berhasil membangkitkan animasi berdasarkan path yang diberikan dan mengeluarkan status pesan "Animasi timelapse berhasil dibuat"	sesuai
2.	-project-path C:/xampp/htdocs	Program gagal membuat animasi dan menge luarkan pesan error "Path proyek tidak valid"	sesuai
3.	-capture-url http://localhost	Program mengambil screenshot berdasarkan alamat yang diberikan. Tampilan dari file hasil animasi dapat dilihat pada Gambar 5.4	sesuai
4.	-capture-url http://localhost http://localhost/about	Program mengambil screenshot berdasarkan alamat-alamat yang diberikan. Tampilan dari file hasil animasi dapat dilihat pada Gambar 5.5	sesuai
5.	-capture-url http://localhost http://localhost/about http://localhost/projects	Program mengambil screenshot berdasarkan alamat-alamat yang diberikan. Tampilan dari file hasil animasi dapat dilihat pada Gambar 5.6	sesuai
6.	-capture-url http://localhost http://localhost/about http://localhost/projects http://localhost/contact	Program mengambil screenshot berdasarkan alamat-alamat yang diberikan. Tampilan dari file hasil animasi dapat dilihat pada Gambar 5.7	sesuai
7.	-capture-url http://localhost http://localhost/about http://localhost/projects http://localhost/contact http://localhost/projects/1	Program gagal membuat animasi dan mengeluarkan pesan error "Jumlah url yang akan dicapture maksimal 4"	sesuai
8.	-seconds-per-commit 2	Program berhasil membangkitkan animasi dan menghasilkan file GIF yang memiliki durasi 116 detik	sesuai
9.	-seconds-per-commit 0	Program gagal membuat animasi dan mengeluarkan pesan error "Seconds per commit harus lebih besar dari 0"	sesuai
10.	-seconds-per-commit 656	Program gagal membuat animasi dan mengeluarkan pesan error "Seconds per commit harus kurang dari sama dengan 655"	sesuai
11.	-before-capture "php script_piktora.php"	Program menjalankan terminal command sebelum mengambil screenshot	sesuai
12.	-before-capture "php "	Program gagal membuat animasi dan mengeluarkan pesan error "Terminal Command Tidak Valid"	sesuai
13.	-title Piktora	Program berhasil membangkitkan animasi dan menghasilkan file GIF, dimana di dalam file tersebut terdapat judul yang terletak di pojok kiri bawah(lihat Gambar 5.2).	sesuai

Tabel 5.2: Tabel pengujian fungsional

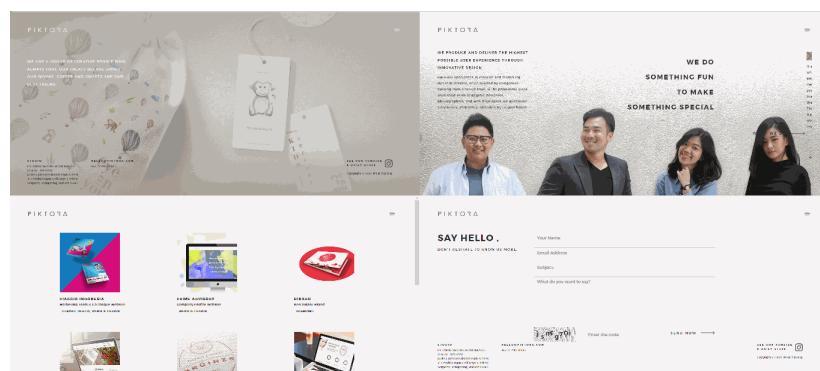
No	Tes Kasus	<i>Output</i> yang diharapkan	<i>Output</i> program
14.	-logo Logo-UNPAR.png	Program berhasil membangkitkan animasi dan menghasilkan file GIF, dimana di dalam file tersebut logo yang terletak di pojok kanan bawah (lihat Gambar 5.3)	sesuai
15.	-logo script_piktora.php	Program gagal membuat animasi dan mengeluarkan pesan error "Path gambar tidak valid"	sesuai
16.	-branch master	Program membangkitkan animasi pada branch master	sesuai
17.	-branch piktora	Program gagal membuat animasi dan mengeluarkan pesan error "Branch tidak valid"	sesuai
18.	-start-commit 9b0a302	Program membangkitkan animasi dimulai dari commit 9b0a302	sesuai
19.	-start-commit 9b0a30	Program gagal membuat animasi dan mengeluarkan pesan error "Panjang commit ID awal harus berada di antara 7-40 karakter"	sesuai
20.	-start-commit 9b0a3023ac8c94c67f4c1b39388277768a488sha9	Program gagal membuat animasi dan mengeluarkan pesan error "Panjang commit ID awal harus berada di antara 7-40 karakter"	sesuai
21.	-stop-commit 6a085c1	Program membangkitkan animasi dimulai dari commit 315d374 sampai dengan commit 6a085c1	sesuai
22.	-stop-commit 6a085c	Program gagal membuat animasi dan mengeluarkan pesan error "Panjang commit ID akhir harus berada di antara 7-40 karakter"	sesuai
23.	-stop-commit 6a085c1c37949e6308cf06a117802e528388e549	Program gagal membuat animasi dan mengeluarkan pesan error "Panjang commit ID akhir harus berada di antara 7-40 karakter"	sesuai
24.	-start-commit 9b0a302 -stop-commit 6a085c1	Program membangkitkan animasi dimulai dari commit 9b0a302 sampai dengan commit 6a085c1	sesuai
25.	-start-commit 6a085c1 -stop-commit 9b0a302	Program gagal membuat animasi dan mengeluarkan pesan error "Commit ID awal dan akhir terbalik"	sesuai
26.	-start-commit 9b0a302 -stop-commit 9b0a302	Program gagal membuat animasi dan mengeluarkan pesan error "Commit ID awal dan akhir tidak boleh sama"	sesuai



Gambar 5.5: Salah satu *commit* pada file hasil animasi jika terdapat dua argumen `-capture-url`.



Gambar 5.6: Salah satu *commit* pada file hasil animasi jika terdapat tiga argumen `-capture-url`.



Gambar 5.7: Salah satu *commit* pada file hasil animasi jika terdapat empat argumen `-capture-url`.

### 1328 5.2.2 Pengujian Eksperimental

1329 Pengujian eksperimental ini dibagi menjadi dua bagian. Pengujian eksperimental bagian pertama  
 1330 akan menguji program menggunakan proyek Piktora dengan WebDriver yang berbeda. WebDriver  
 1331 yang digunakan untuk pengujian ini yaitu FirefoxDriver, EdgeDriver, OperaDriver, dan Interne-  
 1332 tExplorerDriver. Pengujian eksperimental bagian kedua akan menguji progam dengan situs web  
 1333 Bootstrap<sup>1</sup> dan Netflix Open Source Software Center<sup>2</sup>.

<sup>1</sup><https://getbootstrap.com/>

<sup>2</sup><https://netflix.github.io/>

1334 Seperti yang sudah dijelaskan pada subbab 2.3 FirefoxDriver, EdgeDriver, OperaDriver, dan  
 1335 InternetExplorerDriver merupakan implementasi dari WebDriver yang mengontrol *browser* yang  
 1336 mengontrol *browser* tertentu. FirefoxDriver mengontrol Firefox *browser*, OperaDriver mengontrol  
 1337 Opera *browser*, dst. Tujuan dari digunakannya berbagai macam WebDriver ini adalah untuk menguji  
 1338 program dalam membangkitkan animasi menggunakan berbagai macam *browser*.

1339 Pengujian kedua ini dilakukan dengan tujuan menguji program dengan berbagai situs *web* yang  
 1340 berbeda dan memiliki jumlah *commit* yang berbeda. Situs *web* Bootstrap dan Netflix Open Source  
 1341 Software Center dipilih karena bersifat Open Source, repositorinya tersimpan pada Github, dan  
 1342 dihosting menggunakan Github Pages. Github Pages<sup>3</sup> merupakan sebuah layanan *hosting* situs *web*  
 1343 statis untuk melakukan *host* halaman *web* secara langsung dari repositori Github.

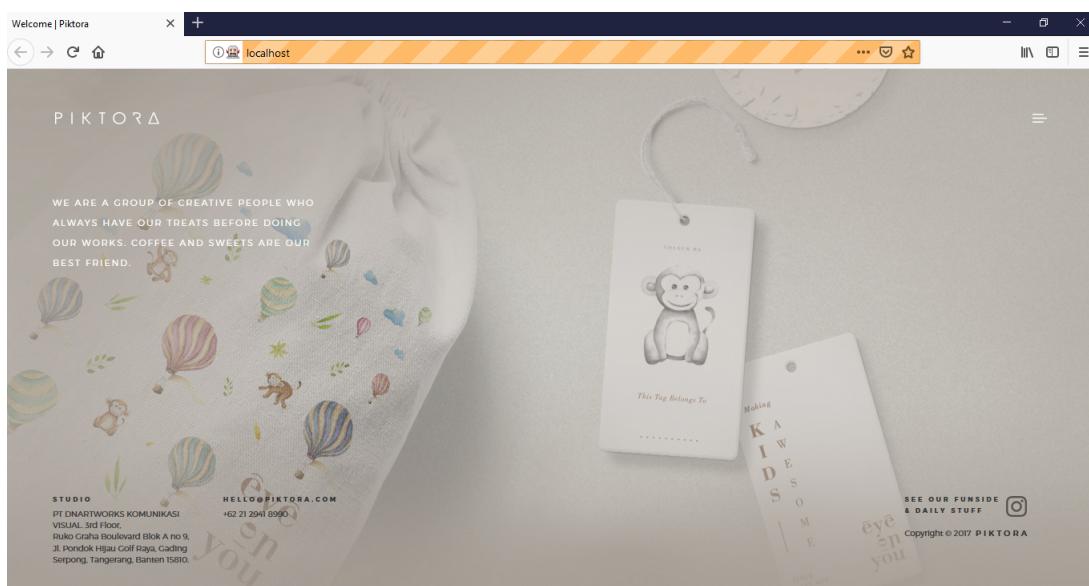
1344 Berikut adalah rincian dari pengujian eksperimental:

### 1345 1. Pengujian Proyek Piktora dengan FirefoxDriver

1346 Pengujian pada proyek Piktora dilakukan menggunakan FirefoxDriver. Pada saat melakukan  
 1347 pengujian, dilakukan sedikit perubahan kode program pada kelas BrowserController baris  
 1348 ke-46 (lihat Lampiran A). *Object* bertipe WebDriver dinisialisasi menggunakan *object* bertipe  
 1349 FirefoxDriver. Versi Firefox *browser* yang digunakan untuk pengujian adalah 66.0.2. Gambar  
 1350 5.8, menunjukkan tampilan Firefox *browser* saat dikontrol oleh FirefoxDriver. Berikut ini  
 1351 adalah *Option* yang digunakan untuk menguji program:

- 1352 • -project-path C:/xampp/htdocs/Piktora/.git
- 1353 • -capture-url http://localhost
- 1354 • -before-capture "php script\_piktora.php"

1355 Program berhasil membangkitkan animasi *timelapse* pada proyek Piktora menggunakan  
 1356 FirefoxDriver.



Gambar 5.8: Tampilan pada Firefox *browser* saat dikontrol oleh FirefoxDriver.

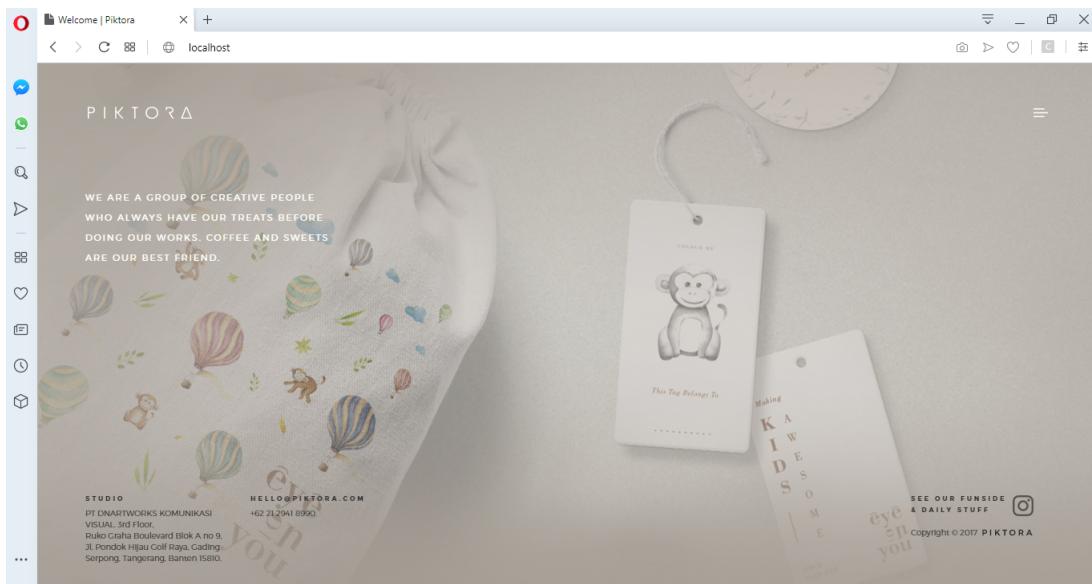
<sup>3</sup><https://pages.github.com/>

## 1357 2. Pengujian Proyek Piktora dengan OperaDriver

1358 Pengujian pada proyek Piktora dilakukan menggunakan OperaDriver. Pada saat melakukan  
 1359 pengujian, dilakukan sedikit perubahan kode program pada kelas BrowserController baris  
 1360 ke-46 (lihat Lampiran A). *Object* bertipe WebDriver dinisialisasi menggunakan *object* bertipe  
 1361 OperaDriver. Versi Opera *browser* yang digunakan untuk pengujian adalah 60.0.3255.27  
 1362 (portable version). Gambar 5.9 menunjukkan tampilan pada Opera *browser* saat dikontrol  
 1363 oleh OperaDriver. Berikut ini adalah *Option* yang digunakan untuk menguji program:

- 1364 • -project-path C:/xampp/htdocs/Piktora/.git
- 1365 • -capture-url http://localhost
- 1366 • -before-capture "php script\_piktora.php"

1367 Terdapat masalah saat melakukan pengujian. Awalnya Opera *browser* yang digunakan bukan  
 1368 versi *portable* melainkan versi standar. Saat program dijalankan, program mengeluarkan  
 1369 pesan *error* berupa *unknown error: cannot find Opera binary*. Setelah ditelusuri, tidak  
 1370 ditemukan *file* "opera.exe" di dalam direktori "C:/Program Files" atau "C:/Program Files  
 1371 (x86)". Solusi untuk mengatasi masalah ini adalah melakukan instalasi Opera *browser* versi  
 1372 *portable* pada direktori "C:/Program Files" atau "C:/Program Files (x86)". Setelah dilakukan  
 1373 instalasi tersebut, program berhasil membangkitkan animasi *timelapse* pada proyek Piktora  
 1374 menggunakan OperaDriver.



Gambar 5.9: Tampilan pada Opera *browser* saat dikontrol oleh OperaDriver.

## 1375 3. Pengujian Proyek Piktora dengan EdgeDriver

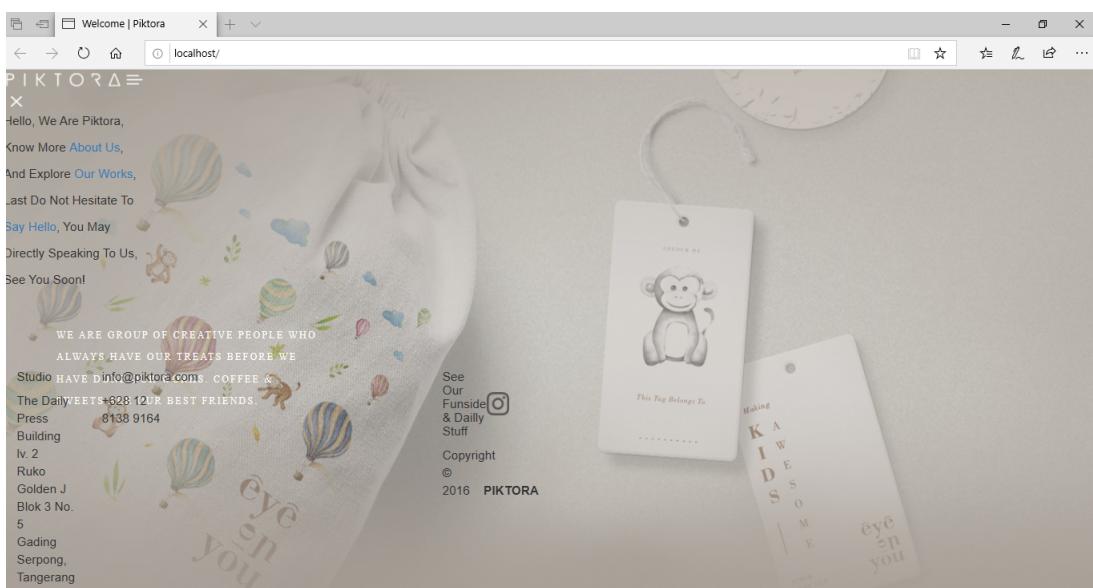
1376 Pengujian pada proyek Piktora dilakukan menggunakan EdgeDriver. Pada saat melakukan  
 1377 pengujian, dilakukan sedikit perubahan kode program pada kelas BrowserController baris  
 1378 ke-46 (lihat Lampiran A). *Object* bertipe WebDriver dinisialisasi menggunakan *object* bertipe  
 1379 EdgeDriver. Versi Microsoft Edge *browser* yang digunakan untuk pengujian adalah  
 1380 44.17763.1.0. Berikut ini adalah *Option* yang digunakan untuk menguji program:

```

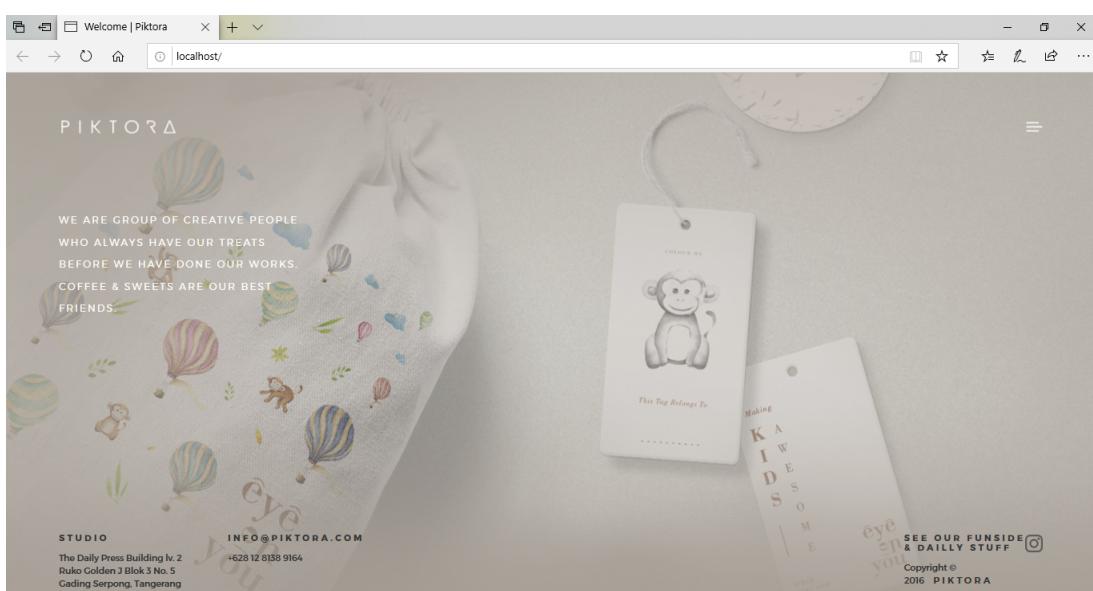
1381 • -project-path C:/xampp/htdocs/Piktora/.git
1382 • -capture-url http://localhost
1383 • -before-capture "php script_piktora.php"

```

Hasil pengujian ini tidak sesuai dengan harapan. Tampilan halaman *web* yang ditampilkan oleh Microsoft Edge *browser* saat dikontrol oleh EdgeDriver tidak rapih seperti yang diperlihatkan pada Gambar 5.10. Tidak diketahui apa yang menyebabkan halaman *web* tersebut menjadi tidak rapih. Sebagai perbandingan, Gambar 5.11 menunjukkan tampilan halaman *web* yang ditampilkan oleh Microsoft Edge *browser* saat tidak dikontrol oleh EdgeDriver. Meskipun tampilan dari halaman *web* tidak rapih, program tetap dapat membangkitkan animasi pada proyek Piktora.



Gambar 5.10: Tampilan halaman *web* pada *browser* saat dikontrol oleh EdgeDriver.



Gambar 5.11: Tampilan halaman *web* pada *browser* saat tidak dikontrol oleh EdgeDriver.

1391     **4. Pengujian Proyek Piktora dengan InternetExplorer**

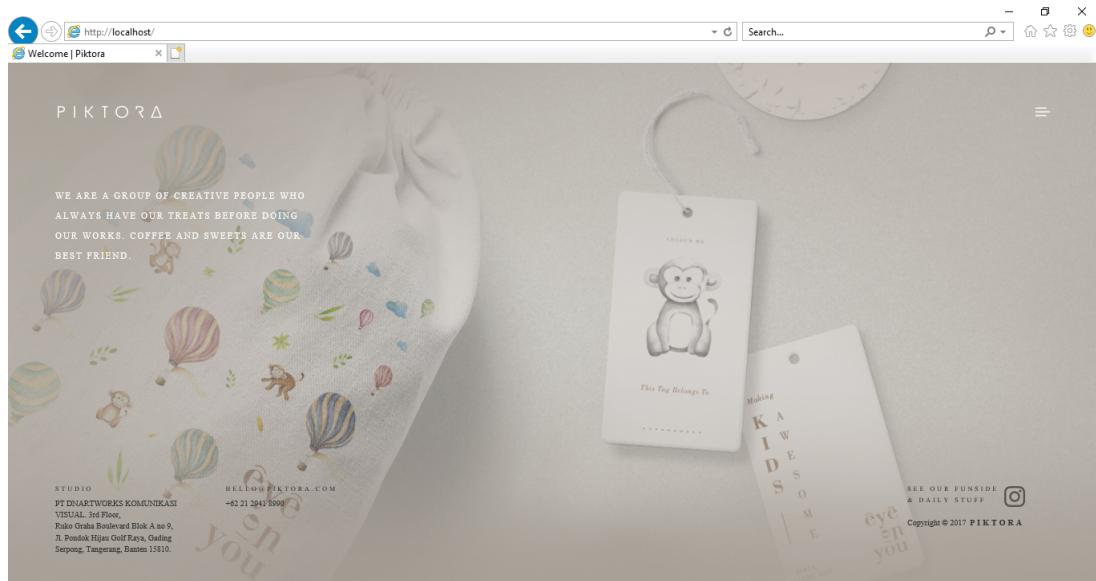
1392     Pengujian pada proyek Piktora dilakukan menggunakan InternetExplorerDriver. Pada saat  
1393     melakukan pengujian, dilakukan sedikit perubahan kode program pada kelas BrowserController  
1394     baris ke-46 (lihat Lampiran A). *Object* bertipe WebDriver dinisialisasi menggunakan *object*  
1395     bertipe InternetExplorerDriver. Versi Internet Explorer *browser* yang digunakan untuk  
1396     pengujian adalah 11.379.17763.0. Gambar 5.12 dan Gambar 5.13 menunjukkan tampilan pada  
1397     Internet Explorer *browser* saat sedang dikontrol oleh InternetExplorerDriver. Berikut ini  
1398     adalah *Option* yang digunakan untuk menguji program:

- 1399       • `-project-path C:/xampp/htdocs/Piktora/.git`  
1400       • `-capture-url http://localhost`  
1401       • `-before-capture "php script_piktora.php"`

1402     Program berhasil membangkitkan animasi *timelapse* pada proyek Piktora menggunakan  
1403     InternetExplorerDriver.



Gambar 5.12: Keadaan awal dari *browser* saat dikontrol oleh InternetExplorerDriver.



Gambar 5.13: Tampilan pada *browser* saat dikontrol oleh InternetExplorerDriver.

#### 5. Pengujian *Website* Netflix Open Source Software Center

Netflix Open Source Software Center merupakan proyek Open Source yang dimiliki oleh Netflix. Repotori *website* ini disimpan pada Github<sup>4</sup>. Repotori ini memiliki 393 *commit*. Lingkungan pengujian eksperimental ini sama dengan lingkungan implementasi yang terdapat pada subbab 5.1.1. Berikut ini adalah *Option* yang digunakan untuk menguji program:

- -project-path :/xampp/htdocs/netflix.github.com/.git
- -capture-url http://localhost
- -seconds-per-commit 0.1

Program berhasil membangkitkan animasi *timelapse* dari *website* Netflix Open Source Software Center. Tidak ditemukan masalah saat melakukan pengujian. Hasil dari pengujian berupa *file* hasil animasi bertipe GIF dengan ukuran 17.5 MB dan berdurasi 39 detik. Karena *file* hasil animasi bertipe GIF, maka hasilnya tidak bisa ditampilkan pada skripsi ini, tapi bisa dilihat pada link berikut<sup>5</sup>.

#### 6. Pengujian *Website* Proyek Bootstrap

Bootstrap<sup>6</sup> adalah kakas *open source* untuk membangun *website* yang dipakai bersama dengan HTML, CSS, dan JavaScript. Repotori<sup>7</sup> *website* ini disimpan pada Github. Pengujian ini dilakukan pada *branch* gh-pages, dimana di pada *branch* tersebut terdapat 8547 *commit*. Lingkungan pengujian eksperimental ini sama dengan lingkungan implementasi yang terdapat pada subbab 5.1.1. Berikut ini adalah *Option* yang digunakan untuk menguji program:

- -project-path C:/xampp/htdocs/bootstrap/.git
- -capture-url http://localhost

<sup>4</sup><https://github.com/Netflix/netflix.github.com>

<sup>5</sup><https://github.com/billyAdi/Skripsi/tree/master/Program/TimeLapseGenerator/hasil%20pengujian/Netflix>

<sup>6</sup><https://getbootstrap.com/>

<sup>7</sup><https://github.com/twbs/bootstrap>

- 1425     ● `-seconds-per-commit 0.05`
- 1426     ● `-branch gh-pages`

1427 Terdapat beberapa masalah saat melakukan pengujian *website* Bootstrap. Program suatu  
1428 ketika berhenti dan mengeluarkan pesan error: "short SHA1 685039d is ambiguous". Pesan  
1429 error ini muncul karena terdapat dua Git *object* yang mempunyai 7 digit ID yang sama,  
1430 sehingga tidak bisa melakukan *checkout* ke *commit* 685039d. Masalah berikutnya adalah  
1431 perbedaan letak *file* "index.html". Di beberapa *commit*, *file* "index.html" ini terletak di dalam  
1432 direktori "docs". Karena perbedaan letak *file* ini, halaman *web* menjadi tidak muncul, yang  
1433 muncul adalah struktur direktori dari *website* Boostrap. Masalah yang terakhir yaitu di  
1434 beberapa *commit* tidak terdapat *file* "index.html", sama seperti masalah sebelumnya hal ini  
1435 menyebabkan halaman *web* menjadi tidak muncul.

1436 Solusi untuk mengatasi masalah pertama yaitu dengan mengubah kode program di kelas VCS.  
1437 Awalnya program hanya menyimpan *commit* ID dengan panjang 7 digit. Setelah itu kode  
1438 program diubah supaya bisa menyimpan *commit* ID dengan panjang 10 digit.

1439 Solusi untuk mengatasi masalah kedua yaitu dengan menambahkan *Option -before-capture*  
1440 saat menjalankan program. Argumen dari Option tersebut berisi *terminal command* yang  
1441 menjalankan *script* PHP. *Script* PHP ini akan mengecek letak *file* "index.html" pada *folder*  
1442 utama dan "docs". *Script* kemudian akan mengecek *directory root* apache pada *file* "httpd.conf".  
1443 Jika *directory root* sudah mengarah ke *folder* tempat "index.html" berada, maka *script* tidak  
1444 akan mengubah isi *file* "httpd.conf". Jika *directory root* tidak mengarah ke *folder* tempat  
1445 "index.html" berada, maka *script* akan mengubah *directory root* pada *file* "httpd.conf" dan  
1446 melakukan *restart* pada apache.

1447 Untuk masalah ketiga, belum ditemukan solusinya. Jadi program akan tetap mengam-  
1448 bil *screenshot* meskipun tidak terdapat *file* "index.html". Setelah menambahkan *Option*  
1449 *-before-capture* dan mengubah kode program sehingga menyimpan *commit* ID dengan  
1450 panjang 10 digit, program berhasil membangkitkan animasi. Hasil dari pengujian berupa *file*  
1451 hasil animasi bertipe GIF dengan ukuran 281 MB dan berdurasi 7 menit 7 detik. Karena *file*  
1452 hasil animasi bertipe GIF, maka hasilnya tidak bisa ditampilkan pada skripsi ini.



1453

## BAB 6

1454

### KESIMPULAN DAN SARAN

#### 1455 6.1 Kesimpulan

1456 Dari hasil penelitian yang dilakukan, didapatkan kesimpulan-kesimpulan sebagai berikut:

- 1457 • Animasi *timelapse* pada pengembangan proyek perangkat lunak berbasis web dapat dibang-  
1458 kitkan dengan bantuan Git. Dengan Git, bisa didapatkan halaman-halaman *web* dari proyek  
1459 perangkat lunak berbasis web, kemudian *screenshot* dari halaman-halaman tersebut digabung  
1460 menjadi satu *file* bertipe GIF.
- 1461 • Aplikasi untuk membangkitkan *timelapse* pada pengembangan proyek perangkat lunak berbasis  
1462 web dapat diimplementasi dengan bantuan *library* JGit, Selenium WebDriver, dan Apache  
1463 Commons CLI.
- 1464 • Program sudah berjalan dengan baik dan dapat membangkitkan animasi *timelapse* dari proyek  
1465 pengembangan perangkat lunak berbasis *web*. Hal ini ditunjukkan dengan kesesuaian *output*  
1466 program dengan *output* yang diharapkan pada saat dilakukan pengujian fungsional.
- 1467 • Berdasarkan hasil pengujian menggunakan berbagai macam WebDriver, dapat diambil ke-  
1468 simpulan bahwa program dapat membangkitkan animasi dengan baik saat menggunakan  
1469 OperaDriver, FirefoxDriver, dan InternetExplorerDriver. Saat dilakukan pengujian menggu-  
1470 nakan EdgeDriver, tampilan *layout* dari halaman menjadi tidak rapih.

#### 1471 6.2 Saran

1472 Dari hasil penelitian yang dilakukan, berikut adalah beberapa saran untuk pengembangan:

- 1473 1. Saat ini *output* dari program berupa *file* bertipe GIF. *File* bertipe GIF ini memiliki dua  
1474 kekurangan, yaitu keterbatasan warna dan ukuran *file* yang besar. GIF memakai sistem palet  
1475 warna dan hanya terbatas pada 256 warna saja. Selain itu, *file* hasil animasi dari situs *web*  
1476 Bootstrap berukuran 281 MB. Karena adanya kedua kekurangan itu, sebaiknya memakai  
1477 format *file* yang lebih modern seperti MP4 atau format lainnya.
- 1478 2. Saat ini ukuran resolusi dari *file* hasil animasi bergantung pada resolusi layar yang digunakan.  
1479 Dalam pengembangan berikutnya, sebaiknya ditambahkan Command Line Option untuk  
1480 mengatur ukuran resolusi dari *file* hasil animasi.



## DAFTAR REFERENSI

- [1] Chacon, S. dan Straub, B. (2014) *Pro Git* The expert's voice. Apress.
- [2] Jgit | the eclipse foundation. <https://www.eclipse.org/jgit/>. [Online; diakses 2-September-2018].
- [3] Selenium webdriver. <https://www.seleniumhq.org/about/>. [Online; diakses 2-September-2018].
- [4] Jgit - parent 5.0.3.201809091024-r api. <http://download.eclipse.org/jgit/site/5.0.3.201809091024-r/apidocs/index.html>. [Online; diakses 17-September-2018].
- [5] Selenium documentation. <https://www.seleniumhq.org/docs/>. [Online; diakses 17-September-2018].
- [6] Generated documentation. <https://seleniumhq.github.io/selenium/docs/api/java/>. [Online; diakses 17-September-2018].
- [7] Commons - home. <https://commons.apache.org/proper/commons-cli/index.html>. [Online; diakses 11-Oktober-2018].
- [8] Apache commons cli 1.3.1 api. <https://commons.apache.org/proper/commons-cli/javadocs/api-release/index.html>. [Online; diakses 11-Oktober-2018].
- [9] Gource - a software version control visualization tool. <https://gource.io/>. [Online; diakses 29-Oktober-2018].



## LAMPIRAN A

### KODE PROGRAM

Listing A.1: BrowserController.java

```
1 package id.ac.unpar.timelapsegenerator;
2
3 import java.io.File;
4 import java.util.ArrayList;
5 import java.util.List;
6 import org.apache.commons.io.FileUtils;
7 import org.openqa.selenium.OutputType;
8 import org.openqa.selenium.TakesScreenshot;
9 import org.openqa.selenium.WebDriver;
10 import org.openqa.selenium.chrome.ChromeDriver;
11
12 /**
13 * Kelas ini digunakan untuk mengatur browser. Operasi-operasi yang dilakukan
14 * terhadap browser yaitu membuka browser, mengambil screenshot, membuat window
15 * browser menjadi maksimal, dan menutup browser.
16 *
17 * Hasil screenshot disimpan dalam bentuk List of File. Variabel driver
18 * merupakan suatu representasi dari browser. Kelas ini dapat memiliki maksimal
19 * empat browser.
20 *
21 * @author Billy Adiwijaya
22 */
23 public class BrowserController {
24
25     private final WebDriver[] drivers;
26     private final List<File> screenshotFiles;
27     private final int numberOfWorkers;
28     int ct=0;//debug
29
30     /**
31      * Constructor yang berfungsi untuk menginisialisasi variabel yang dimiliki
32      * oleh kelas ini.
33      *
34      * @param numberOfWorkers jumlah browser.
35      */
36     public BrowserController(int numberOfWorkers) {
37         this.drivers = new WebDriver[numberOfWorkers];
38         this.screenshotFiles = new ArrayList<>();
39         this.numberOfWorkers = numberOfWorkers;
40     }
41
42     /**
43      * Method ini berfungsi untuk membuka semua browser, kemudian mengatur
44      * ukuran window browser menjadi maksimal.
45      */
46     public void open() {
47         for (int i = 0; i < numberOfWorkers; i++) {
48             this.drivers[i] = new ChromeDriver();
49             this.drivers[i].manage().window().maximize();
50         }
51     }
52
53     /**
54      * Method ini berfungsi untuk mengembalikan jumlah browser yang dimiliki
55      * kelas ini.
56      *
57      * @return jumlah browser yang dimiliki oleh kelas ini.
58      */
59     public int getNumberOfWorkers() {
60         return numberOfWorkers;
61     }
62
63     /**
64      * Method ini berfungsi untuk berpindah halaman pada browser tertentu.
65      *
66      * @param browserIndex indeks browser yang akan diubah halamannya.
67      * @param url alamat URL untuk berpindah halaman.
68      */
69     public void changePage(int browserIndex, String url) {
70         this.drivers[browserIndex].get(url);
71     }
72
73     /**
74      * Method ini berfungsi untuk menutup semua browser.
75      */
76     public void quit() {
```

```

76     for (WebDriver driver : this.drivers) {
77         driver.quit();
78     }
79 }
80 /**
81 * Method ini berfungsi untuk mengambil screenshot pada browser tertentu dan
82 * menyimpannya ke atribut screenshotFiles.
83 *
84 * @param browserIndex indeks browser yang akan diambil screenshotnya.
85 */
86 public void takeScreenshot(int browserIndex) {
87     File scrFile
88         = ((TakesScreenshot) drivers[0]).getScreenshotAs(OutputType.FILE);
89     try {
90         FileUtils.copyFile(scrFile, new File("hasil_screenshot/z" + ct + ".png"));
91     } catch (Exception ex) {
92     }
93     ct++;
94     this.screenshotFiles.add(((TakesScreenshot) this.drivers[browserIndex]).getScreenshotAs(OutputType.FILE));
95 }
96 /**
97 * Method ini berfungsi untuk mengembalikan hasil screenshot.
98 *
99 * @return hasil screenshot berupa List of File.
100 */
101 public List<File> getScreenshotFiles() {
102     return this.screenshotFiles;
103 }
104 }
```

Listing A.2: CommandLineOptions.java

```

1 package id.ac.unpar.timelapsegenerator;
2
3 import java.util.Properties;
4 import org.apache.commons.cli.CommandLine;
5 import org.apache.commons.cli.CommandLineParser;
6 import org.apache.commons.cli.DefaultParser;
7 import org.apache.commons.cli.Option;
8 import org.apache.commons.cli.Options;
9 import org.apache.commons.cli.ParseException;
10
11 /**
12 * Kelas ini berfungsi untuk menyimpan semua Option yang terdapat pada program
13 * dan melakukan parsing argumen Command Line Option.
14 *
15 * @author Billy Adiwijaya
16 */
17 public class CommandLineOptions {
18
19     private final CommandLine commandLine;
20
21     /**
22      * Constructor dari kelas ini. Berfungsi untuk menentukan Option yang
23      * terdapat pada program dan melakukan parsing argumen Command Line.
24      *
25      * @param args merupakan argumen Command Line Option yang didapatkan dari
26      * kelas Main.
27      * @throws ParseException jika terjadi masalah saat melakukan parsing atau
28      * jumlah argumen capture-url lebih dari 4.
29      */
30     public CommandLineOptions(String[] args) throws ParseException {
31         CommandLineParser parser = new DefaultParser();
32
33         Options options = new Options();
34         options.addOption(Option.builder().required().hasArgs().longOpt("capture-url").argName("url").desc("link yang akan di
35             capture").build());
36         options.addOption(Option.builder().required().longOpt("project-path").argName("path").hasArg().desc("path proyek perangkat
37             lunak").build());
38         options.addOption(Option.builder().longOpt("seconds-per-commit").argName("seconds").hasArg().desc("durasi satu commit").
39             build());
40         options.addOption(Option.builder().longOpt("before-capture").argName("terminal command").hasArg().desc("terminal command
41             yang dijalankan sebelum melakukan screenshot").build());
42         options.addOption(Option.builder().longOpt("start-commit").argName("commit id").hasArg().desc("commit id awal untuk
43             memangkitkan animasi").build());
44         options.addOption(Option.builder().longOpt("stop-commit").argName("commit id").hasArg().desc("commit id akhir untuk
45             memangkitkan animasi").build());
46         options.addOption(Option.builder().longOpt("title").argName("title").hasArg().desc("judul proyek yang akan ditampilkan di
47             pojok kiri bawah").build());
48         options.addOption(Option.builder().longOpt("logo").argName("image path").hasArg().desc("logo yang akan ditampilkan di
49             pojok kanan bawah").build());
50         options.addOption(Option.builder().longOpt("branch").argName("branch").hasArg().desc("branch yang digunakan untuk
51             membangkitkan animasi").build());
52
53         this.commandLine = parser.parse(options, args);
54         if (this.commandLine.getOptionValues("capture-url").length > 4) {
55             throw new ParseException("Jumlah url yang akan dicapture maksimal 4");
56         }
57
58     /**
59      * Method ini berfungsi untuk mengembalikan Option yang sudah diparsing.
60      *
61      * @return Option yang sudah diparsing berupa objek dengan tipe Properties.
62     }
```

```

54    */
55    public Properties getParsedOptions() {
56        Properties properties = new Properties();
57        for (Option option : this.commandLine.getOptions()) {
58            if (option.getLongOpt().equals("capture-url")) {
59                String[] values = option.getValues();
60                String value = values[0];
61
62                for (int i = 1; i < values.length; i++) {
63                    value = value + ";" + values[i];
64                }
65                properties.setProperty(option.getLongOpt(), value);
66            } else {
67                properties.setProperty(option.getLongOpt(), option.getValue());
68            }
69        }
70        return properties;
71    }
72}

```

Listing A.3: Main.java

```

1 package id.ac.unpar.timelapsegenerator;
2
3 import java.io.File;
4 import java.net.HttpURLConnection;
5 import java.net.URL;
6 import java.util.Properties;
7 import javax.imageio.ImageIO;
8 import org.eclipse.jgit.api.errors.GitAPIException;
9
10 /**
11 * @author Billy Adiwijaya
12 */
13 public class Main {
14
15     public static void main(String[] args) {
16         CommandLineOptions commandLineOptions = null;
17         try {
18             commandLineOptions = new CommandLineOptions(args);
19         } catch (Exception e) {
20             System.out.println("Animasi timelapse gagal dibuat");
21             System.out.println(e.getMessage());
22             System.exit(0);
23         }
24         Properties properties = commandLineOptions.getParsedOptions();
25         int numberOfWorkers = properties.getProperty("capture-url").split(";").length;
26
27         VCS vcs = null;
28         try {
29             String branch="master";
30             if(properties.getProperty("branch")!=null){
31                 branch=properties.getProperty("branch");
32             }
33             vcs = new VCS(properties.getProperty("project-path"),branch);
34             if (properties.getProperty("seconds-per-commit") != null) {
35                 if (Double.parseDouble(properties.getProperty("seconds-per-commit")) <= 0) {
36                     throw new Exception("Seconds per commit harus lebih besar dari 0");
37                 }
38                 else if(Double.parseDouble(properties.getProperty("seconds-per-commit"))>655){
39                     throw new Exception("Seconds per commit harus kurang dari sama dengan 655");
40                 }
41             }
42
43             if (properties.getProperty("logo") != null) {
44                 File file = new File(properties.getProperty("logo"));
45                 if (!file.exists() || ImageIO.read(new File(properties.getProperty("logo"))) == null) {
46                     throw new Exception("Path gambar tidak valid");
47                 }
48             }
49
50             if (properties.getProperty("start-commit") != null) {
51                 if (properties.getProperty("start-commit").length() < 7 || properties.getProperty("start-commit").length() > 40) {
52                     throw new Exception("Panjang commit ID awal harus berada di antara 7-40 karakter");
53                 }
54                 if (vcs.getCommitIndex(properties.getProperty("start-commit")) == -1) {
55                     throw new Exception("Commit ID awal tidak ditemukan");
56                 }
57             }
58
59             if (properties.getProperty("stop-commit") != null) {
60                 if (properties.getProperty("stop-commit").length() < 7 || properties.getProperty("stop-commit").length() > 40) {
61                     throw new Exception("Panjang commit ID akhir harus berada di antara 7-40 karakter");
62                 }
63                 if (vcs.getCommitIndex(properties.getProperty("stop-commit")) == -1) {
64                     throw new Exception("Commit ID akhir tidak ditemukan");
65                 }
66             }
67
68             if (properties.getProperty("start-commit") != null && properties.getProperty("stop-commit") != null) {
69                 if (vcs.getCommitIndex(properties.getProperty("start-commit")) > vcs.getCommitIndex(properties.getProperty("stop-
70                     commit"))) {
71                     throw new Exception("Commit ID awal dan akhir terbalik");
72                 } else if (vcs.getCommitIndex(properties.getProperty("start-commit")) == vcs.getCommitIndex(properties.getProperty(
73                     "stop-commit"))) {
74                     throw new Exception("Commit ID awal dan akhir tidak boleh sama");
75                 }
76             }
77         }
78     }
79 }

```

```

75    }
76    } catch (NumberFormatException e) {
77        System.out.println("Animasi timelapse gagal dibuat");
78        System.out.println("Seconds per commit harus berupa bilangan riil atau bilangan bulat");
79        System.exit(0);
80    } catch (Exception e) {
81        System.out.println("Animasi timelapse gagal dibuat");
82        System.out.println(e.getMessage());
83        System.exit(0);
84    }
85
86    BrowserController browserController = new BrowserController(numberOfBrowsers);
87    TimeLapseGenerator timeLapseGenerator = new TimeLapseGenerator();
88    try {
89        timeLapseGenerator.generateTimelapse(properties, vcs, browserController);
90    } catch (Exception e) {
91        try {
92            vcs.checkoutMaster();
93        } catch (GitAPIException ex) {
94        } finally {
95            System.out.println("Animasi timelapse gagal dibuat");
96            System.out.println(e.getMessage());
97            System.exit(0);
98        }
99    }
100   System.out.println("Animasi timelapse berhasil dibuat");
101 }
102 }
```

Listing A.4: TimeLapseGenerator.java

```

1 package id.ac.unpar.timelapsegenerator;
2
3 import java.awt.Color;
4 import java.awt.Font;
5 import java.awt.FontMetrics;
6 import java.awt.Graphics2D;
7 import java.awt.image	BufferedImage;
8 import java.io.File;
9 import java.io.IOException;
10 import java.text.SimpleDateFormat;
11 import java.util.ArrayList;
12 import java.util.Date;
13 import java.util.List;
14 import java.util.Properties;
15 import java.util.concurrent.TimeUnit;
16 import javax.imageio.ImageIO;
17 import javax.imageio.stream.FileImageOutputStream;
18 import javax.imageio.stream.ImageOutputStream;
19 import org.eclipse.jgit.api.errors.GitAPIException;
20
21 /**
22  * Kelas ini digunakan untuk membangkitkan animasi timelapse.
23  *
24  * @author Billy Adiwijaya
25  */
26 public class TimeLapseGenerator {
27
28 /**
29  * Method ini berfungsi untuk membangkitkan animasi timelapse. Hasil dari
30  * animasi berupa File dengan tipe GIF.
31  *
32  * @param properties variabel yang menampung key dan value Option yang sudah
33  * diparsing.
34  * @param vcs variabel bertipe VCS yang digunakan untuk berinteraksi pada
35  * proyek perangkat lunak berbasis web yang terekam oleh Git.
36  * @param browserController variabel bertipe BrowserController untuk mengatur
37  * browser.
38  * @throws GitAPIException jika terjadi masalah saat melakukan operasi Git
39  * Checkout atau Git Reset.
40  * @throws IOException jika terjadi masalah menjalankan terminal command.
41  * @throws InterruptedException jika terjadi interupsi pada thread saat
42  * menjalankan terminal command.
43  */
44 public void generateTimelapse(Properties properties, VCS vcs, BrowserController browserController) throws GitAPIException,
45     IOException, InterruptedException {
46     int indexAwal = 0;
47     int indexAkhir = vcs.getNumberOfCommit() - 1;
48
49     if (properties.getProperty("start-commit") != null) {
50         indexAwal = vcs.getCommitIndex(properties.getProperty("start-commit"));
51     }
52
53     if (properties.getProperty("stop-commit") != null) {
54         indexAkhir = vcs.getCommitIndex(properties.getProperty("stop-commit"));
55     }
56
57     String captureURL[] = properties.getProperty("capture-url").split(";");
58     browserController.open();
59     for (int i = indexAwal; i <= indexAkhir; i++) {
60         vcs.checkoutCommit(i);
61
62         if (properties.getProperty("before-capture") != null) {
63             Process process;
64             process = Runtime.getRuntime().exec(properties.getProperty("before-capture"));
65             if (!process.waitFor(30, TimeUnit.SECONDS)) {
66                 throw new IOException("Terminal Command Tidak Valid");
67             }
68     }
```

```

67
68
69     for (int j = 0; j < captureURL.length; j++) {
70         browserController.changePage(j, captureURL[j]);
71         browserController.takeScreenshot(j);
72     }
73
74     vcs.hardReset();
75 }
76 browserController.quit();
77 vcs.checkoutMaster();
78
79 List<File> screenshotFiles = new ArrayList<>();
80 List<File> resultImages = new ArrayList<>();
81 screenshotFiles = browserController.getScreenshotFiles();
82
83 switch (browserController.getNumberofBrowser()) {
84     case 1:
85         resultImages = screenshotFiles;
86         break;
87     case 2:
88         for (int i = 0, j = 0; i < screenshotFiles.size() / browserController.getNumberofBrowser(); i++, j += 2) {
89             resultImages.add(screenshotFiles.get(i));
90             BufferedImage bufferedImage1, bufferedImage2, bufferedImageResult;
91             bufferedImage1 = ImageIO.read(screenshotFiles.get(j));
92             bufferedImage2 = ImageIO.read(screenshotFiles.get(j + 1));
93             bufferedImageResult = new BufferedImage(bufferedImage1.getWidth(), bufferedImage1.getHeight(), bufferedImage1.getType());
94
95             Graphics2D graphics = bufferedImageResult.createGraphics();
96             graphics.setBackground(Color.WHITE);
97             graphics.fillRect(0, 0, bufferedImageResult.getWidth(), bufferedImageResult.getHeight());
98
99             graphics.drawImage(bufferedImage1, bufferedImage1.getWidth() / 4, 0, bufferedImage1.getWidth() / 2,
100                                bufferedImage1.getHeight() / 2, null);
101            graphics.drawImage(bufferedImage2, bufferedImage2.getWidth() / 4, bufferedImage2.getHeight() / 2,
102                                bufferedImage2.getWidth() / 2, bufferedImage2.getHeight() / 2, null);
103            graphics.dispose();
104            ImageIO.write(bufferedImageResult, "png", resultImages.get(i));
105        }
106        break;
107    case 3:
108        for (int i = 0, j = 0; i < screenshotFiles.size() / browserController.getNumberofBrowser(); i++, j += 3) {
109            resultImages.add(screenshotFiles.get(i));
110            BufferedImage bufferedImage1, bufferedImage2, bufferedImage3, bufferedImageResult;
111            bufferedImage1 = ImageIO.read(screenshotFiles.get(j));
112            bufferedImage2 = ImageIO.read(screenshotFiles.get(j + 1));
113            bufferedImage3 = ImageIO.read(screenshotFiles.get(j + 2));
114            bufferedImageResult = new BufferedImage(bufferedImage1.getWidth(), bufferedImage1.getHeight(), bufferedImage1.getType());
115
116            Graphics2D graphics = bufferedImageResult.createGraphics();
117            graphics.setBackground(Color.WHITE);
118            graphics.fillRect(0, 0, bufferedImageResult.getWidth(), bufferedImageResult.getHeight());
119
120            graphics.drawImage(bufferedImage1, bufferedImage1.getWidth() / 4, 0, bufferedImage1.getWidth() / 2,
121                                bufferedImage1.getHeight() / 2, null);
122            graphics.drawImage(bufferedImage2, 0, bufferedImage2.getHeight() / 2, bufferedImage2.getWidth() / 2,
123                                bufferedImage2.getHeight() / 2, null);
124            graphics.drawImage(bufferedImage3, bufferedImage3.getWidth() / 2, bufferedImage3.getHeight() / 2,
125                                bufferedImage3.getWidth() / 2, bufferedImage3.getHeight() / 2, null);
126            graphics.dispose();
127            ImageIO.write(bufferedImageResult, "png", resultImages.get(i));
128        }
129        break;
130    case 4:
131        for (int i = 0, j = 0; i < screenshotFiles.size() / browserController.getNumberofBrowser(); i++, j += 4) {
132            resultImages.add(screenshotFiles.get(i));
133            BufferedImage bufferedImage1, bufferedImage2, bufferedImage3, bufferedImage4, bufferedImageResult;
134            bufferedImage1 = ImageIO.read(screenshotFiles.get(j));
135            bufferedImage2 = ImageIO.read(screenshotFiles.get(j + 1));
136            bufferedImage3 = ImageIO.read(screenshotFiles.get(j + 2));
137            bufferedImage4 = ImageIO.read(screenshotFiles.get(j + 3));
138
139            bufferedImageResult = new BufferedImage(bufferedImage1.getWidth(), bufferedImage1.getHeight(), bufferedImage1.getType());
140
141            Graphics2D graphics = bufferedImageResult.createGraphics();
142            graphics.setBackground(Color.WHITE);
143            graphics.fillRect(0, 0, bufferedImageResult.getWidth(), bufferedImageResult.getHeight());
144
145            graphics.drawImage(bufferedImage1, 0, 0, bufferedImage1.getWidth() / 2, bufferedImage1.getHeight() / 2, null);
146            graphics.drawImage(bufferedImage2, bufferedImage2.getWidth() / 2, 0, bufferedImage2.getWidth() / 2,
147                                bufferedImage2.getHeight() / 2, null);
148            graphics.drawImage(bufferedImage3, 0, bufferedImage3.getHeight() / 2, bufferedImage3.getWidth() / 2,
149                                bufferedImage3.getHeight() / 2, null);
150            graphics.drawImage(bufferedImage4, bufferedImage4.getWidth() / 2, bufferedImage4.getHeight() / 2,
151                                bufferedImage4.getWidth() / 2, bufferedImage4.getHeight() / 2, null);
152            graphics.dispose();
153            ImageIO.write(bufferedImageResult, "png", resultImages.get(i));
154        }
155        break;
156    default:
157        break;
158    }
159
160    for (File resultImage : resultImages) {
161        if (properties.getProperty("title") == null && properties.getProperty("logo") == null) {
162            break;
163        } else {
164

```

```

155     if (properties.getProperty("title") != null) {
156         BufferedImage bufferedImage = ImageIO.read(resultImage);
157         Graphics2D graphic = (Graphics2D) bufferedImage.getGraphics();
158         graphic.setFont(new Font("Times New Roman", Font.BOLD, 18));
159         FontMetrics fontMetrics = graphic.getFontMetrics(graphic.getFont());
160         graphic.setColor(Color.black);
161         graphic.drawString(properties.getProperty("title"), 5, bufferedImage.getHeight() - fontMetrics.getDescent() - 5);
162         graphic.dispose();
163         ImageIO.write(bufferedImage, "png", resultImage);
164     }
165
166     if (properties.getProperty("logo") != null) {
167         BufferedImage logo = ImageIO.read(new File(properties.getProperty("logo")));
168         BufferedImage bufferedImage = ImageIO.read(resultImage);
169         Graphics2D graphic = (Graphics2D) bufferedImage.getGraphics();
170         graphic.drawImage(logo, bufferedImage.getWidth() - logo.getWidth() - 5, bufferedImage.getHeight() - logo.getHeight() - 5, null);
171         graphic.dispose();
172         ImageIO.write(bufferedImage, "png", resultImage);
173     }
174 }
175 String fileName = String.format("%s.gif", new SimpleDateFormat("yyyy-MM-dd.HH.mm.ss").format(new Date()));
176 try (ImageOutputStream output = new FileOutputStream(new File(fileName))) {
177     int frameDelay = 1000;
178     if (properties.getProperty("seconds-per-commit") != null) {
179         frameDelay = (int) (Double.parseDouble(properties.getProperty("seconds-per-commit")) * 1000);
180     } else if (frameDelay == 0) {
181         frameDelay = 1000;
182     }
183     GifSequenceWriter writer = new GifSequenceWriter(output, ImageIO.read(resultImages.get(0)).getType(), frameDelay,
184         false);
185
186     for (File resultImage : resultImages) {
187         BufferedImage bufferedImage = ImageIO.read(resultImage);
188         writer.writeToSequence(bufferedImage);
189     }
190     writer.close();
191 }
192 }
193 }
194 }

```

Listing A.5: VCS.java

```

1 package id.ac.unpar.timelapsegenerator;
2
3 import java.io.IOException;
4 import java.util.ArrayList;
5 import java.util.Collections;
6 import java.util.List;
7 import org.eclipse.jgit.api.Git;
8 import org.eclipse.jgit.api.ListBranchCommand;
9 import org.eclipse.jgit.api.ResetCommand.ResetType;
10 import org.eclipse.jgit.api.errors.GitAPIException;
11 import org.eclipse.jgit.internal.storage.file.FileRepository;
12 import org.eclipse.jgit.lib.Ref;
13 import org.eclipse.jgit.lib.Repository;
14 import org.eclipse.jgit.revwalk.RevCommit;
15
16 /**
17 * Kelas ini berfungsi untuk berinteraksi pada proyek perangkat lunak berbasis
18 * web yang terekam oleh Git.
19 *
20 * @author Billy Adiwijaya
21 */
22 public class VCS {
23
24     private final Git git;
25     private final List<String> commitIDs;
26
27     /**
28      * Constructor dari kelas ini. Berfungsi untuk menginisialisasi variabel git
29      * dan mendapatkan seluruh histori commit proyek perangkat lunak berbasis
30      * web pada branch tertentu. Dimana branch tersebut diambil dari parameter
31      * constructor.
32      *
33      * @param path merupakan path dari proyek perangkat lunak berbasis web.
34      * @param branch nama branch yang digunakan untuk membangkitkan animasi.
35      * @throws IOException jika path proyek tidak valid atau repositori tidak
36      * bisa diakses.
37      * @throws GitAPIException jika terjadi masalah saat melakukan operasi Git
38      * Log.
39      * @throws Exception jika branch tidak valid
40      */
41     public VCS(String path, String branch) throws IOException, GitAPIException, Exception {
42         Repository repository = new FileRepository(path);
43         if (repository.getRef("HEAD") == null) {
44             throw new IOException("Path proyek tidak valid");
45         }
46         this.git = new Git(repository);
47         List<Ref> refs = git.branchList().setListMode(ListBranchCommand.ListMode.ALL).call();
48         int refIdx = -1;
49         for (int i = 0; i < refs.size(); i++) {
50             if (refs.get(i).getName().contains(branch)) {
51                 refIdx = i;
52                 break;
53             }
54         }
55         if (refIdx == -1) {
56             throw new IOException("Branch tidak ditemukan");
57         }
58         Commit commit = git.getCommit(refIdx);
59         commitIDs = commit.getCommitList();
60     }
61
62     /**
63      * Mengembalikan daftar ID commit yang tersimpan dalam objek VCS.
64      *
65      * @return Daftar ID commit sebagai list string.
66      */
67     public List<String> getCommitIDs() {
68         return commitIDs;
69     }
70
71     /**
72      * Mengembalikan objek Git yang digunakan untuk berinteraksi dengan repositori.
73      *
74      * @return Objek Git.
75      */
76     public Git getGit() {
77         return git;
78     }
79
80     /**
81      * Mengembalikan nama branch yang digunakan untuk membangkitkan animasi.
82      *
83      * @return Nama branch.
84      */
85     public String getBranch() {
86         return branch;
87     }
88
89     /**
90      * Mengembalikan path proyek perangkat lunak berbasis web.
91      *
92      * @return Path proyek.
93      */
94     public String getPath() {
95         return path;
96     }
97 }

```

```

53|         }
54|
55|     if (refIdx == -1) {
56|         throw new Exception("Branch tidak valid");
57|     }
58|     git.checkout().setName(refs.get(refIdx).getName()).call();
59|     Iterable<RevCommit> commits = git.log().call();
60|     this.commitIDs = new ArrayList<>();
61|     for (RevCommit commit : commits) {
62|         this.commitIDs.add(commit.getName());
63|     }
64|     Collections.reverse(commitIDs);
65|
66| }
67|
68| /**
69| * Berfungsi untuk melakukan checkout ke commit tertentu.
70| *
71| * @param commitIndex indeks dari variabel commitIDs.
72| * @throws GitAPIException jika terjadi masalah saat melakukan operasi Git
73| * Checkout.
74| */
75| public void checkoutCommit(int commitIndex) throws GitAPIException {
76|     this.git.checkout().setName(this.commitIDs.get(commitIndex)).call();
77| }
78|
79| /**
80| * Berfungsi untuk melakukan checkout ke commit terakhir.
81| *
82| * @throws GitAPIException jika terjadi masalah saat melakukan operasi Git
83| * Checkout.
84| */
85| public void checkoutMaster() throws GitAPIException {
86|     this.git.checkout().setName("master").call();
87| }
88|
89| /**
90| * Berfungsi untuk melakukan operasi Git Reset. Operasi ini menghapus
91| * perubahan pada working tree dan staging area di commit tertentu.
92| *
93| * @throws GitAPIException jika terjadi masalah saat melakukan operasi Git
94| * Reset.
95| */
96| public void hardReset() throws GitAPIException {
97|     this.git.reset().setMode(ResetType.HARD).call();
98| }
99|
100| /**
101| * Berfungsi untuk mendapatkan jumlah commit.
102| *
103| * @return jumlah commit.
104| */
105| public int getNumberOfCommit() {
106|     return this.commitIDs.size();
107| }
108|
109| /**
110| * Berfungsi untuk mendapatkan index dari variabel commitID.
111| *
112| * @param commitID merupakan Commit ID yang akan dicari indeksnya.
113| * @return indeks dari variabel commitIDs.
114| */
115| public int getCommitIndex(String commitID) {
116|     int result = -1;
117|     for (int i = 0; i < this.commitIDs.size(); i++) {
118|         if (commitID.equals(this.commitIDs.get(i).substring(0, commitID.length()))) {
119|             result = i;
120|             break;
121|         }
122|     }
123|     return result;
124| }
125| }

```



## LAMPIRAN B

### LIBRARY UNTUK MEMBUAT FILE GIF

Listing B.1: GifSequenceWriter.java

```
1 package id.ac.unpar.timelapsegenerator;
2 //
3 //  GifSequenceWriter.java
4 //
5 //  Created by Elliot Kroo on 2009-04-25.
6 //
7 // This work is licensed under the Creative Commons Attribution 3.0 Unported
8 // License. To view a copy of this license, visit
9 // http://creativecommons.org/licenses/by/3.0/ or send a letter to Creative
10 // Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.
11 //
12 //
13 import javax.imageio.*;
14 import javax.imageio.metadata.*;
15 import javax.imageio.stream.*;
16 import java.awt.image.*;
17 import java.io.*;
18 import java.util.Iterator;
19
20 public class GifSequenceWriter {
21     protected ImageWriter gifWriter;
22     protected ImageWriteParam imageWriteParam;
23     protected IIOMetadata imageMetaData;
24
25     /**
26      * Creates a new GifSequenceWriter
27      *
28      * @param outputStream the ImageOutputStream to be written to
29      * @param imageType one of the imageTypes specified in BufferedImage
30      * @param timeBetweenFramesMS the time between frames in milliseconds
31      * @param loopContinuously wether the gif should loop repeatedly
32      * @throws IOException if no gif ImageWriters are found
33      *
34      * @author Elliot Kroo (elliot[at]kroo[dot]net)
35      */
36     public GifSequenceWriter(
37         ImageOutputStream outputStream,
38         int imageType,
39         int timeBetweenFramesMS,
40         boolean loopContinuously) throws IOException, IOException {
41         // my method to create a writer
42         gifWriter = getWriter();
43         imageWriteParam = gifWriter.getDefaultWriteParam();
44         ImageTypeSpecifier imageTypeSpecifier =
45             ImageTypeSpecifier.createFromBufferedImageType(imageType);
46
47         imageMetaData =
48             gifWriter.getDefaultImageMetadata(imageTypeSpecifier,
49             imageWriteParam);
50
51         String metaFormatName = imageMetaData.getNativeMetadataFormatName();
52
53         IIOMetadataNode root = (IIOMetadataNode)
54             imageMetaData.getAsTree(metaFormatName);
55
56         IIOMetadataNode graphicsControlExtensionNode = getNode(
57             root,
58             "GraphicControlExtension");
59
60         graphicsControlExtensionNode.setAttribute("disposalMethod", "none");
61         graphicsControlExtensionNode.setAttribute("userInputFlag", "FALSE");
62         graphicsControlExtensionNode.setAttribute(
63             "transparentColorFlag",
64             "FALSE");
65         graphicsControlExtensionNode.setAttribute(
66             "delayTime",
67             Integer.toString(timeBetweenFramesMS / 10));
68         graphicsControlExtensionNode.setAttribute(
69             "transparentColorIndex",
70             "0");
71
72         IIOMetadataNode commentsNode = getNode(root, "CommentExtensions");
73         commentsNode.setAttribute("CommentExtension", "Created by MAH");
74
75 }
```

```

76 IIOMetadataNode appExtensionsNode = getNode(
77   root,
78   "ApplicationExtensions");
79
80 IIOMetadataNode child = new IIOMetadataNode("ApplicationExtension");
81
82 child.setAttribute("applicationID", "NETSCAPE");
83 child.setAttribute("authenticationCode", "2.0");
84
85 int loop = loopContinuously ? 0 : 1;
86
87 child.setUserObject(new byte[]{ 0x1, (byte) (loop & 0xFF),
88   ((loop >> 8) & 0xFF)} );
89 appExtensionsNode.appendChild(child);
90
91 imageMetaData.setFromTree(metaFormatName, root);
92
93 gifWriter.setOutput(outputStream);
94
95 gifWriter.prepareWriteSequence(null);
96 }
97
98 public void writeToSequence(RenderedImage img) throws IOException {
99   gifWriter.writeToSequence(
100     new IIOMImage(
101       img,
102       null,
103       imageMetaData),
104     imageWriteParam);
105 }
106
107 /**
108 * Close this GifSequenceWriter object. This does not close the underlying
109 * stream, just finishes off the GIF.
110 */
111 public void close() throws IOException {
112   gifWriter.endWriteSequence();
113 }
114
115 /**
116 * Returns the first available GIF ImageWriter using
117 * ImageIO.getImageWritersBySuffix("gif").
118 *
119 * @return a GIF ImageWriter object
120 * @throws IOException if no GIF image writers are returned
121 */
122 private static ImageWriter getWriter() throws IOException {
123   Iterator<ImageWriter> iter = ImageIO.getImageWritersBySuffix("gif");
124   if(iter.hasNext()) {
125     throw new IOException("No GIF Image Writers Exist");
126   } else {
127     return iter.next();
128   }
129 }
130
131 /**
132 * Returns an existing child node, or creates and returns a new child node (if
133 * the requested node does not exist).
134 *
135 * @param rootNode the <tt>IIOMetadataNode</tt> to search for the child node.
136 * @param nodeName the name of the child node.
137 *
138 * @return the child node, if found or a new node created with the given name.
139 */
140 private static IIOMetadataNode getNode(
141   IIOMetadataNode rootNode,
142   String nodeName) {
143   int nNodes = rootNode.getLength();
144   for (int i = 0; i < nNodes; i++) {
145     if (rootNode.item(i).getNodeName().compareToIgnoreCase(nodeName)
146         == 0) {
147       return((IIOMetadataNode) rootNode.item(i));
148     }
149   }
150   IIOMetadataNode node = new IIOMetadataNode(nodeName);
151   rootNode.appendChild(node);
152   return(node);
153 }
154
155 /**
156 * public GifSequenceWriter(
157 *   BufferedOutputStream outputStream,
158 *   int imageType,
159 *   int timeBetweenFramesMS,
160 *   boolean loopContinuously) {
161 */
162
163
164 public static void main(String[] args) throws Exception {
165   if (args.length > 1) {
166     // grab the output image type from the first image in the sequence
167     BufferedImage firstImage = ImageIO.read(new File(args[0]));
168
169     // create a new BufferedOutputStream with the last argument
170     ImageOutputStream output =
171       new FileImageOutputStream(new File(args[args.length - 1]));
172
173     // create a gif sequence with the type of the first image, 1 second
174     // between frames, which loops continuously

```

```
175|     GifSequenceWriter writer =
176|         new GifSequenceWriter(output, firstImage.getType(), 1, false);
177|
178|     // write out the first image to our sequence...
179|     writer.writeToSequence(firstImage);
180|     for(int i=1; i<args.length-1; i++) {
181|         BufferedImage nextImage = ImageIO.read(new File(args[i]));
182|         writer.writeToSequence(nextImage);
183|     }
184|
185|     writer.close();
186|     output.close();
187| } else {
188|     System.out.println(
189|         "Usage: java GifSequenceWriter [list of gif files] [output file]");
190| }
191|
192| }
```



# LAMPIRAN C

## KODE PROGRAM PROYEK PIKTORA PADA COMMIT 89000BE (12 JANUARI 2018)

Listing C.1: auth.php

```
1 | <?php
2 | defined('BASEPATH') OR exit('No direct script access allowed');
3 |
4 | $config['domain'] = 'http://localhost';
5 | $config['google-clientid'] = '9911963967-e4apbt2ksb0492p59rtoc8qhcent6a4e.apps.googleusercontent.com';
6 | $config['google-clientsecret'] = 'tw-7N1uNuqdCPRkXRFvqz7xR';
7 | $config['google-redirecturi'] = $config['domain'] . '/admin/oauth2callback';
```

Listing C.2: config.php

```
1 | <?php
2 | defined('BASEPATH') OR exit('No direct script access allowed');
3 |
4 | /*
5 | -----
6 | Base Site URL
7 | -----
8 |
9 | URL to your CodeIgniter root. Typically this will be your base URL,
10 | WITH a trailing slash:
11 |
12 | http://example.com/
13 |
14 | WARNING: You MUST set this value!
15 |
16 | If it is not set, then CodeIgniter will try guess the protocol and path
17 | your installation, but due to security concerns the hostname will be set
18 | to $_SERVER['SERVER_ADDR'] if available, or localhost otherwise.
19 | The auto-detection mechanism exists only for convenience during
20 | development and MUST NOT be used in production!
21 |
22 | If you need to allow multiple domains, remember that this file is still
23 | a PHP script and you can easily do that on your own.
24 |
25 */
26 $config['base_url'] = 'http://piktora.localhost';
27
28 /*
29 | -----
30 | Index File
31 | -----
32 |
33 | Typically this will be your index.php file, unless you've renamed it to
34 | something else. If you are using mod_rewrite to remove the page set this
35 | variable so that it is blank.
36 |
37 */
38 $config['index_page'] = '';
39
40 /*
41 | -----
42 | URI PROTOCOL
43 | -----
44 |
45 | This item determines which server global should be used to retrieve the
46 | URI string. The default setting of 'REQUEST_URI' works for most servers.
47 | If your links do not seem to work, try one of the other delicious flavors:
48 |
49 | 'REQUEST_URI'    Uses $_SERVER['REQUEST_URI']
50 | 'QUERY_STRING'   Uses $_SERVER['QUERY_STRING']
51 | 'PATH_INFO'      Uses $_SERVER['PATH_INFO']
52 |
53 | WARNING: If you set this to 'PATH_INFO', URIs will always be URL-decoded!
54 */
55 $config['uri_protocol'] = 'REQUEST_URI';
56
57 /*
58 | -----
59 | URL suffix
60 | -----
```

```
61 | This option allows you to add a suffix to all URLs generated by CodeIgniter.  
62 | For more information please see the user guide:  
63 |  
64 | https://codeigniter.com/user_guide/general/urls.html  
65 */  
66 $config['url_suffix'] = '';  
67 /*  
68 |-----  
69 | Default Language  
70 |-----  
71 |  
72 | This determines which set of language files should be used. Make sure  
73 | there is an available translation if you intend to use something other  
74 | than english.  
75 |  
76 |  
77 |  
78 */  
79 $config['language'] = 'english';  
80 /*  
81 |-----  
82 | Default Character Set  
83 |-----  
84 |  
85 | This determines which character set is used by default in various methods  
86 | that require a character set to be provided.  
87 |  
88 | See http://php.net/htmlspecialchars for a list of supported charsets.  
89 |  
90 |  
91 */  
92 $config['charset'] = 'UTF-8';  
93 /*  
94 |-----  
95 | Enable/Disable System Hooks  
96 |-----  
97 |  
98 | If you would like to use the 'hooks' feature you must enable it by  
99 | setting this variable to TRUE (boolean). See the user guide for details.  
100 |  
101 |  
102 */  
103 $config['enable_hooks'] = FALSE;  
104 /*  
105 |-----  
106 | Class Extension Prefix  
107 |-----  
108 |  
109 |  
110 | This item allows you to set the filename/classname prefix when extending  
111 | native libraries. For more information please see the user guide:  
112 |  
113 | https://codeigniter.com/user_guide/general/core_classes.html  
114 | https://codeigniter.com/user_guide/general/creating_libraries.html  
115 |  
116 */  
117 $config['subclass_prefix'] = 'MY_';  
118 /*  
119 |-----  
120 | Composer auto-loading  
121 |-----  
122 |  
123 | Enabling this setting will tell CodeIgniter to look for a Composer  
124 | package auto-loader script in application/vendor/autoload.php.  
125 |  
126 | $config['composer_autoload'] = TRUE;  
127 |  
128 | Or if you have your vendor/ directory located somewhere else, you  
129 | can opt to set a specific path as well:  
130 |  
131 | $config['composer_autoload'] = '/path/to/vendor/autoload.php';  
132 |  
133 | For more information about Composer, please visit http://getcomposer.org/  
134 |  
135 | Note: This will NOT disable or override the CodeIgniter-specific  
136 | auto-loading (application/config/autoload.php)  
137 |  
138 */  
139 $config['composer_autoload'] = '../vendor/autoload.php';  
140 /*  
141 |-----  
142 | Allowed URL Characters  
143 |-----  
144 |  
145 |  
146 | This lets you specify which characters are permitted within your URLs.  
147 | When someone tries to submit a URL with disallowed characters they will  
148 | get a warning message.  
149 |  
150 | As a security measure you are STRONGLY encouraged to restrict URLs to  
151 | as few characters as possible. By default only these are allowed: a-z 0-9~%.:_-  
152 |  
153 | Leave blank to allow all characters -- but only if you are insane.  
154 |  
155 | The configured value is actually a regular expression character group  
156 | and it will be executed as: ! preg_match('/^<permitted_uri_chars>+$/i  
157 |  
158 | DO NOT CHANGE THIS UNLESS YOU FULLY UNDERSTAND THE REPERCUSSIONS!!  
159 |
```

```

160 */
161 $config['permitted_uri_chars'] = 'a-z 0-9%.:_-';
162 /*
163 |-----
164 | Enable Query Strings
165 |-----
166 |
167 | By default CodeIgniter uses search-engine friendly segment based URLs:
168 | example.com/who/what/where/
169 |
170 | By default CodeIgniter enables access to the $_GET array. If for some
171 | reason you would like to disable it, set 'allow_get_array' to FALSE.
172 |
173 | You can optionally enable standard query string based URLs:
174 | example.com?who=me&what=something&where=here
175 |
176 | Options are: TRUE or FALSE (boolean)
177 |
178 | The other items let you set the query string 'words' that will
179 | invoke your controllers and its functions:
180 | example.com/index.php?c=controller&m=function
181 |
182 | Please note that some of the helpers won't work as expected when
183 | this feature is enabled, since CodeIgniter is designed primarily to
184 | use segment based URLs.
185 |
186 */
187 $config['allow_get_array'] = TRUE;
188 $config['enable_query_strings'] = FALSE;
189 $config['controller_trigger'] = 'c';
190 $config['function_trigger'] = 'm';
191 $config['directory_trigger'] = 'd';
192 |
193 /*
194 |-----
195 | Error Logging Threshold
196 |-----
197 |
198 | You can enable error logging by setting a threshold over zero. The
199 | threshold determines what gets logged. Threshold options are:
200 |
201 | 0 = Disables logging, Error logging TURNED OFF
202 | 1 = Error Messages (including PHP errors)
203 | 2 = Debug Messages
204 | 3 = Informational Messages
205 | 4 = All Messages
206 |
207 | You can also pass an array with threshold levels to show individual error types
208 |
209 | array(2) = Debug Messages, without Error Messages
210 |
211 | For a live site you'll usually only enable Errors (1) to be logged otherwise
212 | your log files will fill up very fast.
213 |
214 */
215 $config['log_threshold'] = 0;
216 |
217 /*
218 |-----
219 | Error Logging Directory Path
220 |-----
221 |
222 | Leave this BLANK unless you would like to set something other than the default
223 | application/logs/ directory. Use a full server path with trailing slash.
224 |
225 */
226 $config['log_path'] = '';
227 |
228 /*
229 |-----
230 | Log File Extension
231 |-----
232 |
233 | The default filename extension for log files. The default 'php' allows for
234 | protecting the log files via basic scripting, when they are to be stored
235 | under a publicly accessible directory.
236 |
237 | Note: Leaving it blank will default to 'php'.
238 |
239 */
240 $config['log_file_extension'] = '';
241 |
242 /*
243 |-----
244 | Log File Permissions
245 |-----
246 |
247 | The file system permissions to be applied on newly created log files.
248 |
249 | IMPORTANT: This MUST be an integer (no quotes) and you MUST use octal
250 |           integer notation (i.e. 0700, 0644, etc.)
251 |
252 */
253 $config['log_file_permissions'] = 0644;
254 |
255 /*
256 |-----
257 | Date Format for Logs
258 |-----

```

```
259 | Each item that is logged has an associated date. You can use PHP date  
260 | codes to set your own date formatting  
261 |  
262 | */  
263 $config['log_date_format'] = 'Y-m-d H:i:s';  
264 /*  
265 |-----  
266 | Error Views Directory Path  
267 |-----  
268 | Leave this BLANK unless you would like to set something other than the default  
269 | application/views/errors/ directory. Use a full server path with trailing slash.  
270 |  
271 |  
272 |-----  
273 |  
274 $config['error_views_path'] = '';  
275 /*  
276 |-----  
277 | Cache Directory Path  
278 |-----  
279 | Leave this BLANK unless you would like to set something other than the default  
280 | application/cache/ directory. Use a full server path with trailing slash.  
281 |  
282 |-----  
283 $config['cache_path'] = '';  
284 /*  
285 |-----  
286 | Cache Include Query String  
287 |-----  
288 | Whether to take the URL query string into consideration when generating  
289 | output cache files. Valid options are:  
290 |  
291 |-----  
292 | FALSE      = Disabled  
293 | TRUE       = Enabled, take all query parameters into account.  
294 | Please be aware that this may result in numerous cache  
295 | files generated for the same page over and over again.  
296 | array('q') = Enabled, but only take into account the specified list  
297 | of query parameters.  
298 |  
299 |-----  
300 $config['cache_query_string'] = FALSE;  
301 /*  
302 |-----  
303 | Encryption Key  
304 |-----  
305 | If you use the Encryption class, you must set an encryption key.  
306 | See the user guide for more info.  
307 |-----  
308 | https://codeigniter.com/user\_guide/libraries/encryption.html  
309 |-----  
310 |-----  
311 |-----  
312 |-----  
313 |-----  
314 |-----  
315 |-----  
316 |-----  
317 $config['encryption_key'] = '';  
318 /*  
319 |-----  
320 | Session Variables  
321 |-----  
322 |  
323 | 'sess_driver'  
324 |  
325 | The storage driver to use: files, database, redis, memcached  
326 |  
327 | 'sess_cookie_name'  
328 |  
329 | The session cookie name, must contain only [0-9a-z_-] characters  
330 |  
331 | 'sess_expiration'  
332 |  
333 | The number of SECONDS you want the session to last.  
334 | Setting to 0 (zero) means expire when the browser is closed.  
335 |  
336 | 'sess_save_path'  
337 |  
338 | The location to save sessions to, driver dependent.  
339 |  
340 | For the 'files' driver, it's a path to a writable directory.  
341 | WARNING: Only absolute paths are supported!  
342 |  
343 | For the 'database' driver, it's a table name.  
344 | Please read up the manual for the format with other session drivers.  
345 |  
346 |  
347 | IMPORTANT: You are REQUIRED to set a valid save path!  
348 |  
349 | 'sess_match_ip'  
350 |  
351 | Whether to match the user's IP address when reading the session data.  
352 |  
353 | WARNING: If you're using the database driver, don't forget to update  
354 | your session table's PRIMARY KEY when changing this setting.  
355 |  
356 | 'sess_time_to_update'  
357 |
```

```

358 |     How many seconds between CI regenerating the session ID.
359 |
360 |     'sess_regenerate_destroy'
361 |
362 |     Whether to destroy session data associated with the old session ID
363 |     when auto-regenerating the session ID. When set to FALSE, the data
364 |     will be later deleted by the garbage collector.
365 |
366 |     Other session cookie settings are shared with the rest of the application,
367 |     except for 'cookie_prefix' and 'cookie_httponly', which are ignored here.
368 |
369 */
370 $config['sess_driver'] = 'files';
371 $config['sess_cookie_name'] = 'ci_session';
372 $config['sess_expiration'] = 7200;
373 $config['sess_save_path'] = NULL;
374 $config['sess_match_ip'] = FALSE;
375 $config['sess_time_to_update'] = 300;
376 $config['sess_regenerate_destroy'] = FALSE;
377 /*
378 |-----
379 |-----|Cookie Related Variables|-----|
380 |-----|
382 |
383 |     'cookie_prefix' = Set a cookie name prefix if you need to avoid collisions
384 |     'cookie_domain' = Set to .your-domain.com for site-wide cookies
385 |     'cookie_path' = Typically will be a forward slash
386 |     'cookie_secure' = Cookie will only be set if a secure HTTPS connection exists.
387 |     'cookie_httponly' = Cookie will only be accessible via HTTP(S) (no javascript)
388 |
389 |     Note: These settings (with the exception of 'cookie_prefix' and
390 |           'cookie_httponly') will also affect sessions.
391 |
392 */
393 $config['cookie_prefix'] = '';
394 $config['cookie_domain'] = '';
395 $config['cookie_path'] = '/';
396 $config['cookie_secure'] = FALSE;
397 $config['cookie_httponly'] = FALSE;
398 /*
399 |-----
400 |-----|Standardize newlines|-----|
401 |-----|
403 |
404 |     Determines whether to standardize newline characters in input data,
405 |     meaning to replace \r\n, \r, \n occurrences with the PHP_EOL value.
406 |
407 |     This is particularly useful for portability between UNIX-based OSes,
408 |     (usually \n) and Windows (\r\n).
409 |
410 */
411 $config['standardize_newlines'] = FALSE;
412 /*
413 |-----
414 |-----|Global XSS Filtering|-----|
415 |-----|
417 |
418 |     Determines whether the XSS filter is always active when GET, POST or
419 |     COOKIE data is encountered
420 |
421 |     WARNING: This feature is DEPRECATED and currently available only
422 |             for backwards compatibility purposes!
423 |
424 */
425 $config['global_xss_filtering'] = FALSE;
426 /*
427 |-----
428 |-----|Cross Site Request Forgery|-----|
429 |-----|
431 |
432 |     Enables a CSRF cookie token to be set. When set to TRUE, token will be
433 |     checked on a submitted form. If you are accepting user data, it is strongly
434 |     recommended CSRF protection be enabled.
435 |
436 |     'csrf_token_name' = The token name
437 |     'csrf_cookie_name' = The cookie name
438 |     'csrf_expire' = The number in seconds the token should expire.
439 |     'csrf_regenerate' = Regenerate token on every submission
440 |     'csrf_exclude_uris' = Array of URIs which ignore CSRF checks
441 */
442 $config['csrf_protection'] = FALSE;
443 $config['csrf_token_name'] = 'csrf_test_name';
444 $config['csrf_cookie_name'] = 'csrf_cookie_name';
445 $config['csrf_expire'] = 7200;
446 $config['csrf_regenerate'] = TRUE;
447 $config['csrf_exclude_uris'] = array();
448 /*
449 |-----
450 |-----|Output Compression|-----|
451 |-----|
452 |
453 |     Enables Gzip output compression for faster page loads. When enabled,
454 |     the output class will test whether your server supports Gzip.
455 |     Even if it does, however, not all browsers support compression
456 |     so enable only if you are reasonably sure your visitors can handle it.

```

```

457 | Only used if zlib.output_compression is turned off in your php.ini.
458 | Please do not use it together with httpd-level output compression.
459 |
460 |
461 | VERY IMPORTANT: If you are getting a blank page when compression is enabled it
462 | means you are prematurely outputting something to your browser. It could
463 | even be a line of whitespace at the end of one of your scripts. For
464 | compression to work, nothing can be sent before the output buffer is called
465 | by the output class. Do not 'echo' any values with compression enabled.
466 |
467 */
468 $config['compress_output'] = FALSE;
469 /*
470 |-----
471 |----- Master Time Reference -----
472 |----- 
473 |----- 
474 |
475 | Options are 'local' or any PHP supported timezone. This preference tells
476 | the system whether to use your server's local time as the master 'now'
477 | reference, or convert it to the configured one timezone. See the 'date
478 | helper' page of the user guide for information regarding date handling.
479 |
480 */
481 $config['time_reference'] = 'local';
482 /*
483 |-----
484 |----- Rewrite PHP Short Tags -----
485 |----- 
486 |----- 
487 |
488 | If your PHP installation does not have short tag support enabled CI
489 | can rewrite the tags on-the-fly, enabling you to utilize that syntax
490 | in your view files. Options are TRUE or FALSE (boolean)
491 |
492 | Note: You need to have eval() enabled for this to work.
493 |
494 */
495 $config['rewrite_short_tags'] = FALSE;
496 /*
497 |-----
498 |----- Reverse Proxy IPs -----
499 |----- 
500 |----- 
501 |
502 | If your server is behind a reverse proxy, you must whitelist the proxy
503 | IP addresses from which CodeIgniter should trust headers such as
504 | HTTP_X_FORWARDED_FOR and HTTP_CLIENT_IP in order to properly identify
505 | the visitor's IP address.
506 |
507 | You can use both an array or a comma-separated list of proxy addresses,
508 | as well as specifying whole subnets. Here are a few examples:
509 |
510 | Comma-separated: '10.0.1.200,192.168.5.0/24'
511 | Array: array('10.0.1.200', '192.168.5.0/24')
512 */
513 $config['proxy_ips'] = '';

```

Listing C.3: database.php

```

1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3 /*
4 |-----
5 |----- DATABASE CONNECTIVITY SETTINGS
6 |----- 
7 |
8 | This file will contain the settings needed to access your database.
9 |
10 | For complete instructions please consult the 'Database Connection'
11 | page of the User Guide.
12 |
13 |-----
14 |----- EXPLANATION OF VARIABLES
15 |----- 
16 |
17 | ['dsn'] The full DSN string describe a connection to the database.
18 | ['hostname'] The hostname of your database server.
19 | ['username'] The username used to connect to the database
20 | ['password'] The password used to connect to the database
21 | ['database'] The name of the database you want to connect to
22 | ['dbdriver'] The database driver. e.g.: mysqli.
23 |     Currently supported:
24 |         cubrid, ibase, mssql, mysql, mysqli, oci8,
25 |         odbc, pdo, postgres, sqlite, sqlite3, sqlsrv
26 | ['dbprefix'] You can add an optional prefix, which will be added
27 |             to the table name when using the Query Builder class
28 | ['pconnect'] TRUE/FALSE - Whether to use a persistent connection
29 | ['db_debug'] TRUE/FALSE - Whether database errors should be displayed.
30 | ['cache_on'] TRUE/FALSE - Enables/disables query caching
31 | ['cachedir'] The path to the folder where cache files should be stored
32 | ['char_set'] The character set used in communicating with the database
33 | ['dbcollat'] The character collation used in communicating with the database
34 |             NOTE: For MySQL and MySQLi databases, this setting is only used
35 |                   as a backup if your server is running PHP < 5.2.3 or MySQL < 5.0.7
36 |                   (and in table creation queries made with DB Forge).
37 |             There is an incompatibility in PHP with mysql_real_escape_string() which
38 |                   can make your site vulnerable to SQL injection if you are using a

```

```

39 |         multi-byte character set and are running versions lower than these.
40 |         Sites using Latin-1 or UTF-8 database character set and collation are unaffected.
41 |         ['swap_pre'] A default table prefix that should be swapped with the dbprefix
42 |         ['encrypt'] Whether or not to use an encrypted connection.
43 |
44 |         'mysql' (deprecated), 'sqlsrv' and 'pdo/sqlsrv' drivers accept TRUE/FALSE
45 |         'mysqli' and 'pdo/mysql' drivers accept an array with the following options:
46 |
47 |             'ssl_key' - Path to the private key file
48 |             'ssl_cert' - Path to the public key certificate file
49 |             'ssl_ca' - Path to the certificate authority file
50 |             'ssl_capath' - Path to a directory containing trusted CA certificates in PEM format
51 |             'ssl_cipher' - List of *allowed* ciphers to be used for the encryption, separated by colons (':')
52 |             'ssl_verify' - TRUE/FALSE; Whether verify the server certificate or not ('mysqli' only)
53 |
54 |             ['compress'] Whether or not to use client compression (MySQL only)
55 |             ['stricton'] TRUE/FALSE - forces 'Strict Mode' connections
56 |                         - good for ensuring strict SQL while developing
57 |             ['ssl_options'] Used to set various SSL options that can be used when making SSL connections.
58 |             ['failover'] array - A array with 0 or more data for connections if the main should fail.
59 |             ['save_queries'] TRUE/FALSE - Whether to "save" all executed queries.
60 |                         NOTE: Disabling this will also effectively disable both
61 |                         $this->db->last_query() and profiling of DB queries.
62 |                         When you run a query, with this setting set to TRUE (default),
63 |                         CodeIgniter will store the SQL statement for debugging purposes.
64 |                         However, this may cause high memory usage, especially if you run
65 |                         a lot of SQL queries ... disable this to avoid that problem.
66 |
67 |             The $active_group variable lets you choose which connection group to
68 |             make active. By default there is only one group (the 'default' group).
69 |
70 |             The $query_builder variables lets you determine whether or not to load
71 |             the query builder class.
72 */
73 $active_group = 'default';
74 $query_builder = TRUE;
75
76 $db['default'] = array(
77     'dsn' => '',
78     'hostname' => 'localhost',
79     'username' => 'root',
80     'password' => '',
81     'database' => 'piktora',
82     'dbdriver' => 'mysqli',
83     'dbprefix' => '',
84     'pconnect' => FALSE,
85     'db_debug' => (ENVIRONMENT !== 'production'),
86     'cache_on' => FALSE,
87     'cachedir' => '',
88     'char_set' => 'utf8',
89     'dbcollat' => 'utf8_general_ci',
90     'swap.pre' => '',
91     'encrypt' => FALSE,
92     'compress' => FALSE,
93     'stricton' => FALSE,
94     'failover' => array(),
95     'save_queries' => TRUE
96 );

```

Listing C.4: About.php

```

1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 class About extends CI_Controller {
5
6     public function index()
7     {
8         $this->load->view('about');
9     }
10 }

```

Listing C.5: Contact.php

```

1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 class Contact extends CI_Controller {
5
6     public function index()
7     {
8         $this->load->helper('string');
9         $this->load->config('piktora');
10        if($this->input->method()=='post'){
11            $expiration = time() - $this->config->item('captcha-settings')['expiration'];
12            $this->db->where('captcha_time < ', $expiration)->delete('captcha');
13            $this->db->where('word', strtolower($this->input->post('captcha')));
14            $this->db->where('ip_address', $this->input->ip_address());
15            $this->db->where('captcha_time >', $expiration);
16            $query = $this->db->get('captcha');
17            if ($query->num_rows() == 0)
18            {
19                $this->session->set_flashdata('warning', "Sorry. Wrong Captcha.");
20            }
21        }else{
22            $this->load->model("Email_model");
23            $name = $this->input->post('name');

```

```

24     $email = $this->input->post('email');
25     $subject = $this->input->post('subject');
26     $tempMessage = $this->input->post('message');
27     $message = "<p>Name: $name</p><p>Email: $email</p><p>Message: $tempMessage</p><p>*note: jika ingin membalas email,
28         balas dengan email tujuan yang tertera di pesan (tidak melalui reply)</p>";
29     $this->Email_model->send_email($email, $name, $subject, $message);
30     $this->session->set_flashdata('success', "Thanks for your response.");
31   }
32   redirect('/contact', 'refresh');
33 }
34 else{
35   $this->load->helper('captcha');
36   $cap = create_captcha($this->config->item('captcha-settings'));
37   $data = array(
38     'captcha_time' => $cap['time'],
39     'ip_address' => $this->input->ip_address(),
40     'word' => strtolower($cap['word'])
41   );
42   $query = $this->db->insert_string('captcha', $data);
43   $this->db->query($query);
44   $this->load->view('contact', array('captcha'=>$cap['image']));
45 }
46 }
47 }

```

Listing C.6: About.php

```

1 <?php
2
3 defined('BASEPATH') OR exit('No direct script access allowed');
4
5 class Migrate extends CI_Controller {
6
7 /**
8 * Jika dibuka di http://localhost/migrate akan mengupdate ke db terbaru.
9 * Jika ada data sendiri, mohon dibackup dulu.
10 */
11 public function index() {
12   $this->load->config('migration');
13   if ($this->config->item('migration_enabled') === TRUE) {
14     $this->load->library('migration');
15     set_time_limit(300);
16     $version = $this->migration->latest();
17     if ($version === FALSE) {
18       show_error($this->migration->error_string());
19     } else {
20       echo "Migrate success! DB Version: $version";
21     }
22   } else {
23     echo 'Migration not enabled in config!';
24   }
25 }
26 }
27

```

Listing C.7: Projects.php

```

1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 class Projects extends CI_Controller {
5
6   public function index()
7   {
8     $query = $this->db->get_where('projects', array(
9       'title IS NOT NULL' => null
10      ));
11     $this->load->view('projects', array('projects' => $query->result_array()));
12   }
13
14   public function detail($id)
15   {
16     $query = $this->db->get_where('projects', array(
17       'id' => $id
18      ));
19     $project = $query->row_array();
20     $photos = [];
21     for ($i = 1; file_exists("assets/img/projects/$id-$i.jpg"); $i++) {
22       $photos[] = "/assets/img/projects/$id-$i.jpg";
23     }
24     $project['photos'] = $photos;
25     $this->load->view('project_details', $project);
26   }
27

```

Listing C.8: Welcome.php

```

1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 class Welcome extends CI_Controller {
5
6   public function index()
7   {

```

```

8     $photos = [];
9     for ($i = 1; file_exists("assets/img/home/$i.jpg"); $i++) {
10        $photos[] = "/assets/img/home/$i.jpg";
11    }
12    $this->load->view('welcome', array('photos'=>$photos));
13 }
14 }
```

Listing C.9: Auth\_model.php

```

1 <?php
2
3 defined('BASEPATH') OR exit('No direct script access allowed');
4
5 class Auth_model extends CI_Model {
6
7     private $client;
8
9     public function __construct() {
10         parent::__construct();
11
12         $this->load->config('auth');
13         $this->client = new Google_Client();
14         $this->client->setHttpClient(new GuzzleHttpClient(array(
15             'base_uri' => Google_Client::API_BASE_PATH, // if doesn't work, try to uncomment this.
16             'verify' => getcwd() . '/..../certs/cacert.pem'
17         )));
18         $this->client->setClientId($this->config->item('google-clientid'));
19         $this->client->setClientSecret($this->config->item('google-clientsecret'));
20         $this->client->setRedirectUri($this->config->item('google-redirecturi'));
21         $this->client->addScope('https://www.googleapis.com/auth/userinfo.email');
22         $this->client->addScope('https://www.googleapis.com/auth/userinfo.profile');
23     }
24
25     /**
26      * Panggil method ini untuk mendapatkan hyperlink untuk melakukan OAuth.
27      * Biasanya digunakan saat akan menampilkan halaman "Login with Google".
28      * @return string URL untuk login
29      */
30     public function createAuthURL() {
31         return $this->client->createAuthUrl();
32     }
33
34     /**
35      * Panggil ini untuk menerima kode autentikasi hasil redirect dari Google,
36      * dan menentukan email dan role user yang berhasil login.
37      * @param string $oauthCode kode oauth, didapat dari parameter GET "code".
38      * @return boolean TRUE selalu.
39      * @throws Exception jika autentikasi gagal (ditolak). Exception message
40      * berisi penjelasan kenapa.
41      */
42     public function authenticateOAuthCode($oauthCode) {
43         $this->client->authenticate($oauthCode);
44         $oauth2Service = new Google_Service_Oauth2($this->client);
45         $userInfo = $oauth2Service->userinfo->get();
46         $email = $userInfo['email'];
47         $name = $userInfo['name'];
48
49         $this->load->model('Configuration_model');
50         $allowedUsers = $this->Configuration_model->getCSV('administrator_emails');
51         if (in_array($email, $allowedUsers)) {
52             $this->session->set_userdata('auth', array(
53                 'email' => $email,
54                 'name' => $name,
55             ));
56         } else {
57             throw new Exception("$email ($name) does not have access to admin page!");
58         }
59     }
60
61     public function getUserInfo() {
62         return $this->session->userdata('auth');
63     }
64
65     public function logout() {
66         $this->session->unset_userdata('auth');
67     }
68 }
69 }
```

Listing C.10: Configuration\_model.php

```

1 <?php
2
3 defined('BASEPATH') OR exit('No direct script access allowed');
4
5 class Configuration_model extends CI_Model {
6
7     public function set($id, $value) {
8         $this->db->set('value', $value);
9         $this->db->where('id', $id);
10        $this->db->update('configuration');
11    }
12
13    public function get($id) {
14        $query = $this->db->get_where('configuration', array('id' => $id));
15        $data = $query->row();
```

```

16     if ($data !== NULL) {
17         return $data->value;
18     }
19     return NULL;
20 }
21
22 public function getCSV($id) {
23     return preg_split('/ *, */', $this->get($id));
24 }
25 }

```

Listing C.11: Email\_model.php

```

1 <?php
2
3 defined('BASEPATH') OR exit('No direct script access allowed');
4
5 class Email_model extends CI_Model {
6
7     public function send_email($email, $name, $subject, $message, $debug = FALSE) {
8         if ($debug === TRUE) {
9             echo $message;
10            exit();
11        }
12        $this->load->config('secrets');
13        $config = $this->config->item('email-config');
14        $this->load->library('email', $config);
15        $this->email->set_newline("\r\n");
16        $this->email->set_crlf("\r\n");
17        $this->email->from("piktora@mailgun.dnartworks.com.au", "PIKTORA");
18        $this->email->to("hello@piktora.com");
19        $this->email->subject($subject);
20        $this->email->message($message);
21        if (!$this->email->send()) {
22            throw new Exception("Sorry. There was system error when trying to sent your email. Please try again later.");
23        }
24    }
25 }

```

Listing C.12: about.php

```

1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3 ?><!DOCTYPE html>
4 <html lang="en">
5 <head>
6     <?php $this->load->view('templates/head', array('title' => 'About', 'extra_css' => array('slick', 'slick-theme'))); ?>
7     <style>
8         .one-page-background{
9             background-color:white;
10            background-size:contain;
11            width:100%;
12        }
13
14         .content{
15             padding-left:15px;
16             padding-right:15px;
17         }
18
19         .slick-track > div{
20             outline:none;
21         }
22
23         h4{
24             text-transform: uppercase;
25             line-height: 35px;
26             font-weight: bold;
27             font-family: "Montserrat-Bold";
28         }
29
30         .content-next{
31             text-align:right;
32         }
33
34         .title{
35             text-transform: uppercase;
36             line-height: 25px;
37             font-size: 11px;
38             letter-spacing: 2px;
39             font-weight: bold;
40         }
41
42         .description{
43             font-size:10px;
44             line-height: 20px;
45             letter-spacing: 1px;
46         }
47
48         .content-navigation{
49             letter-spacing: 3px;
50             font-size:12px;
51             cursor: pointer;
52             margin-top: 50px;
53         }
54
55         .content-2 h5{

```

```

56     background-image: url("/assets/img/rectangle-services.png");
57     height:28px;
58     background-size: auto;
59     background-repeat: no-repeat;
60     padding-left:20px;
61     padding-top:7px;
62     font-size:12px;
63     letter-spacing: 3px;
64   }
65
66 .content-2 .row > div{
67   padding-top: 0px;
68   padding-left: 0px;
69   padding-right: 0px;
70 }
71
72 .content-2 .description{
73   max-width: 235px;
74 }
75
76 .slick-slide img{
77   display: inline;
78 }
79
80 .special-description{
81   letter-spacing: 3px;
82   font-weight: bold;
83 }
84
85 @media screen and (min-width: 768px) {
86   h4{
87     line-height: 50px;
88     font-size:28px;
89   }
90   .one-page-background{
91     background-image: url("/assets/img/about/img-background-about.jpg");
92     height:100vh;
93     background-size:contain;
94     background-repeat:no-repeat;
95     background-position:center bottom;
96   }
97
98   .content{
99     padding-left:20px;
100    padding-right:20px;
101  }
102
103  .content-1 div{
104    max-width: 50%;
105    margin-left:50%;
106    text-align: right;
107  }
108  .content-1 div:first-child{
109    margin-left:0%;
110    text-align:left;
111  }
112}
113
114 @media screen and (min-width: 992px) {
115   h4{
116     line-height: 50px;
117     font-size:22px;
118     margin-top: 0px;
119     margin-bottom: 0px;
120   }
121
122   .one-page-background{
123     background-size:cover;
124   }
125   .content{
126     padding-left:60px;
127     padding-right:60px;
128   }
129
130   .content-1 div{
131     width: 25%;
132   }
133
134   .content-1 div:first-child{
135     float:left;
136   }
137
138   .content-1 .content-next{
139     margin-left:75%;
140   }
141
142   .content-1 .special-description{
143     padding-top:18px;
144     width:30%;
145     margin-left:45%;
146   }
147 }
148 </style>
149 </head>
150 <body>
151   <div class="one-page-background">
152     <?php $this->load->view('templates/navbar'); ?>
153     <div class="content">
154       <div class="content-1">

```

```

155     <div>
156         <p class="title">We Produce And Deliver The Highest Possible User Experience Through Innovative Design</p>
157         <p class="description">PIKTORA specializes in creative and marketing material designs, often needed by
158             companies ranging from start-up level to the prominent areas. With solid team of graphic designers,
159             photographers, and web developers, we guarantee a creativity, originality, and identity to your brand.</p>
160     </div>
161     <div class="special-description">
162         <h4>We Do<br>Something Fun<br>To Make<br>Something Special</h4>
163     </div>
164     <div class="content-navigation content-next">
165         <p>WHAT WE DO </p>
166     </div>
167     <div class="content-2">
168         <div class="container-fluid">
169             <div class="row">
170                 <div class="col-xs-12 col-sm-6 col-md-3">
171                     <h5>BRANDING</h5>
172                     <p class="description">It all started with a discovery session where we will chat about your vision
173                         and expectation for your creative needs. Just mention and describe what you have in mind, or we
174                         can even brainstorm and gather ideas together; then we'll explore them from scratch. From
175                         there, leave the rest to us, we'll translate your creative ideas into a workable concept with a
176                         strong impression.</p>
177                 </div>
178                 <div class="col-xs-12 col-sm-6 col-md-3">
179                     <h5>GRAPHIC DESIGN</h5>
180                     <p class="description">Let us know what your target market is. We can design every single marketing
181                         material that you need. We'll act as your artwork consultant to narrow down your needs in order
182                         to achieve your goals. We always take a closer look in every detail, to make sure that the
183                         appearance of your brand looks consistent and meaningful.</p>
184                 </div>
185                 <div class="col-xs-12 col-sm-6 col-md-3">
186                     <h5>WEBSITE DESIGN</h5>
187                     <p class="description">We believe that a good design comes with a good user experience on how people
188                         access a brand in the virtual world. Piktora specializes in developing a brand, while also
189                         collaborating with the current trend. Remember, a website is not only a tool that tells your
190                         audience about what your brand does, but also creates an impressive display in today's modern
191                         industry, as well as an affective user experience from the very beginning.</p>
192                 </div>
193             </div>
194             <div class="content-navigation content-prev">
195                 <p> BACK TO OUR STORY</p>
196             </div>
197         </div>
198     </div>
199     <?php $this->load->view('templates/scripts', array('extra_js' => array('slick.min'))); ?>
200 <script>
201     $(document).ready(function(){
202         $(".content").slick({
203             swipe : false,
204             arrows: false
205         });
206         $(".content-next").click(function(event) {
207             $(".content").slick("slickNext");
208         });
209         $(".content-prev").click(function(event) {
210             $(".content").slick("slickPrev");
211         });
212     });
213 </script>
214 </body>
215 </html>

```

Listing C.13: contact.php

```

1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3 ?<!DOCTYPE html>
4 <html lang="en">
5 <head>
6     <?php $this->load->view('templates/head', array('title' => 'Contact', 'extra_css'=> array('font-awesome.min'))); ?>
7     <style>
8         body{
9             background-color:white;
10        }
11
12         .form-control{
13             color:#474342;
14             background-color:#eeeeae9;
15             border:0px;
16        }
17
18         .captcha-container > div {
19             padding-left:0px;
20             padding-right:0px;
21        }
22
23         .captcha-container div .form-control{
24             height:50px;
25             text-align:center;
26        }
27
28         .captcha-container div:first-child .form-control{
29             background-color:white;

```

```

30         border:1px solid;
31     }
32
33     .send-button{
34         background:none;
35         border:none;
36         box-shadow: none;
37         padding-left:0px;
38         text-transform: uppercase;
39         font-size:11px;
40         letter-spacing: 3px;
41     }
42
43     .send-button:hover{
44         background:none;
45     }
46
47     .captcha-container div:last-child{
48         margin-top:20px;
49     }
50
51     .container-title{
52         padding-right:20%;
53     }
54
55     .title{
56         margin-top:0px;
57         letter-spacing: 2px;
58         font-family: "Montserrat-Bold";
59     }
60
61     .description{
62         font-size:11px;
63         text-transform: uppercase;
64         letter-spacing: 2px;
65         line-height: 25px;
66     }
67
68     #captcha{
69         width:100%;
70         height:50px;
71     }
72
73     @media screen and (min-width: 768px) {
74         .container-contact{
75             padding-left:20px;
76             padding-right:20px;
77         }
78
79         .container-footer{
80             position:absolute;
81             bottom:0;
82         }
83
84         div .captcha-code{
85             padding-right: 10px;
86         }
87
88         div .captcha-input{
89             padding-left: 10px;
90         }
91     }
92
93     @media screen and (min-width: 992px) {
94         .container-contact{
95             padding-left:60px;
96             padding-right:60px;
97         }
98
99         .container-title{
100             padding-right:0px;
101         }
102
103         .captcha-container div:last-child{
104             margin-top:0px;
105         }
106     }
107
108     </style>
109 </head>
110 <body>
111     <?php $this->load->view('templates/navbar'); ?>
112     <?php $this->load->view('templates/flashmessage'); ?>
113     <div class="container-fluid container-contact">
114         <div class="row">
115             <div class="col-xs-12 col-sm-6 col-md-3 container-title">
116                 <h2 class="title">SAY HELLO ,</h2>
117                 <p class="description">DonâŽt hesitate to know us more.</p>
118             </div>
119             <div class="col-xs-12 col-sm-6 col-md-6">
120                 <?= form_open('/contact'); ?>
121                     <div class="form-group">
122                         <input type="text" name="name" class="form-control" placeholder="Your Name" required="required">
123                     </div>
124                     <div class="form-group">
125                         <input type="email" name="email" class="form-control" placeholder="Email Address" required="required">
126                     </div>
127                     <div class="form-group">
128                         <input type="text" name="subject" class="form-control" placeholder="Subject:" required="required">

```

```

129         </div>
130         <div class="form-group">
131             <textarea class="form-control" name="message" rows="7" placeholder="What do you want to say?" required="" required></textarea>
132         </div>
133         <div class="form-group captcha-container">
134             <div class="col-xs-6 col-md-3 captcha-code">
135                 <?=$captcha; ?>
136             </div>
137             <div class="col-xs-6 col-md-3 captcha-input">
138                 <input type="text" name="captcha" class="form-control" placeholder="Enter the code" required="required" required>
139             </div>
140             <div class="col-xs-6 col-md-3 col-md-offset-3">
141                 <button type="submit" class="btn btn-default send-button">Send Now </button>
142             </div>
143         </div>
144         <?= form_close(); ?>
145     </div>
146 </div>
147 </div>
148 </div>
149 <?php $this->load->view('templates/footer'); ?>
150 <?php $this->load->view('templates/scripts'); ?>
151 </body>
152 </html>

```

Listing C.14: projects.php

```

1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3 ?><!DOCTYPE html>
4 <html lang="en">
5 <head>
6     <?php $this->load->view('templates/head', array('title' => 'Projects'));>
7     <style>
8         .content{
9             color:black;
10            margin-bottom:30px;
11        }
12
13        .content h5{
14            line-height-bottom: 0px;
15            padding-top:10px;
16            line-height: 3px;
17            letter-spacing: 2px;
18            font-size:11px;
19            font-weight: bold;
20        }
21
22        .content p{
23            font-size:10px;
24            letter-spacing: 1px;
25            margin-bottom: 10px;
26        }
27
28        .content .description{
29            text-transform: uppercase;
30            font-size:9px;
31        }
32
33        @media screen and (min-width: 768px) {
34            .content{
35                min-height:300px;
36            }
37        }
38
39        @media screen and (min-width: 992px) {
40            .content{
41                padding-left:6.67%;
42            }
43        }
44    </style>
45 </head>
46 <body>
47     <?php $this->load->view('templates/navbar'); ?>
48     <div class="container container-content">
49         <div class="row">
50             <?php foreach ($projects as $project): ?>
51                 <a href="/projects/<?=$project['id'] ?>">
52                     <div class="col-xs-12 col-sm-6 col-md-4 content">
53                         ">
54                         <h3><?=$project['title'] ?></h3>
55                         <p><?=$project['products'] ?></p>
56                         <p class="description">- <?=$project['categories'] ?></p>
57                     </div>
58                 </a>
59             <?php endforeach; ?>
60         </div>
61     </div>
62     <?php $this->load->view('templates/footer'); ?>
63     <?php $this->load->view('templates/scripts'); ?>
64 </body>
65 </html>

```

Listing C.15: welcome.php

```

1 | <?php
2 | defined('BASEPATH') OR exit('No direct script access allowed');
3 | ?><!DOCTYPE html>
4 | <html lang="en">
5 | <head>
6 |   <?php $this->load->view('templates/head', array('title' => 'Welcome', 'extra_css' => array('slick', 'slick-theme'))); ?>
7 |   <style>
8 |     .banner{
9 |       position:absolute;
10 |       width:100%;
11 |       height:100vh;
12 |       top:0;
13 |       z-index:-100;
14 |     }
15 |
16 |     .banner div {
17 |       background-size:cover;
18 |       background-position:center;
19 |       width:100%;
20 |       height:100%;
21 |     }
22 |
23 |     .text-banner{
24 |       background:transparent;
25 |       padding-left:15px;
26 |       padding-right:25%;
27 |       padding-top:30px;
28 |       padding-bottom:10px;
29 |       color:white;
30 |       font-family: "Montserrat-Light";
31 |       letter-spacing: 2px;
32 |       text-transform: uppercase;
33 |       line-height: 20px;
34 |       font-size:8px;
35 |       color:#fffffa;
36 |     }
37 |
38 |     .container-footer{
39 |       position:absolute;
40 |       bottom:0;
41 |     }
42 |
43 |     @media screen and (min-width: 768px) {
44 |       .text-banner{
45 |         font-size:11px;
46 |         padding-right:15px;
47 |         width:50%;
48 |         background-color:transparent;
49 |         padding-left:20px;
50 |         line-height: 25px;
51 |       }
52 |     }
53 |
54 |     @media screen and (min-width: 992px) {
55 |       .text-banner{
56 |         width:30%;
57 |         padding-left:60px;
58 |       }
59 |     }
60 |   </style>
61 | </head>
62 | <body>
63 |   <?php $this->load->view('templates/navbar', array('menuColor' => 'white')); ?>
64 |   <div class="banner">
65 |     <?php foreach ($photos as $photo): ?>
66 |       <div style="background-image: url(<?= $photo?>);"></div>
67 |     <?php endforeach ?>
68 |   </div>
69 |   <div class="text-banner">
70 |     <p>We are a group of creative people who always have our treats before doing our works. Coffee and sweets are our best friend.</p>
71 |   </div>
72 |   <?php $this->load->view('templates/footer'); ?>
73 |   <?php $this->load->view('templates/scripts', array('extra_js' => array('slick.min'))); ?>
74 |   <script>
75 |     $(document).ready(function(){
76 |       $(".banner").slick({
77 |         autoplay: true,
78 |         autoplaySpeed: 2000,
79 |         arrows: false,
80 |         swipe:false
81 |       });
82 |     });
83 |   </script>
84 | </body>
85 | </html>

```

Listing C.16: project\_details.php

```

1 | <?php
2 | defined('BASEPATH') OR exit('No direct script access allowed');
3 | ?><!DOCTYPE html>
4 | <html lang="en">
5 | <head>
6 |   <?php $this->load->view('templates/head', array('title' => 'Project Details')); ?>
7 |   <style>
8 |     .content-text{

```

```
9     background-color:#a59c93;
10    color:#4f4745;
11    padding-top: 15px;
12    padding-left: 15px;
13    padding-bottom: 40px;
14  }
15
16 .content-text a{
17   color:#4f4745;
18   text-decoration: underline;
19   font-weight: bold;
20 }
21
22 .container-content-text{
23   padding-left: 0px;
24 }
25
26 .container-content-img img{
27   margin-bottom: 20px;
28   max-width: 80%;
29 }
30
31 .back-to-previous{
32   margin-top:20px;
33 }
34
35 .back-to-previous a{
36   color:#4f4745;
37   font-weight: bold;
38   text-transform: uppercase;
39   letter-spacing: 3px;
40   font-size:10px;
41 }
42
43 .content-text p{
44   font-size: 10px;
45   letter-spacing: 1px;
46 }
47
48 h5{
49   letter-spacing: 2px;
50   font-size: 11px;
51 }
52
53 .content-text > div{
54   margin-bottom: 30px;
55 }
56
57 @media screen and (min-width: 768px) {
58   .content-text{
59     padding-top: 30px;
60     padding-left: 20px;
61     padding-bottom: 250px;
62   }
63
64   .container-content-img img{
65     max-width: 100%;
66     padding-right:10px;
67   }
68
69   .back-to-previous{
70     padding-left:50px;
71   }
72
73   .container-content-img{
74     overflow-y: auto;
75   }
76
77   .container-content-text{
78     overflow-y: auto;
79   }
80 }
81
82 @media screen and (min-width: 992px) {
83   .container-content-text{
84     padding-right:80px;
85   }
86
87   .content-text{
88     padding-left: 60px;
89     padding-right:60px;
90     padding-bottom: 100px;
91   }
92
93   .container-content-img img{
94     padding-right:50px;
95   }
96
97 }
98 </style>
99 </head>
100 <body>
101   <?php $this->load->view('templates/navbar'); ?>
102   <div class="container-fluid container-all">
103     <div class="row">
104       <div class="col-xs-12 col-sm-6 col-sm-push-6 col-md-7 col-md-push-5 container-content-img">
105         <?php foreach($photos as $photo): ?>
106           ">
107         <?php endforeach; ?>
```

```

108         </div>
109     <div class="col-xs-12 col-sm-6 col-sm-pull-6 col-md-5 col-md-pull-7 container-content-text">
110         <div class="content-text">
111             <div>
112                 <h5><?= $title ?></h5>
113                 <p><?= $products ?></p>
114                 <p>- <?= $categories ?></p>
115             </div>
116             <div>
117                 <h5>About Client</h5>
118                 <p><?= $aboutClient ?></p>
119             </div>
120             <div>
121                 <h5>Our Concept</h5>
122                 <p><?= $ourConcept ?></p>
123             </div>
124             <?= $additionalInfo ?>
125         </div>
126         <div class="back-to-previous">
127             <p><a href="/projects"> See All Projects</a></p>
128         </div>
129     </div>
130 </div>
131 <?php $this->load->view('templates/scripts'); ?>
132 <script>
133     $(document).ready(function() {
134         function setContentHeight(){
135             var height = $(window).height() - $(".navigation").height() - 50;
136             $('.container-content-img').css("height", (height + 'px'));
137             $('.container-content-text').css("height", (height + 'px'));
138         }
139         if($window.width()>=768){
140             setContentHeight();
141         }
142         $(window).resize(function() {
143             if($window.width()>=768){
144                 setContentHeight();
145             }
146         });
147     });
148 });
149 </script>
150 </body>
151 </html>
152

```

Listing C.17: flashmessage.php

```

1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3 ?>
4 <?php if (isset($_SESSION['success'])): ?>
5     <div class="alert alert-success">
6         <i class="fa fa-check-circle-o"></i> <?= strip_tags($_SESSION['success']) ?>
7     </div>
8 <?php endif; ?>
9
10 <?php if (isset($_SESSION['info'])): ?>
11     <div class="alert alert-info">
12         <i class="fa fa-info-circle"></i> <?= strip_tags($_SESSION['info']) ?>
13     </div>
14 <?php endif; ?>
15
16 <?php if (isset($_SESSION['warning'])): ?>
17     <div class="alert alert-warning">
18         <i class="fa fa-exclamation-triangle"></i> <?= strip_tags($_SESSION['warning']) ?>
19     </div>
20 <?php endif; ?>
21
22 <?php if (isset($_SESSION['danger'])): ?>
23     <div class="alert alert-danger">
24         <i class="fa fa-exclamation-circle"></i> <?= strip_tags($_SESSION['danger']) ?>
25     </div>
26 <?php endif; ?>

```

Listing C.18: footer.php

```

1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3 ?><!--footer-->
4 <footer class="container-fluid container-footer">
5     <div class="row">
6         <div class="col-xs-12 col-sm-3 col-md-2">
7             <h5>Studio</h5>
8             <p>PT DNARTWORKS KOMUNIKASI VISUAL. 3rd Floor,<br>Ruko Graha Boulevard Blok A no 9,<br>Jl. Pondok Hijau Golf Raya,
9 Gading Serpong, Tangerang, Banten 15810.</p>
10        </div>
11        <div class="col-xs-12 col-sm-3 col-md-2">
12            <h5>hello@piktoria.com</h5>
13            <p>+62 21 2941 8990</p>
14        </div>
15        <div class="col-xs-12 col-sm-3 col-sm-offset-3 col-md-2 col-md-offset-6">
16            <table class="daily-stuff-container">
17                <tbody>
18                    <tr>

```

```

19         <td class="social-media"><a target="_blank" href="https://www.instagram.com/piktorastudio/"></a></td>
21     </tr>
22   </tbody>
23   <p>Copyright &copy; 2017&ampnbsp&nbsp;<b class="product-title">PIKTORA</b></p>
24 </div>
25 </div>
26 </footer>
27 <!--/footer-->
```

Listing C.19: head.php

```

1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3 ?><!--head-->
4 <meta charset="utf-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1">
7 <title>?= $title ?> | Piktora</title>
8 <link href="/assets/css/bootstrap.min.css" rel="stylesheet">
9 <link href="/assets/css/bootstrap-theme.min.css" rel="stylesheet">
10 <?php if (isset($extra_css)): ?>
11     <?php foreach ($extra_css as $css): ?>
12         <link href="/assets/css/<?= $css ?>.css" rel="stylesheet">
13     <?php endforeach; ?>
14 <?php endif; ?>
15 <link href="/assets/css/style.css" rel="stylesheet">
16 <!--[if lt IE 9]>
17     <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
18     <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
19 <![endif]-->
20 <!--/head-->
```

Listing C.20: navbar.php

```

1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3 if (!isset($menuColor)){
4     $menuColor='dark';
5 }
6 ?><!--navbar-->
7 <table class="navigation">
8     <tbody>
9         <tr>
10            <td class="navigation-left"><a href="/welcome"></
11                a></td>
12            <td class="navigation-right"></td>
13        </tbody>
14    </table>
15    <div class="menu-fullscreen">
16        <div class="menu-navigation-close">
17            <a id="close-menu" href="#"></a>
18        </div>
19        <div class="menu-navigation-content">
20            <p>Hello, we are Piktora.</p>
21            <p>Know more <a href="/about">about us</a>, </p>
22            <p>explore <a href="/projects">our works</a>, and</p>
23            <p>of course <a href="/contact">say hello</a> to</p>
24            <p>us! We'd love to meet up</p>
25            <p>over coffee (or beer)!</p>
26        </div>
27    </div>
28 <!--/navbar-->
```

Listing C.21: scripts.php

```

1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3 ?><!--scripts-->
4 <script src="/assets/js/jquery.min.js"></script>
5 <script src="/assets/js/bootstrap.min.js"></script>
6 <?php if (isset($extra_js)): ?>
7     <?php foreach ($extra_js as $js): ?>
8         <script src="/assets/js/<?= $js ?>.js"></script>
9     <?php endforeach; ?>
10 <?php endif; ?>
11 <script src="/assets/js/script.js"></script>
12 <!--/scripts-->
```