

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**

LEMBAR PENGESAHAN

PEMBANGKIT TIMELAPSE PENGEMBANGAN PROYEK PERANGKAT LUNAK BERBASIS WEB

Billy Adiwijaya

NPM: 2015730053

Bandung, «tanggal» «bulan» 2019

Menyetujui,

Pembimbing

Pascal Alfadian, M.Comp.

Ketua Tim Penguji

Anggota Tim Penguji

«penguji 1»

«penguji 2»

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

PEMBANGKIT TIMELAPSE PENGEMBANGAN PROYEK PERANGKAT LUNAK BERBASIS WEB

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal «tanggal» «bulan» 2019

Meterai Rp. 6000

Billy Adiwijaya
NPM: 2015730053

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»

«kepada siapa anda mempersembahkan skripsi ini...?»

KATA PENGANTAR

«Tuliskan kata pengantar dari anda di sini ...»

Bandung, «bulan» 2019

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan	1
1.4 Batasan Masalah	2
1.5 Metodologi	2
1.6 Sistematika Pembahasan	2
2 LANDASAN TEORI	5
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	11
2.2 JGit	12
2.2.1 Kelas Repository	12
2.2.2 Kelas FileRepository	12
2.2.3 Kelas Git	13
2.2.4 Kelas Checkout Command	14
2.2.5 Kelas Log Command	14
2.2.6 Kelas Reset Command	14
2.2.7 Kelas ResetCommand.ResetType	15
2.2.8 Kelas RevCommit	15
2.2.9 Kelas PersonIdent	15
2.3 Selenium WebDriver	16
2.3.1 Interface WebDriver	16
2.3.2 Interface WebDriver.Window	17
2.3.3 Kelas ChromeDriver	17
2.3.4 Kelas FirefoxDriver	17
2.3.5 Kelas OperaDriver	17
2.3.6 Kelas InternetExplorerDriver	18
2.3.7 Kelas EdgeDriver	18
2.3.8 Kelas HtmlUnitDriver	18
2.3.9 Interface OutputType	18
2.3.10 Interface TakesScreenshot	18

2.4	Apache Commons CLI	18
2.4.1	CommandLineParser	19
2.4.2	DefaultParser	19
2.4.3	CommandLine	19
2.4.4	Options	20
2.4.5	Option	20
2.4.6	Option.Builder	21
3	ANALISIS	23
3.1	Analisis Aplikasi Sejenis	23
3.2	Analisis Penggunaan JGit dan Selenium WebDriver	25
3.2.1	Analisis Penggunaan JGit	25
3.2.2	Analisis Penggunaan Selenium WebDriver	26
3.3	Analisis Fitur Aplikasi yang Dibangun	27
3.4	Prapengujian Website Piktora	30
3.4.1	Perbedaan Letak <i>File</i>	32
3.4.2	Permasalahan Konfigurasi <i>Database</i>	32
3.4.3	Permasalahan Migrasi <i>Database</i>	33
4	PERANCANGAN	35
4.1	Perancangan Kelas	35
4.2	Perancangan Antarmuka	37
5	IMPLEMENTASI DAN PENGUJIAN	41
5.1	Implementasi	41
5.1.1	Lingkungan Implementasi	41
5.1.2	Hasil Implementasi	41
5.2	Pengujian	43
	DAFTAR REFERENSI	45
	A KODE PROGRAM	47
	B LIBRARY UNTUK MEMBUAT FILE GIF	55
	C KODE PROGRAM PROYEK PIKTORA	59

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.	23
3.2	Histori <i>commit</i> direpresentasikan sebagai Directed Acyclic Graph.	26
3.3	<i>Browser</i> yang dikontrol oleh ChromeDriver.	27
3.4	<i>Use case diagram</i> perangkat lunak.	28
4.1	Diagram kelas.	35
4.2	Rancangan <i>output</i> jika terdapat satu halaman <i>web</i>	39
4.3	Rancangan <i>output</i> jika terdapat dua halaman <i>web</i>	39
4.4	Rancangan <i>output</i> jika terdapat tiga halaman <i>web</i>	40
4.5	Rancangan <i>output</i> jika terdapat empat halaman <i>web</i>	40
5.1	Isi dari <i>file</i> GIF jika terdapat satu halaman <i>web</i>	42
5.2	Isi dari <i>file</i> GIF jika terdapat dua halaman <i>web</i>	42
5.3	Isi dari <i>file</i> GIF jika terdapat tiga halaman <i>web</i>	42
5.4	Isi dari <i>file</i> GIF jika terdapat empat halaman <i>web</i>	43

DAFTAR TABEL

3.1	<i>Scenario case</i> membangkitkan animasi <i>timelapse</i>	29
-----	---	----

BAB 1

PENDAHULUAN

1.1 Latar Belakang

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

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

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

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

1.2 Rumusan Masalah

Rumusan masalah dari skripsi ini adalah sebagai berikut:

1. Bagaimana cara membangkitkan animasi *timelapse* pada pengembangan proyek perangkat lunak berbasis web?
2. Bagaimana cara mengimplementasikan aplikasi untuk membangkitkan *timelapse* pada pengembangan proyek perangkat lunak berbasis web?

1.3 Tujuan

Tujuan dari skripsi ini adalah sebagai berikut:

1. Mengetahui cara untuk membangkitkan animasi *timelapse* pada pengembangan proyek perangkat lunak berbasis web.
2. Mengetahui cara untuk mengimplementasikan aplikasi untuk membangkitkan *timelapse* pada pengembangan proyek perangkat lunak berbasis web.

1.4 Batasan Masalah

Batasan masalah pada skripsi ini adalah sebagai berikut:

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

1.5 Metodologi

Metodologi penelitian yang digunakan dalam skripsi ini adalah sebagai berikut:

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

1.6 Sistematika Pembahasan

Setiap bab dalam penelitian ini memiliki sistematika penulisan yang dijelaskan ke dalam poin-poin sebagai berikut:

1. Bab 1: Pendahuluan, yaitu membahas mengenai gambaran umum penelitian ini. Berisi tentang latar belakang, rumusan masalah, tujuan, batasan masalah, metode penelitian, dan sistematika penulisan.
2. Bab 2: Dasar Teori, yaitu membahas mengenai teori-teori yang mendukung berjalannya penelitian ini. Berisi tentang teori Git, JGit, Selenium WebDriver, dan Apache Commons CLI.
3. Bab 3: Analisis, yaitu membahas mengenai analisa masalah. Berisi tentang analisis aplikasi sejenis, analisis penggunaan JGit dan Selenium WebDriver untuk membangkitkan animasi *timelapse*, prapengujian, dan analisis fitur aplikasi yang dibangun.
4. Bab 4: Perancangan, yaitu membahas mengenai perancangan yang dilakukan sebelum melakukan tahapan implementasi. Berisi tentang perancangan perangkat lunak pembangkit *timelapse* proyek pengembangan perangkat lunak.

-
5. Bab 5: Implementasi dan Pengujian, yaitu membahas mengenai implementasi dan pengujian aplikasi yang telah dilakukan. Berisi tentang implementasi dan hasil pengujian aplikasi.
 6. Bab 6: Kesimpulan dan Saran, yaitu membahas hasil kesimpulan dari keseluruhan penelitian ini dan saran-saran yang dapat diberikan untuk penelitian berikutnya.

BAB 2

LANDASAN TEORI

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

2.1 Git

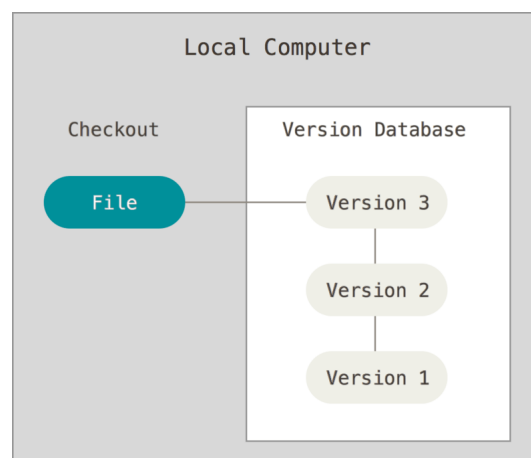
Git merupakan perangkat lunak *Version Control Systems*. Pada subbab ini, dijelaskan mengenai *Version Control Systems*, cara kerja Git, *Git checkout*, dan operasi-operasi dasar pada Git. Subbab ini mengacu pada [1].

2.1.1 Version Control Systems

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

Local Version Control Systems

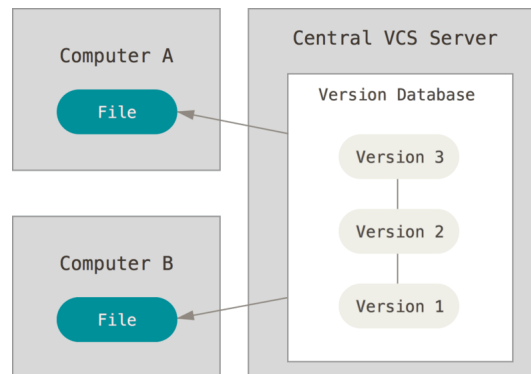
Metode *version-controlled* yang banyak digunakan orang adalah dengan cara menyalin sekumpulan *file* ke direktori lain. Namun cara tersebut rentan terhadap *error*. Misalnya, terdapat direktori A dan B, pengguna ingin mengubah *file* yang terdapat pada direktori B, tetapi pengguna lupa kalau dia sedang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



Gambar 2.1: Local version control[1].

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

Centralized Version Control Systems



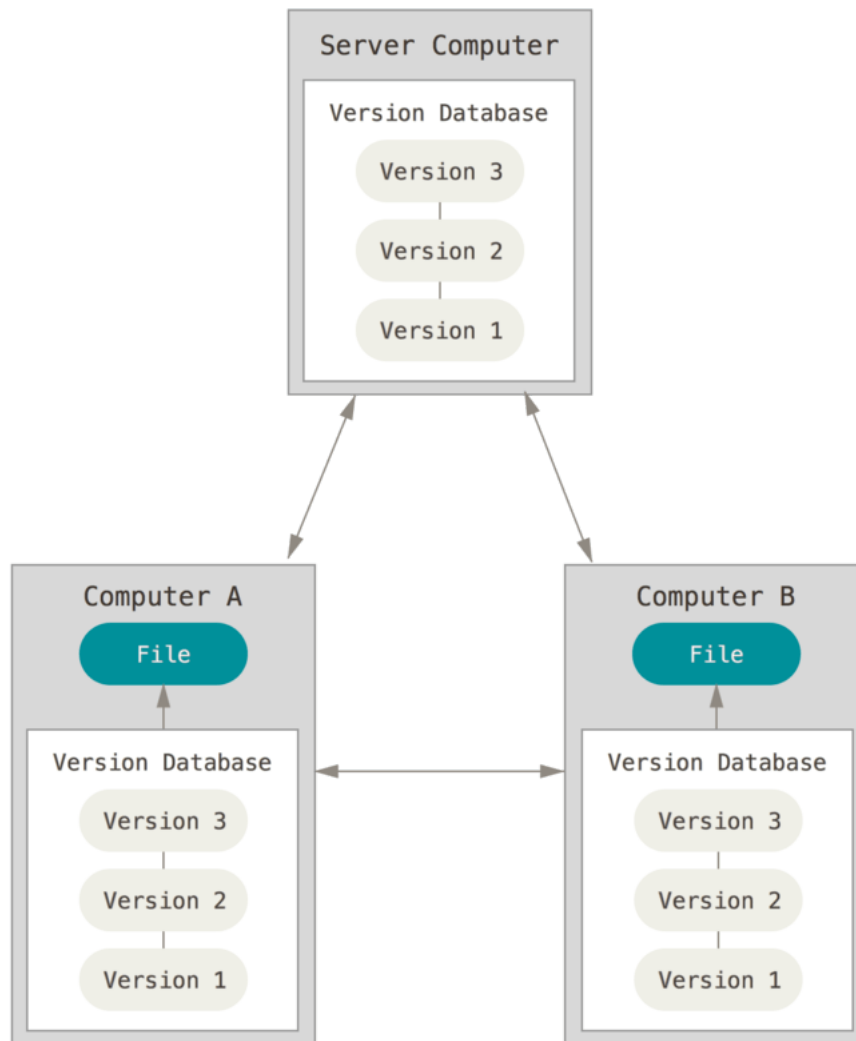
Gambar 2.2: Centralized version control[1].

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

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

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

Distributed Version Control Systems

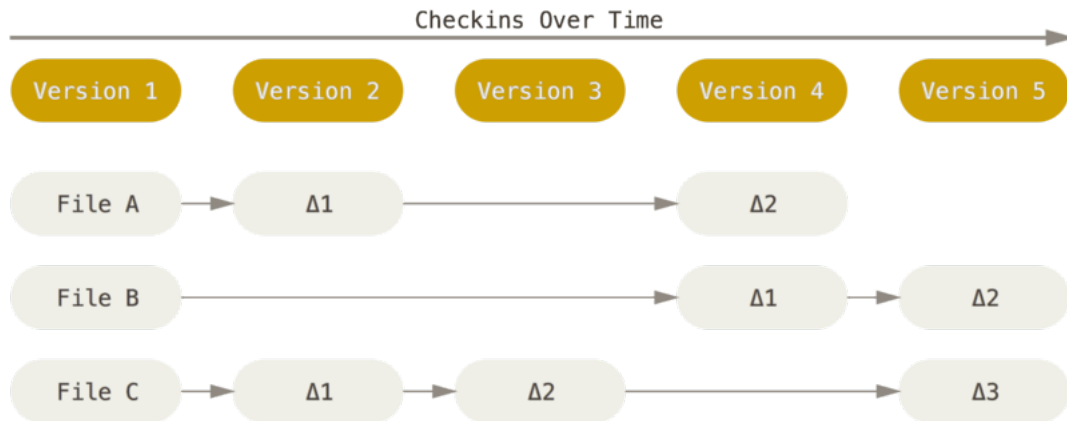


Gambar 2.3: Distributed version control[1].

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

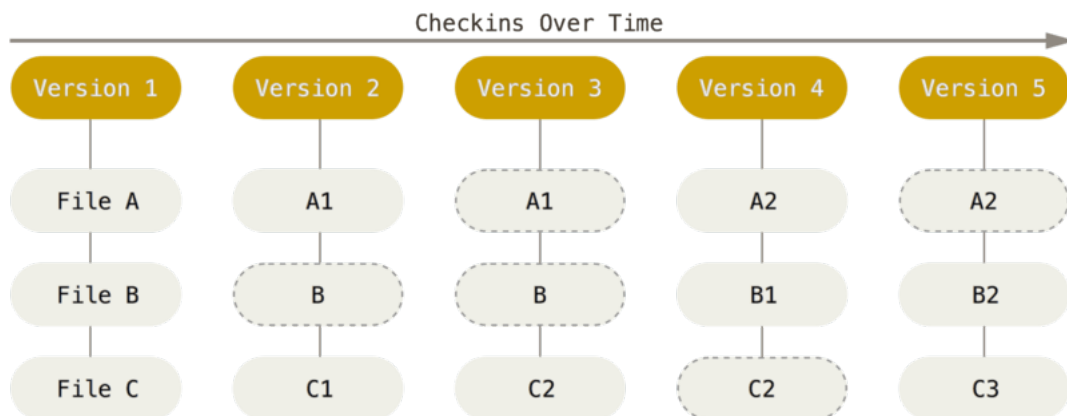
2.1.2 Cara Kerja Git

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



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

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



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

State pada Git

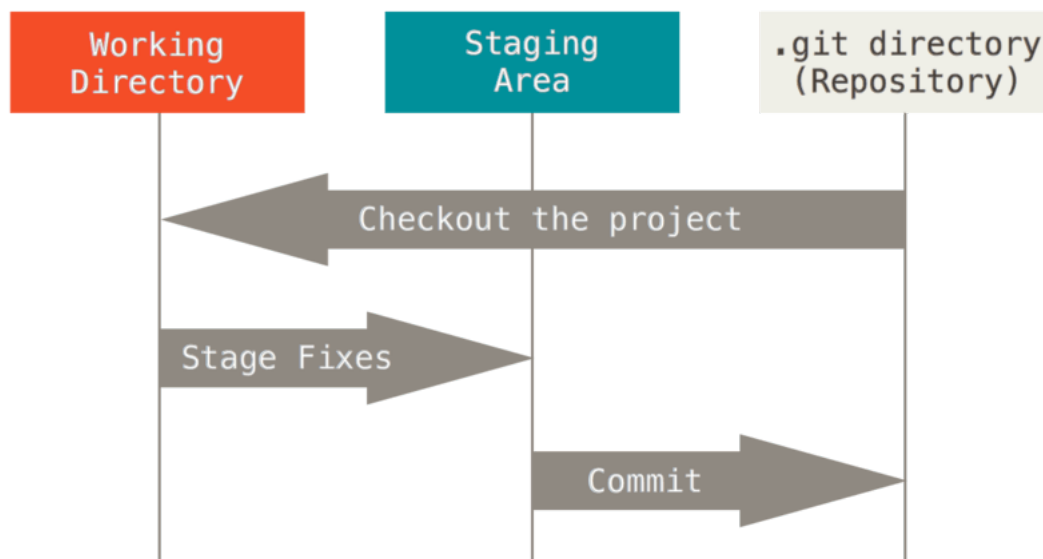
Terdapat tiga *state* pada Git yaitu *committed*, *modified*, and *staged*. *Committed* adalah *state* dimana data sudah disimpan di *local database*. *Modified* adalah *state* dimana terdapat perubahan pada *file*, namun *file* tersebut belum di *commit* ke *database*. *Staged* adalah *state* dimana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

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

Alur kerja dari Git adalah sebagai berikut:

1. Melakukan modifikasi pada *file*.
2. Menandai perubahan pada *file* dan memindahkannya ke *staging area*.

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[1].

Commit

Commit merupakan sebuah *snapshot* dari suatu *file* atau direktori. *Commit* menggambarkan *state* dari *working directory*. Gambar 2.5 menunjukkan terdapat tiga *file* pada *version 4*. Dimana terdapat *file* A1, B1, dan C1 pada *working directory*. *File* A1, B1, dan C2 merupakan *state file* A, B, dan C pada *version 4*.

Listing 2.1: Contoh histori commit dalam pengembangan perangkat lunak

```

1 | commit 89000be7ce7d16f006813cddefb4ec6d70d15ed6 (HEAD -> master, origin/master,
   | origin/HEAD)
2 | Author: Hizkia Steven <xvii.hs@gmail.com>
3 | Date:   Fri Jan 12 12:25:30 2018 +0700
4 |
5 |     Update new company address
6 |
7 | commit 6a085c1c37949e6308cfe06a117302e528388e54
8 | Author: Hizkia Steven <xvii.hs@gmail.com>
9 | Date:   Tue Dec 12 14:38:38 2017 +0700
10 |
11 |     Update company address
12 |
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
  
```

Seperti yang diperlihatkan pada Listing 2.1, setiap *commit* memiliki beberapa informasi. Baris pertama menunjukkan *commit* ID berupa *SHA-1 hash value*, dengan panjang empat puluh karakter

heksadesimal. *SHA-1 hash value* ini digunakan untuk pengecekan integritas pada *file*. Pada baris pertama, *master* menunjukkan *branch* yang sedang aktif, *master* juga merupakan *reference* ke *commit* terakhir. *HEAD* merupakan *reference* ke *master*. *Origin/master* dan *origin/HEAD* merupakan *master* dan *HEAD* pada *remote repository*. Baris kedua menunjukkan orang yang melakukan *commit* dan alamat emailnya. Baris ketiga menunjukkan waktu *commit*. Baris terakhir berisi deskripsi dari *commit* tersebut.

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]"
```

4. Branch

Operasi ini digunakan untuk menampilkan semua *branch* yang ada pada repositori Git, membuat *branch* baru, dan menghapus *branch*. Berikut adalah sintaks-sintaks untuk melakukan operasi *branch*:

```
$ git branch
```

```
$ git branch [branch-name]
```

```
$ git branch -d [branch-name]
```

```
$ git branch -D [branch-name]
```

5. Diff

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

```
$ git diff
```

```
$ git diff -staged
```

```
$ git diff [first-branch]...[second-branch]
```

6. Clone

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

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


7. Fetch

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

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

8. Merge

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

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

9. Pull

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

```
$ git pull
```

10. Push

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

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

11. Checkout

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

```
$ git checkout [commit ID]
```

```
$ git checkout [branch-name]
```

12. Log

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

```
$ git log
```

13. Reset

Operasi ini digunakan untuk memindahkan posisi *HEAD* ke *commit* tertentu, selain itu secara opsional melakukan *reset* pada *staging area* dan *working tree* berdasarkan tipe *reset*. Terdapat tiga tipe *reset* yaitu, *hard*, *soft*, dan *mixed*. Pada tipe *hard*, dilakukan *reset* pada *staging area* dan *working tree*, selain itu, posisi *HEAD* dipindah sehingga menunjuk ke *commit* tertentu. Pada tipe *soft*, hanya posisi *HEAD* saja yang berpindah. Pada tipe *mixed*, dilakukan *reset* pada *staging area* dan posisi *HEAD* dipindah sehingga menunjuk ke *commit* tertentu.

Berikut ini adalah sintaks untuk melakukan operasi *reset*:

```
$ git reset -hard [commit]
```

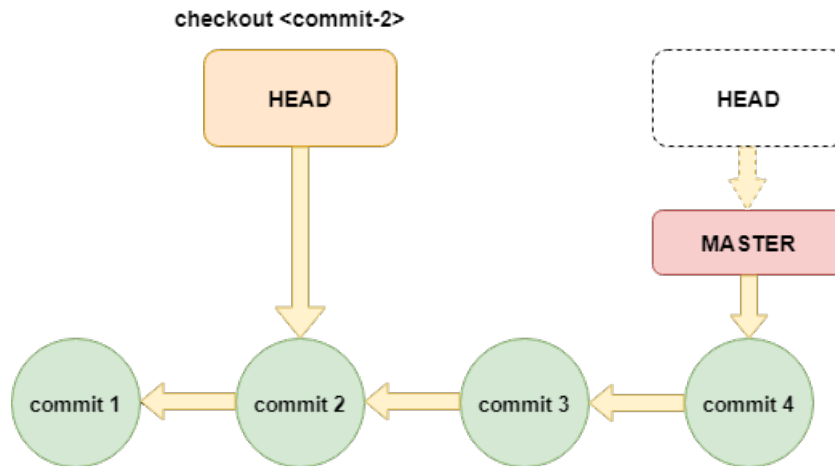
```
$ git reset -mixed [commit]
```

```
$ git reset -soft [commit]
```

2.1.4 Git Checkout

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

HEAD yang menunjuk langsung ke suatu *commit* disebut dengan *detached HEAD*. Perubahan yang terjadi pada *detached HEAD* tidak akan terekam oleh Git. Jika terdapat perubahan, kemudian dilakukan *checkout commit* atau *branch*, perubahan tersebut akan hilang. *HEAD* dapat dipindahkan ke posisi semula (menunjuk pada *branch master*) dengan menggunakan sintaks `$ git checkout 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*. *Ref* merupakan variabel yang menyimpan ID dari Git object.
Salah satu contoh dari Git Object adalah *commit*.

2.2.2 Kelas FileRepository

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

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

2.2.3 Kelas Git

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

- `public Git(Repository repo)`
Constructor ini membuat objek Git yang digunakan untuk berinteraksi dengan repositori Git. Parameter: objek *Repository* yang digunakan untuk berinteraksi. Parameter tidak boleh bernilai *null*.
- `public static InitCommand init()`
Method ini mengembalikan objek *command* untuk mengeksekusi operasi *init*.
Kembalian: objek *InitCommand* yang berfungsi untuk mengumpulkan parameter opsional dan akhirnya mengeksekusi operasi *init*.
- `public AddCommand add()`
Method ini mengembalikan objek *command* untuk mengeksekusi operasi *add*.
Kembalian: objek *AddCommand* yang berfungsi untuk mengumpulkan parameter opsional dan akhirnya mengeksekusi operasi *add*.
- `public LogCommand log()`
Method ini mengembalikan objek *command* untuk mengeksekusi operasi *log*.
Kembalian: objek *LogCommand* yang berfungsi untuk mengumpulkan parameter opsional dan akhirnya mengeksekusi operasi *log*.
- `public CheckoutCommand checkout()`
Method ini mengembalikan objek *command* untuk mengeksekusi operasi *checkout*.
Kembalian: objek *CheckoutCommand* yang berfungsi untuk mengumpulkan parameter opsional dan akhirnya mengeksekusi operasi *checkout*.
- `public CommitCommand commit()`
Method ini mengembalikan objek *command* untuk mengeksekusi operasi *commit*.
Kembalian: objek *CommitCommand* yang berfungsi untuk mengumpulkan parameter opsional dan akhirnya mengeksekusi operasi *commit*.
- `public FetchCommand fetch()`
Method ini mengembalikan objek *command* untuk mengeksekusi operasi *fetch*.
Kembalian: objek *FetchCommand* yang berfungsi untuk mengumpulkan parameter opsional dan akhirnya mengeksekusi operasi *fetch*.
- `public PushCommand push()`
Method ini mengembalikan objek *command* untuk mengeksekusi operasi *push*.
Kembalian: objek *PushCommand* yang berfungsi untuk mengumpulkan parameter opsional dan akhirnya mengeksekusi operasi *push*.
- `public DiffCommand diff()`
Method ini mengembalikan objek *command* untuk mengeksekusi operasi *diff*.
Kembalian: objek *DiffCommand* yang berfungsi untuk mengumpulkan parameter opsional dan akhirnya mengeksekusi operasi *diff*.
- `public static CloneCommand cloneRepository()`
Method ini mengembalikan objek *command* untuk mengeksekusi operasi *clone*.
Kembalian: objek *CloneCommand* yang berfungsi untuk mengumpulkan parameter opsional dan akhirnya mengeksekusi operasi *clone*.

- `public MergeCommand merge()`
Method ini mengembalikan objek *command* untuk mengeksekusi operasi *merge*.
Kembalian: objek *MergeCommand* yang berfungsi untuk mengumpulkan parameter opsional dan akhirnya mengeksekusi operasi *merge*.
- `public PullCommand pull()`
Method ini mengembalikan objek *command* untuk mengeksekusi operasi *pull*.
Kembalian: objek *PullCommand*.
- `public CreateBranchCommand branchCreate()`
Method ini mengembalikan objek *command* untuk membuat *branch* baru.
Kembalian: objek *CreateBranchCommand*.
- `public ListBranchCommand branchList()`
Method ini mengembalikan objek *command* untuk menampilkan daftar *branch*.
Kembalian: objek *ListBranchCommand*.
- `public DeleteBranchCommand branchDelete()`
Method ini mengembalikan objek *command* untuk menghapus *branch*.
Kembalian: objek *DeleteBranchCommand*.
- `public ResetCommand reset()`
Method ini mengembalikan objek *command* untuk mengeksekusi operasi *reset*.
Kembalian: objek *ResetCommand*.

2.2.4 Kelas Checkout Command

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

- `public CheckoutCommand setName(String name)`
Menentukan nama *branch* atau *commit* untuk melakukan *checkout*.
Parameter: nama dari *branch* atau *commit*.
Kembalian: *object* *CheckoutCommand*.
- `public Ref call()`
Berfungsi untuk menjalankan operasi *checkout*.

2.2.5 Kelas Log Command

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

- `public Iterable<RevCommit> call()`
Berfungsi untuk menjalankan operasi *log*.
Kembalian: histori *commit* pada *branch* yang sedang aktif.

2.2.6 Kelas Reset Command

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

- `public ResetCommand setMode(ResetCommand.ResetType mode)`
Menentukan tipe *reset*.
Parameter: tipe *reset*.
Kembalian: *object* *ResetCommand*.

- `public ResetCommand setRef(String ref)`
Mengatur nama *reference* tujuan dalam operasi *reset*. Secara default, *reference* yang digunakan adalah *HEAD*.
Parameter: nama dari *reference*.
Kembalian: *object* `ResetCommand`.
- `public Ref call()`
Berfungsi untuk menjalankan operasi *reset*.

2.2.7 Kelas `ResetCommand.ResetType`

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

- `public static final ResetCommand.ResetType SOFT`
Hanya mengubah posisi *HEAD*.
- `public static final ResetCommand.ResetType HARD`
Mengubah posisi *HEAD* ke *reference* tujuan. Selain itu, mengubah *staging area* dan *working directory* berdasarkan *reference* tersebut.
- `public static final ResetCommand.ResetType MIXED`
Mengubah posisi *HEAD* ke *reference* tujuan dan mengubah *staging area* berdasarkan *reference* tersebut.

2.2.8 Kelas `RevCommit`

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

- `public final String getFullMessage()`
Berfungsi untuk mendapatkan deskripsi dari suatu *commit*.
Kembalian: deskripsi dari suatu *commit* dalam bentuk `String`.
- `public final String getShortMessage()`
Berfungsi untuk mendapatkan deskripsi dari suatu *commit*. Jika panjang deskripsi *commit* lebih dari satu baris, maka hanya baris pertama yang dikembalikan.
Kembalian: baris pertama dari deskripsi *commit* dalam bentuk `String`.
- `public final String getName()`
Method ini mengembalikan *commit ID*.
Kembalian: *commit ID* berupa SHA-1 hash value dalam bentuk `String`.
- `public final PersonIdent getAuthorIdent()`
Berfungsi untuk mendapatkan informasi mengenai *author* yang melakukan *commit*.
Kembalian: objek *PersonIdent* yang memuat informasi tentang *author* (nama dan *email*) dan waktu dilakukannya *commit*.

2.2.9 Kelas `PersonIdent`

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

- `public String getName()`
Berfungsi untuk mengembalikan nama dari *author* yang melakukan *commit*.
Kembalian: nama dari *author*.

- `public String getEmailAddress()`
Berfungsi untuk mengembalikan alamat *email* dari *author* yang melakukan *commit*.
Kembalian: alamat *email* dari *author*.
- `public Date getWhen()`
Berfungsi mengembalikan waktu dilakukannya suatu *commit* oleh *author*.
Kembalian: waktu dilakukannya suatu *commit* berupa objek `Date`.

2.3 Selenium WebDriver

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

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

ChromeDriver merupakan implementasi *WebDriver* yang mengontrol Chrome *browser*. *FirefoxDriver* merupakan implementasi *WebDriver* yang mengontrol Firefox *browser*. *OperaDriver* merupakan implementasi *WebDriver* yang mengontrol Opera *browser*. *InternetExplorerDriver* merupakan implementasi *WebDriver* yang mengontrol Internet Explorer *browser*. *EdgeDriver* merupakan implementasi *WebDriver* yang mengontrol Microsoft Edge *browser*. *HtmlUnitDriver* merupakan implementasi *WebDriver* yang mengontrol *HTMLUnit browser* yang bersifat *headless*(tidak mempunyai GUI).

Pada skripsi ini *tools* Selenium yang digunakan hanya Selenium *WebDriver*. Bahasa pemrograman yang digunakan adalah java. *WebDriver* yang digunakan adalah *ChromeDriver*. Pemilihan *WebDriver* ini hanya berdasarkan preferensi penulis saja. Penulis tidak memilih *InternetExplorerDriver*, *EdgeDriver*, dan *HtmlUnitDriver* karena *InternetExplorerDriver* dan *EdgeDriver* hanya bisa berjalan pada sistem operasi Windows, selain itu *HtmlUnitDriver* tidak mempunyai GUI sehingga tidak bisa dilakukan pengambilan *screenshot*. Pada subbab berikut dijelaskan beberapa kelas dan *interface* dari *library* Selenium *WebDriver*. Subbab ini mengacu pada [6].

2.3.1 Interface WebDriver

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

- `void close()`
Berfungsi untuk menutup *window* pada *browser*, jika *window* yang sekarang merupakan satu-satunya *window* yang terbuka maka *browser* akan ditutup.
- `void quit()`
Berfungsi untuk menutup *browser* dan semua *window* yang sedang terbuka.
- `void get(String url)`
Berfungsi untuk memuat halaman *web* pada *window* yang sedang aktif. *Method* ini mengirim

HTTP GET Request untuk memuat halaman, dan *method* ini akan melakukan *blocking* sampai halaman *web* selesai dimuat.

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

- `String getTitle()`
Berfungsi untuk mengembalikan judul dari halaman *web* yang sedang aktif.
Kembalian: judul dari halaman *web*.
- `String getCurrentUrl()`
Berfungsi untuk mendapatkan URL yang sedang aktif di *browser*.
Kembalian: URL dari halaman *web* yang sedang dimuat di *browser*.
- `WebDriver.Options manage()`
Berfungsi untuk mendapatkan interface `WebDriver.Options`.
Kembalian: interface dengan tipe `WebDriver.Options`. Di dalam kelas ini terdapat *method* `window()` yang mengembalikan *interface* dengan tipe `WebDriver.Window`.

2.3.2 Interface WebDriver.Window

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

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

2.3.3 Kelas ChromeDriver

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

- `public ChromeDriver()`
Constructor ini membuat instans dengan tipe `ChromeDriver`.

2.3.4 Kelas FirefoxDriver

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

- `public FirefoxDriver()`
Constructor ini membuat instans dengan tipe `FirefoxDriver`.

2.3.5 Kelas OperaDriver

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

- `public OperaDriver()`
Constructor ini membuat instans dengan tipe `OperaDriver`.

2.3.6 Kelas InternetExplorerDriver

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

- `public InternetExplorerDriver()`
Constructor ini membuat instans dengan tipe `InternetExplorerDriver`.

2.3.7 Kelas EdgeDriver

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

- `public EdgeDriver()`
Constructor ini membuat instans dengan tipe `EdgeDriver`.

2.3.8 Kelas HtmlUnitDriver

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

- `public HtmlUnitDriver()`
Constructor ini membuat instans dengan tipe `HtmlUnitDriver`.

2.3.9 Interface OutputType

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

- `static final OutputType<String> BASE64`
 Berfungsi untuk mendapatkan *screenshot* dalam bentuk *base64 data*.
- `static final OutputType<byte[]> BYTES`
 Berfungsi untuk mendapatkan *screenshot* dalam bentuk *raw bytes*.
- `static final OutputType<java.io.File> FILE`
 Berfungsi untuk mendapatkan *screenshot* dalam bentuk *temporary file* yang akan dihapus setelah program keluar dari *Java Virtual Machine*.

2.3.10 Interface TakesScreenshot

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

- `<X> X getScreenshotAs(OutputType<X> target) throws WebDriverException`
Method ini berfungsi untuk mengambil *screenshot* dan mengembalikan hasilnya.
 Kembalian: objek hasil *screenshot*
 Parameter: tipe *output* yang diinginkan (lihat subbab 2.3.9).

2.4 Apache Commons CLI

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

2.4.1 CommandLineParser

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

- CommandLine parse(Options options, String[] arguments) throws ParseException
Berfungsi untuk melakukan *parsing* argumen Command Line berdasarkan pada *option* yang telah ditentukan. *Method* ini melempar *ParseException* jika terjadi masalah saat melakukan *parsing*.
Parameter: *option* yang telah ditentukan, argumen Command Line.
Kembalian: objek *CommandLine*.
- CommandLine parse(Options options, String[] arguments, boolean stopAtNonOption) throws ParseException
Berfungsi untuk melakukan *parsing* argumen Command Line berdasarkan pada *option* yang telah ditentukan.
Parameter: *option* yang telah ditentukan, argumen Command Line, dan suatu *boolean* yang menentukan apakah *parsing* dihentikan jika terdapat argumen yang tidak valid. Jika bernilai *true*, *parsing* akan dihentikan dan semua argumen yang sudah diuraikan akan ditambahkan ke objek *CommandLine*. Jika bernilai *false*, akan dilempar *ParseException* bila terdapat Option yang tidak valid.
Kembalian: objek *CommandLine*.

2.4.2 DefaultParser

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

- CommandLine parse(Options options, String[] arguments) throws ParseException
Berfungsi untuk melakukan *parsing* argumen Command Line berdasarkan pada *option* yang telah ditentukan. *Method* ini melempar *ParseException* jika terjadi masalah saat melakukan *parsing*.
Parameter: *option* yang telah ditentukan, argumen Command Line.
Kembalian: objek *CommandLine*.
- CommandLine parse(Options options, String[] arguments, boolean stopAtNonOption) throws ParseException
Berfungsi untuk melakukan *parsing* argumen Command Line berdasarkan pada *option* yang telah ditentukan.
Parameter: *option* yang telah ditentukan, argumen Command Line, dan suatu *boolean* yang menentukan apakah *parsing* dihentikan jika terdapat argumen yang tidak valid. Jika bernilai *true*, *parsing* akan dihentikan dan semua argumen yang sudah diuraikan akan ditambahkan ke objek *CommandLine*. Jika bernilai *false*, akan dilempar *ParseException* bila terdapat Option yang tidak valid.
Kembalian: objek *CommandLine*.

2.4.3 CommandLine

Kelas ini merepresentasikan kumpulan Option hasil *parsing*. Berikut ini adalah beberapa *method* 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 Option. Kembalian: kumpulan nilai dari suatu Option dengan tipe *array of String*. Jika Option tidak ditemukan, akan mengembalikan *null*.
- `public Option[] getOptions()`
Berfungsi untuk mengembalikan Option-option hasil *parsing*.
Kembalian: *array* dari Option hasil *parsing*.

2.4.4 Options

Kelas ini merepresentasikan kumpulan dari objek *Option*, yang mendeskripsikan kemungkinan *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.

2.4.5 Option

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

- `public String getLongOpt()`
Berfungsi untuk mendapatkan nama panjang dari suatu *option*.
Kembalian: nama panjang dari suatu *option*.
- `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*.
- `public String[] getValues()`
Berfungsi untuk mendapatkan nilai-nilai dari Option.
Kembalian: nilai-nilai dari Option berupa *array of String*. Jika tidak ada nilai, akan mengembalikan *null*.
- `public boolean hasArg()`
Berfungsi untuk mengetahui apakah suatu *option* membutuhkan argumen.
Kembalian: *true* jika *option* ini membutuhkan argumen, *false* jika *option* ini tidak membutuhkan argumen.
- `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.
- `public String getDescription()`
Berfungsi untuk mendapatkan deskripsi dari suatu *option*.
Kembalian: deskripsi dari *option* ini.
- `public String getArgName()`
Berfungsi untuk mendapatkan nama dari suatu *option*.
Kembalian: nama dari argumen suatu *option*.

2.4.6 Option.Builder

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

- `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*.
- `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*.
- `public Option.Builder hasArg()`
Berfungsi untuk menyatakan bahwa *option* ini membutuhkan argumen.
Kembalian: objek *Option.Builder* yang bisa digunakan untuk *method chaining*.
- `public Option.Builder hasArgs()`
Berfungsi untuk menyatakan bahwa *option* ini membutuhkan argumen, dimana jumlah argumen bisa lebih dari satu.
Kembalian: objek *Option.Builder* yang bisa digunakan untuk *method chaining*.
- `public Option.Builder argName(String argName)`
Berfungsi untuk memberi nama pada argumen.
Parameter: nama argumen.
Kembalian: objek *Option.Builder* yang bisa digunakan untuk *method chaining*.
- `public Option.Builder required(boolean required)`
Berfungsi untuk menyatakan bahwa *option* ini wajib ada.
Parameter: variabel bertipe *boolean* yang menentukan apakah *option* ini wajib ada.
Kembalian: objek *Option.Builder* yang bisa digunakan untuk *method chaining*.
- `public Option build()`
Berfungsi untuk membuat objek *Option* berdasarkan nilai pada *Option.Builder*.
Kembalian: objek *Option*.

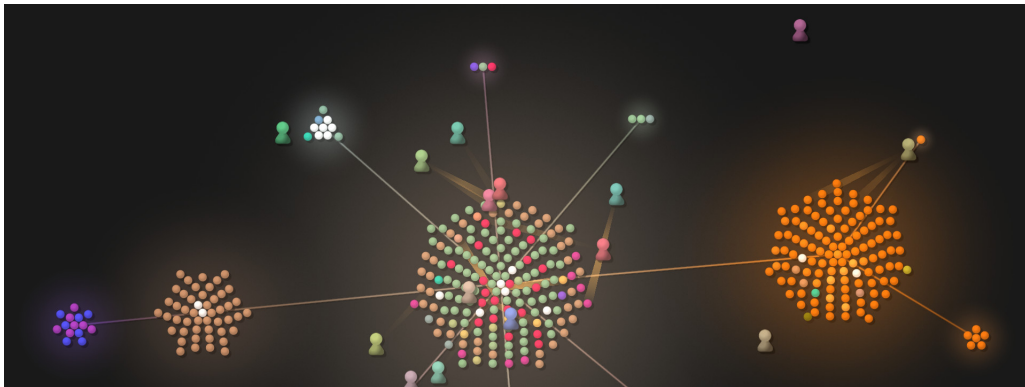
BAB 3

ANALISIS

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

3.1 Analisis Aplikasi Sejenis

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



Gambar 3.1: Visualisasi proyek perangkat lunak menggunakan Gource.

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

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

1. `gource -[WIDTH]x[HEIGHT]`
Opsi ini berfungsi untuk mengatur resolusi layar dari animasi. Parameter dari opsi ini adalah lebar dan panjang layar dalam satuan piksel.
2. `gource -camera-mode [MODE]`
Opsi ini berfungsi untuk mengatur mode kamera pada Gource. Parameter dari opsi ini adalah mode dari kamera. Terdapat dua mode yaitu *overview* dan *track*. Dalam mode *track*, kamera

bergerak mengikuti *user* yang sedang aktif. Dalam mode *overview*, kamera menampilkan seluruh repositori.

3. `gource -path [PATH]`
Opsi ini berfungsi untuk mengatur *path* dari direktori yang akan dibuat animasinya. Opsi dari parameter ini adalah *path* dari direktori.
4. `gource -start-date [YYYY-MM-DD hh:mm:ss +tz] -stop-date [YYYY-MM-DD hh:mm:ss +tz]`
Opsi ini berfungsi untuk mengatur periode waktu dalam menampilkan animasi. Parameter dari opsi ini adalah waktu mulai dan waktu akhir dalam format "YYYY-MM-DD hh:mm:ss +tz". Dimana YYYY adalah tahun, MM adalah bulan, DD adalah tanggal, hh adalah jam, mm adalah menit, ss adalah detik, dan +tz adalah zona waktu. Parameter jam, menit, detik, dan zona waktu bersifat opsional.
5. `gource -bloom-multiplier [FLOAT]`
Opsi ini berfungsi untuk mengatur radius dari efek *bloom*. Parameter dari opsi ini adalah radius dalam format bilangan riil.
6. `gource -bloom-intensity [FLOAT]`
Opsi ini berfungsi untuk mengatur intensitas dari efek *bloom*. Parameter dari opsi ini adalah intensitas *bloom* dalam format bilangan riil.
7. `gource -disable-bloom`
Opsi ini berfungsi untuk menonaktifkan animasi *bloom*.
8. `gource -date-format [FORMAT]`
Opsi untuk mengatur format waktu yang ditampilkan pada bagian tengah atas. Opsi dari parameter ini adalah format waktu dalam bentuk *string*.
9. `gource -background [FFFFFF]`
Opsi ini berfungsi untuk mengatur warna *background*. Parameter dari opsi ini adalah warna dalam format heksadesimal.
10. `gource -background-image [IMAGE]`
Opsi ini berfungsi untuk mengatur gambar *background*. Parameter dari opsi ini adalah nama *file* dari gambar.
11. `gource -font-size [SIZE]`
Opsi ini digunakan untuk mengatur ukuran *font* pada tulisan *title* dan tanggal. Parameter dari opsi ini adalah ukuran *font*.
12. `gource -font-colour [FFFFFF]`
Opsi ini digunakan untuk mengatur warna *font* pada tulisan *title* dan tanggal. Parameter dari opsi ini adalah warna *font* dalam format heksadesimal.
13. `gource -logo [IMAGE]`
Opsi ini berfungsi untuk memasukkan logo. Parameter dari opsi ini adalah nama *file* dari gambar.
14. `gource -logo-offset [X]x[Y]`
Opsi ini berfungsi untuk mengatur posisi dari logo. Parameter dari opsi ini adalah posisi x dan posisi y dari logo.
15. `gource -title [TITLE]`
Opsi ini berfungsi untuk memberi judul. Dimana judul tersebut ditampilkan pada pojok kiri bawah layar.

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

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

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

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

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

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

- *bloom*: efek *bloom*.
- *date*: waktu.
- *dirnames*: nama direktori.
- *files*: ikon dari berkas.
- *filenames*: nama berkas.
- *root*: *root directory*.
- *users*: ikon dari *user*.
- *usernames*: nama dari *user*.

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

Gource dapat digunakan untuk berbagai macam proyek perangkat lunak. Program pada skripsi ini hanya akan berfokus untuk proyek perangkat lunak berbasis *web*. Tidak seperti Gource yang menampilkan direktori dan *file* pada animasi, program pada skripsi ini menampilkan *screenshot* dari halaman suatu *website*.

3.2 Analisis Penggunaan JGit dan Selenium WebDriver

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

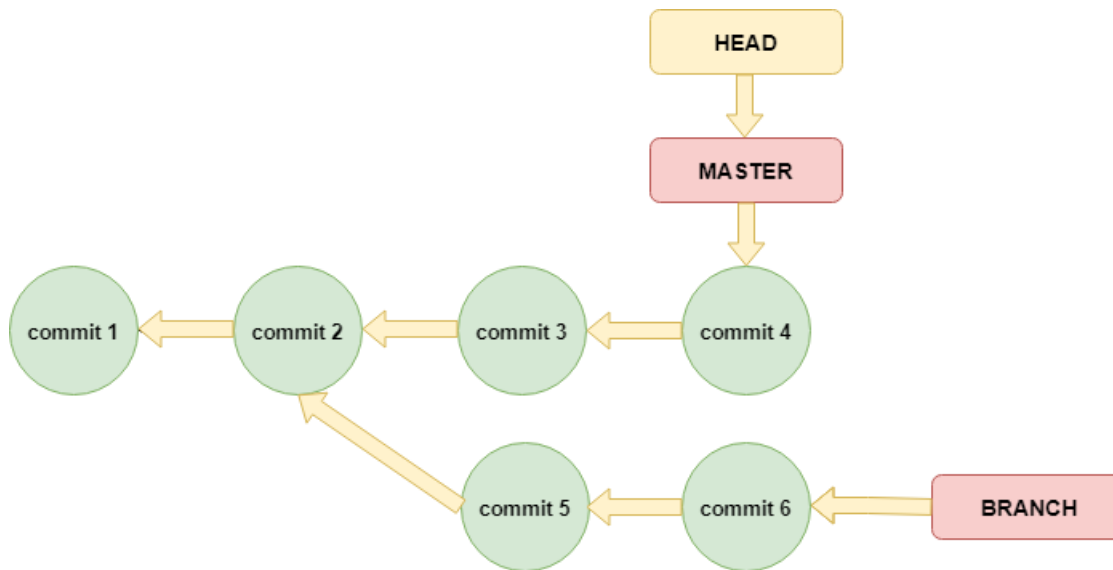
3.2.1 Analisis Penggunaan JGit

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 untuk berinteraksi dengan suatu repositori Git. 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 untuk mendapatkan seluruh histori *commit* dari perangkat lunak berbasis *web*.



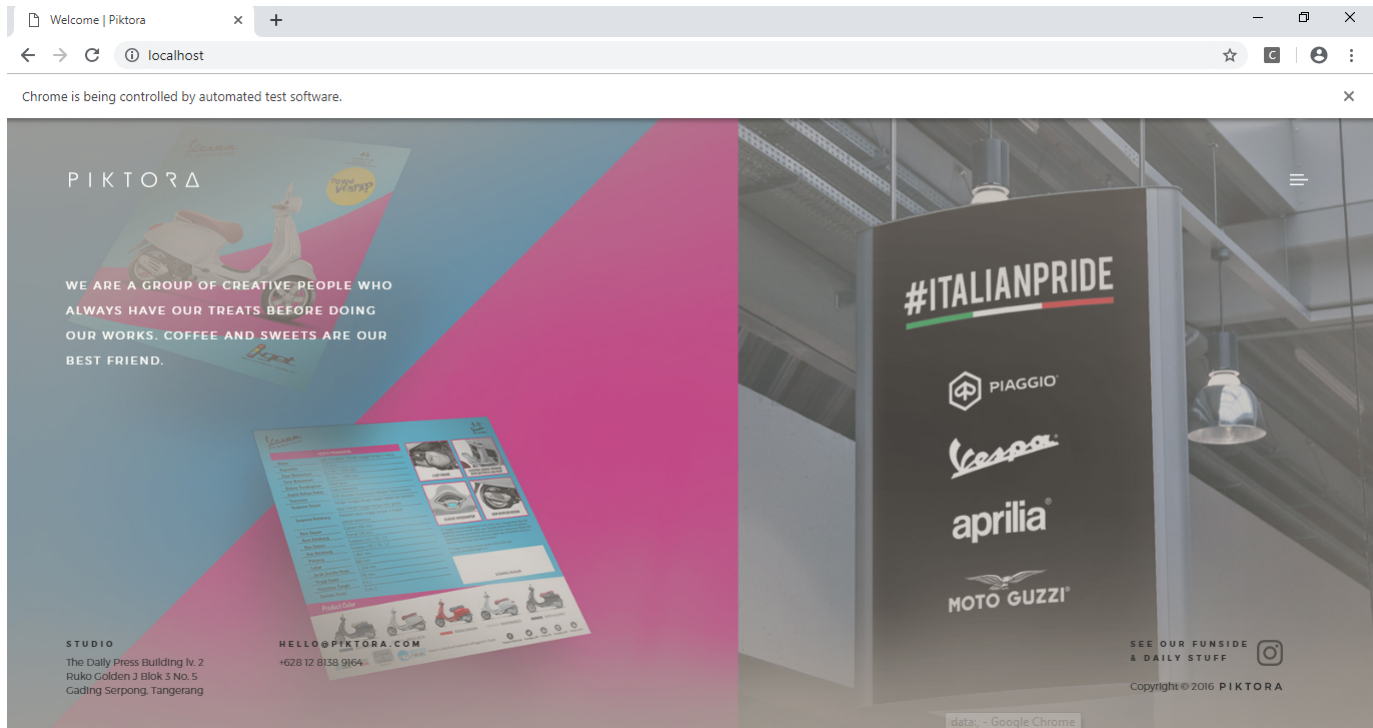
Gambar 3.2: Histori *commit* direpresentasikan sebagai Directed Acyclic Graph.

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 akan digunakan untuk menelusuri *commit* dari perangkat lunak berbasis *web*. Penelusuran ini dilakukan dengan tujuan mendapatkan *screenshot* halaman *web*.

3.2.2 Analisis Penggunaan Selenium WebDriver

Selenium WebDriver dapat digunakan untuk mengotomatisasi *web browser*. Pada analisis ini dibahas kelas-kelas pada *library* Selenium WebDriver yang digunakan untuk mengotomatisasi *web browser*. Kelas-kelas yang dipakai yaitu `WebDriver`, `ChromeDriver`, dan `TakesScreenshot`.

`WebDriver` merupakan *interface* utama yang digunakan untuk pengujian. `WebDriver` dapat diinisialisasi menggunakan *object* dengan tipe `ChromeDriver`. Setelah melakukan inisialisasi pada `WebDriver`, *browser* akan dijalankan. Untuk membuka suatu halaman *web*, digunakan *method* `get()` dengan parameter alamat URL. Gambar 3.3 menunjukkan *Chrome browser* yang dikontrol oleh `ChromeDriver`.



Gambar 3.3: Browser yang dikontrol oleh ChromeDriver.

TakesScreenshot merupakan *interface* yang digunakan untuk menangkap *screenshot* halaman *web* pada WebDriver. Kelas ChromeDriver mengimplementasikan *interface* ini. *Method* yang digunakan untuk menangkap *screenshot* adalah `getScreenshotAs()`, diikuti dengan parameter bertipe `OutputType`. `OutputType` ini merupakan *interface* yang digunakan untuk menentukan tipe *output* dari hasil *screenshot*. `OutputType` yang digunakan pada program dalam skripsi ini adalah `File`.

3.3 Analisis Fitur Aplikasi yang Dibangun

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

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

1. `-project-path [PATH]`
Opsi ini berfungsi untuk mengatur *path* dari direktori yang akan dibuat animasinya. Parameter dari opsi ini adalah *path* dari proyek perangkat lunak web yang terekam oleh Git. Opsi ini wajib ada.
2. `-before-capture [TERMINAL-COMMAND]`
Opsi ini berfungsi untuk menjalankan *terminal command*. *Terminal command* ini dijalankan sebelum melakukan *screenshot*. Parameter dari opsi ini adalah *terminal command* dalam bentuk *string* dan harus diapit dengan tanda petik ganda. Opsi ini bersifat opsional.
3. `-capture-url [URL 1, URL 2, URL 3, URL 4]`
Opsi ini berfungsi untuk mengatur alamat *url* dari halaman *web*, dimana dilakukan pengambilan *screenshot* pada halaman ini. Jumlah halaman yang dicapture bisa lebih dari satu, dengan jumlah maksimal empat halaman. Parameter dari opsi ini adalah alamat *url* dari halaman *web*, dengan jumlah maksimal empat alamat *url*. Opsi ini wajib ada.

4. `-seconds-per-commit` [SECONDS]

Opsi ini berfungsi untuk mengatur durasi munculnya satu *commit* pada animasi. Parameter dari opsi ini adalah durasi munculnya satu *commit* dalam satuan detik. Parameter harus berupa bilangan bulat atau riil, dimana nilainya lebih besar dari nol. Opsi ini bersifat opsional.

5. `-title` [TITLE]

Opsi ini berfungsi untuk memberi judul. Dimana judul tersebut ditampilkan pada pojok kiri bawah layar. Opsi ini bersifat opsional.

6. `-logo` [IMAGE]

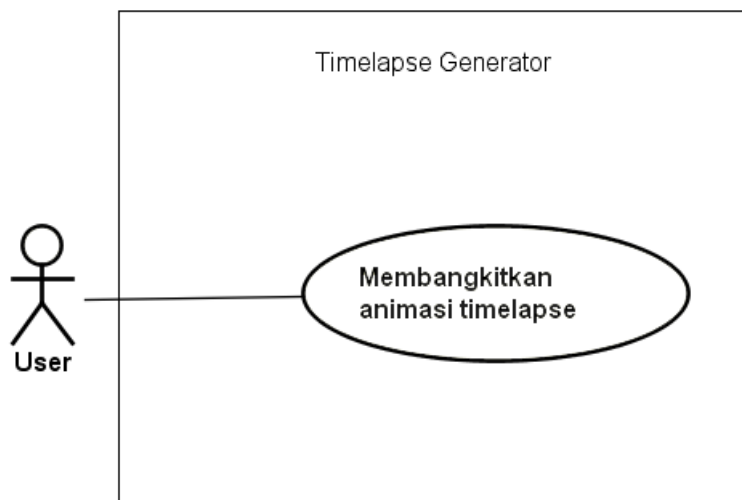
Opsi ini berfungsi untuk memasukkan logo. Dimana logo tersebut ditampilkan pada pojok kanan bawah layar. Parameternya adalah *path* dari *file* gambar. Opsi ini bersifat opsional.

7. `-start-commit` [ID] `-stop-commit` [ID]

Opsi ini berfungsi untuk mengatur rentang *commit* yang akan dibuat animasinya. Parameter dari opsi ini adalah *commit ID* awal dan *commit ID* akhir, dimana panjang *commit ID* adalah tujuh karakter. Opsi ini bersifat opsional.

Opsi `-path`, `-title`, dan `-image` mengacu pada opsi yang terdapat pada Gource. Opsi `-start-commit`, `-stop-commit` dan `-seconds-per-commit` mengacu pada Gource, dengan sedikit penyesuaian. Opsi `-seconds-per-day` pada Gource menyatakan durasi munculnya satu hari, sedangkan Opsi `-seconds-per-commit` menyatakan durasi munculnya satu *commit*. Opsi `-start-date` `-stop-date` pada Gource dan opsi `-start-commit` `-stop-commit` mengatur periode dalam menampilkan animasi. Pada Gource, periode yang digunakan adalah rentang waktu berupa tanggal, bulan, dan tahun. Sedangkan pada opsi `-start-commit` `-stop-commit`, rentangnya berupa *commit ID*. Opsi `-before-capture` dan `-capture-url` tidak mengacu pada Gource. Kedua opsi tersebut secara khusus dibuat karena program pada skripsi ini membuat animasi *timelapse* dari kumpulan halaman *web*, sedangkan program Gource membuat animasi dari struktur *folder* atau *file*.

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



Gambar 3.4: *Use case diagram* perangkat lunak.

Nama	Membangkitkan animasi <i>timelapse</i>
Deskripsi	Fitur untuk membangkitkan animasi <i>timelapse</i> 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"> 1. Sistem membaca masukan <i>input</i> dari Command Line Option. 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. 3. Sistem menggabungkan file <i>screenshot</i> menjadi satu file bertipe GIF.

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

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

1. Program membaca argumen Command Line menggunakan Apache Commons CLI.
2. Program mendapatkan seluruh *commit* histori dari proyek perangkat lunak berbasis web menggunakan JGit.
3. Program membuka semua *browser* menggunakan Selenium WebDriver. Jumlah *browser* bergantung pada jumlah argumen dari Option `-capture-url`.
4. Program melakukan *checkout* dalam suatu rentang *commit* tertentu. Jika tidak terdapat Option `-start-commit` dan Option `-stop-commit`, akan dilakukan *checkout* ke semua *commit*.
5. Program melakukan *checkout* ke suatu *commit*.
6. Program menjalankan *script* PHP jika terdapat Option `-before-capture`.
7. Program membuka setiap URL yang didapatkan dari Option `-capture-url` menggunakan Selenium WebDriver. Setiap *browser* membuka URL yang berbeda.
8. Selenium WebDriver kemudian mengambil *screenshot* pada semua *browser*.
9. Jika saat ini sedang berada pada *commit* terakhir, lanjut ke langkah berikutnya. Jika tidak, ulangi langkah 5-8 untuk *commit* selanjutnya.
10. Jika terdapat lebih dari satu *browser*, hasil *screenshot* dari setiap *browser* digabungkan menjadi satu gambar baru.
11. Program menambahkan judul di pojok kiri bawah jika terdapat Option `-title`.
12. Program menambahkan logo di pojok kanan bawah jika terdapat Option `-logo`.
13. Menggabungkan semua *file* gambar menjadi satu *file* bertipe GIF.

Penulis menggunakan *library* yang didapatkan dari internet¹ untuk menggabungkan gambar menjadi *file* GIF.

3.4 Prapengujian Website Piktora

Pengujian dilakukan dengan proyek Piktora sebagai input dari program. Piktora² merupakan situs *web* yang menawarkan layanan *creative design* dan *branding*. Layanan yang ditawarkan berupa *graphic design* untuk poster, *banner*, *website*, dan aplikasi *mobile*. Repositori dari situs *web* ini disimpan pada Gitlab³, dibutuhkan akses khusus untuk membuka repositori tersebut. Piktora dibangun dengan php menggunakan framework Code Igniter. Piktora menggunakan MySQL sebagai basis datanya.

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

- Processor: Intel Core i3 4030U
- RAM: 6GB
- Sistem Operasi: Windows 10 pro 64-bit
- Versi Apache HTTP Server: 2.4.29
- Versi MySQL Server: 5.5.5
- Versi Netbeans: 8.1
- Versi Google Chrome: 73.0.3683.86

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

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

Listing 3.1: Histori *commit* pada proyek Piktora

```

1 | 315d374 - Oct 31 16:52:46 2016 : Basic CI files + htaccess & webconfig + database.
   | php ignore
2 |
3 | 27ce3d4 - Nov 5 13:12:43 2016 : setup environment for piktora
4 | 65f0c9c - Nov 5 19:22:58 2016 : * create structure for all pages * add dummy
   | images
5 |
6 | bffbae1 - Nov 8 18:00:32 2016 : - basic structure (navbar semi complete) - add
   | fonts
7 |
8 | 5c59916 - Nov 8 19:51:18 2016 : implement navbar, footer, and projects/ page
9 | 7738380 - Nov 8 20:05:27 2016 : fix pc and ipad navbar fontsize
10 | 26bdbbe - Nov 8 20:16:33 2016 : fix position image for desktop /projects
11 | 3db3ce8 - Nov 9 00:28:27 2016 : - implements project details page - fix some minor
   | issue - add some project image
12 |
13 | 5ef34fa - Nov 13 13:01:06 2016 : implement about us (raw version)

```

¹<http://elliott.kroo.net/software/java/GifSequenceWriter>

²<http://www.piktora.com>

³<https://gitlab.com/PNDevworks/Piktora>

```

14 3caf535 - Nov 15 11:55:15 2016 : fix minor issues view/about: - background-image
    position. Make it to the center position - slick.js img need to set to inline
15
16 c5eb3b6 - Nov 15 13:02:42 2016 : implement /welcome page
17 ade9216 - Nov 15 13:12:08 2016 : fix minor issues: - move style footer to global -
    add space before PIKTORA
18
19 c77b5b3 - Nov 18 18:18:25 2016 : implement /contact
20 b42b819 - Nov 18 21:22:22 2016 : cange a href to style cursor: pointer
21 3eb7af8 - Nov 21 16:09:40 2016 : .htaccess to be compatible with cloud kilat
22 e87e84b - Nov 22 14:53:45 2016 : - change vw to 100% - add captcha
23 ff8d829 - Nov 22 15:22:53 2016 : Solved captcha font load: use otf instead of ttf
    Also: create directory assets/img/captcha and ignore everything inside
24
25 dc87342 - Nov 22 15:49:03 2016 : - implement captcha code - remove wrong css
26 9ebe433 - Nov 22 16:12:44 2016 : add scroll feature in project/
27 f0f7270 - Nov 23 15:17:42 2016 : Added Google PHP Client v2 See https://github.com/google/google-api-php-client
28
29 57a239b - Nov 23 15:19:57 2016 : Merge origin/master
30 e2dfebe - Nov 27 14:42:24 2016 : - remove blue outline when click with slick -
    change background image in about to newest one
31
32 a19e7f2 - Nov 27 15:24:42 2016 : Detailing dari Edina: 1. Hal. Project Detail,
    font coba diperkecil saja mungkin ya. 3. Beberapa ukuran font dan spacing ada
    yang kurang pas sedikit, terlampir detail revisinya ya (file pdf) 4. Footer
    dibuat selalu stay terus di bagian bawah dengan posisi yang selalu sama. Di home
    & contact sudah sama, namun di hal. product posisinya agak lebih naik.
33
34 3d79d0a - Nov 27 15:29:58 2016 : fix minor issue
35 0fcd958 - Nov 28 10:11:06 2016 : add raw admin contents
36 add3974 - Nov 28 12:03:21 2016 : add summernote, implement read project
37 0680488 - Nov 29 12:38:23 2016 : update admin for projects
38 fbe7639 - Nov 29 13:10:30 2016 : implement admin for home
39 db0cedd - Nov 29 13:38:29 2016 : - implementasi database bagian user - upload 9
    gambar contoh project
40
41 0fe9aaf - Nov 29 14:17:20 2016 : ubah warna garis captcha
42 f2326dd - Nov 29 14:44:51 2016 : - lewati proses otentikasi sementara
43 f78cdb4 - Dec 2 12:10:47 2016 : (trying to) fix issue #2
44 ef9b62b - Dec 2 17:09:58 2016 : revisi dari edina ke-2
45 c689aa8 - Dec 2 17:11:13 2016 : perbaikan admin sedikit
46 c4e9576 - Dec 2 17:14:06 2016 : perbaikan di /contact, kelewat
47 02d04f1 - Dec 5 14:55:20 2016 : tambah wording
48 a4e4858 - Dec 5 15:08:59 2016 : perbaikan kata2 sedikit
49 bbd82c2 - Dec 6 10:41:40 2016 : implementasi email
50 f8c64fc - Dec 6 11:03:08 2016 : change to httpdocs
51 eb49c2b - Dec 6 11:35:20 2016 : hapus migrasi script di admin
52 ace1988 - Dec 6 11:39:00 2016 : change overflow to auto
53 627e65b - Dec 6 11:45:26 2016 : modify database back to local
54 0896f81 - Dec 7 16:08:30 2016 : update home versi mobile jadi baru (revisi dari
    Edina)
55
56 5cf1292 - Dec 7 16:21:01 2016 : ubah background di about menjadi tidak pecah
57 c83f4aa - Thu Dec 15 15:04:30 2016 : remove piktora secrets
58 57f5ea4 - Thu Dec 15 15:09:43 2016 : remove unimportant data
59 7931c21 - Dec 24 18:40:41 2016 : edit wording
60 9b0a302 - Dec 25 06:03:50 2016 : Another wording fix
61 f1ea410 - Thu Jan 5 15:23:32 2017 : fix instagram link
62 1880a88 - Thu Jan 5 15:24:12 2017 : Merge branch 'master' of https://github.com/pascalalfadian/Piktora
63
64 286aa78 - Jan 16 12:48:45 2017 : Perbaikan wording di admin edit project

```

```

65 | 33702c2 – Feb 21 13:31:08 2017 : change email sender to piktora@mailgun.dnartworks.
    | com.au
66 |
67 | 18c39ef – Thu Apr 13 15:21:49 2017 : Test commit (in gitlab). Nothing much important
68 | 9bfde3c – Apr 17 15:09:54 2017 : add ignore sftp-config.json
69 | 38711f0 – Apr 17 15:15:03 2017 : fix bug ugly display when projects too high
70 | 9f041ef – May 15 10:40:16 2017 : set insta url to https://www.instagram.com/
    | piktorastudio/
71 |
72 | 6a085c1 – Dec 12 14:38:38 2017 : Update company address
73 | 89000be – Jan 12 12:25:30 2018 : Update new company address

```

3.4.1 Perbedaan Letak *File*

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

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* PHP ini akan mengecek letak *file* "index.php" pada *folder* "www" dan "httpdocs". *Script* kemudian akan mengecek *directory root* apache pada *file* "httpd.conf". Jika *directory root* sudah mengarah ke *folder* tempat "index.php" berada, maka *script* tidak akan mengubah isi *file* "httpd.conf". Jika *directory root* tidak mengarah ke *folder* tempat "index.php" berada, maka *script* akan mengubah *directory root* pada *file* "httpd.conf" dan melakukan *restart* pada apache. *Script* PHP ini akan dijalankan sebelum melakukan migrasi *database*.

3.4.2 Permasalahan Konfigurasi *Database*

Pada *commit* f8c64fc(6 Desember 2016) s.d. ace1988(6 Desember 2016), halaman *web* tidak bisa dibuka. Hal ini disebabkan karena perbedaan konfigurasi pada *file* "database.php". Pada *commit* f8c64fc(6 Desember 2016) s.d. 57f5ea4(15 Desember 2016) dan *commit* flea410(5 Januari 2017), di dalam *file* "database.php" terdapat *password*. *Commit* lainnya tidak terdapat *password* pada *file* "database.php". Penulis menggunakan *password* "piktora" pada konfigurasi *database* di MySQL Server.

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

```

1 | $active_group = 'default';
2 | $query_builder = TRUE;
3 |
4 | $db['default'] = array(
5 |     'dsn' => '',
6 |     'hostname' => 'localhost',
7 |     'username' => 'piktora',
8 |     'password' => 'dmHx64%6',
9 |     'database' => 'piktora',
10 |    'dbdriver' => 'mysqli',
11 |    'dbprefix' => '',
12 |    'pconnect' => FALSE,
13 |    'db_debug' => (ENVIRONMENT !== 'production'),
14 |    'cache_on' => FALSE,
15 |    'cachedir' => '',
16 |    'char_set' => 'utf8',
17 |    'dbcollat' => 'utf8_general_ci',
18 |    'swap_pre' => '',

```

```

19 | 'encrypt' => FALSE,
20 | 'compress' => FALSE,
21 | 'stricton' => FALSE,
22 | 'failover' => array() ,
23 | 'save_queries' => TRUE
24 | );

```

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.

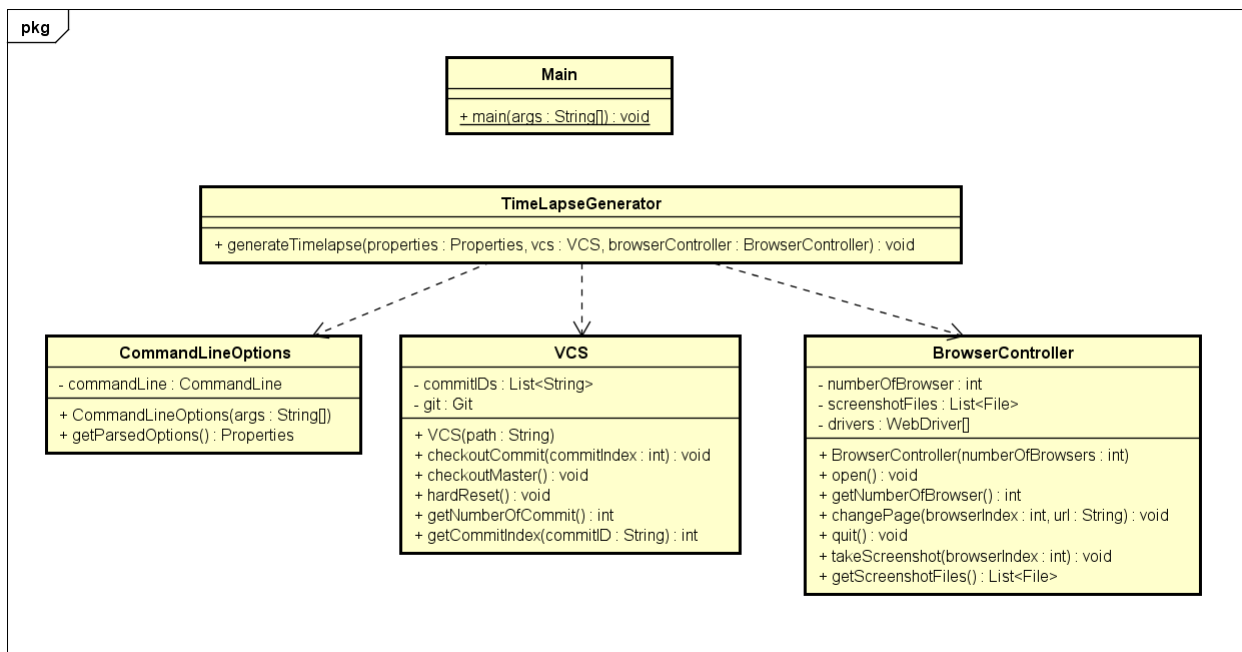
Sama seperti subbab 3.4.1, solusi untuk mengatasi masalah ini adalah dengan menambahkan *command line option* yang akan menjalankan *terminal command*. Dimana *terminal command* ini berisi perintah untuk menjalankan *script* PHP. *Script* ini akan melakukan dua pekerjaan. Pertama, *script* akan menghapus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

BAB 4

PERANCANGAN

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

4.1 Perancangan Kelas



Gambar 4.1: Diagram kelas.

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

- BrowserController

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

- private final WebDriver[] drivers
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 *screenshot*nya.

- 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.

- VCS

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

- private final Git git
Atribut ini digunakan untuk melakukan interaksi pada proyek perangkat lunak yang terekam oleh Git.
- private final List<String> commitIDs
Atribut ini digunakan untuk menampung seluruh *commit* ID dari hasil penelusuran histori.

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

- public VCS(String path)
Constructor dari kelas ini. Berfungsi untuk menginisialisasi variabel git dan mendapatkan seluruh histori commit pada proyek perangkat lunak berbasis web. Parameternya adalah *path* dari proyek perangkat lunak berbasis web.
 - public void checkoutCommit(int commitIndex)
Berfungsi untuk melakukan *checkout* ke *commit* tertentu. Parameter dari *method* ini adalah indeks dari variabel commitIDs.
 - public void checkoutMaster()
Berfungsi untuk melakukan *checkout* ke *commit* terakhir.
 - public void hardReset()
Berfungsi untuk melakukan operasi Git Reset dengan tipe *hard reset*. Operasi ini menghapus perubahan pada *working tree* di *commit* tertentu.
 - public int getNumberOfCommit()
Berfungsi untuk mendapatkan jumlah *commit*.
 - public int getCommitIndex(String commitID)
Berfungsi untuk mendapatkan indeks dari variabel commitIDs. Parameternya adalah *Commit ID* yang akan dicari indeksinya.
- TimeLapseGenerator
Kelas ini digunakan untuk membangkitkan animasi *timelapse*. Berikut adalah *method* yang dimiliki oleh kelas ini:
 - public void generateTimelapse(Properties properties, VCS vcs, BrowserController browserController)
Berfungsi untuk membangkitkan animasi *timelapse* berdasarkan langkah-langkah pada subbab 3.3. Hasil animasi berupa *file* bertipe GIF. Parameternya adalah objek yang bertipe VCS, BrowserController, dan Properties. Parameter properties menampung key 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*.

```
1 | Animasi timelapse berhasil dibuat
```

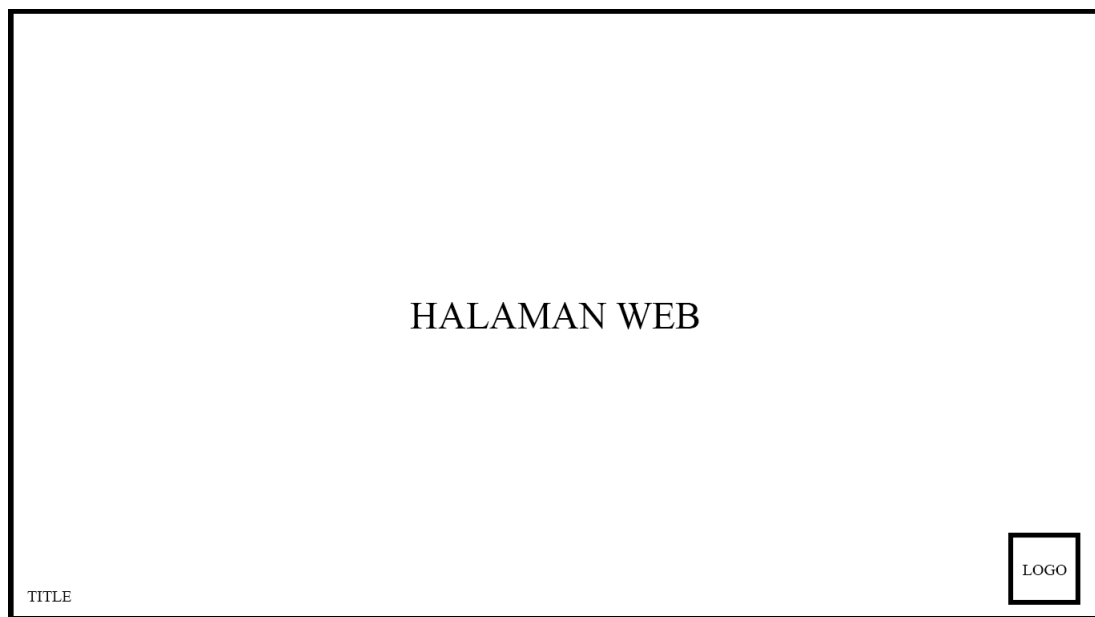
Listing 4.2: Status pesan yang akan muncul pada terminal saat program gagal membangkitkan animasi *timelapse*.

```
1 | Animasi timelapse gagal dibuat
2 | <ERROR MESSAGE>
```

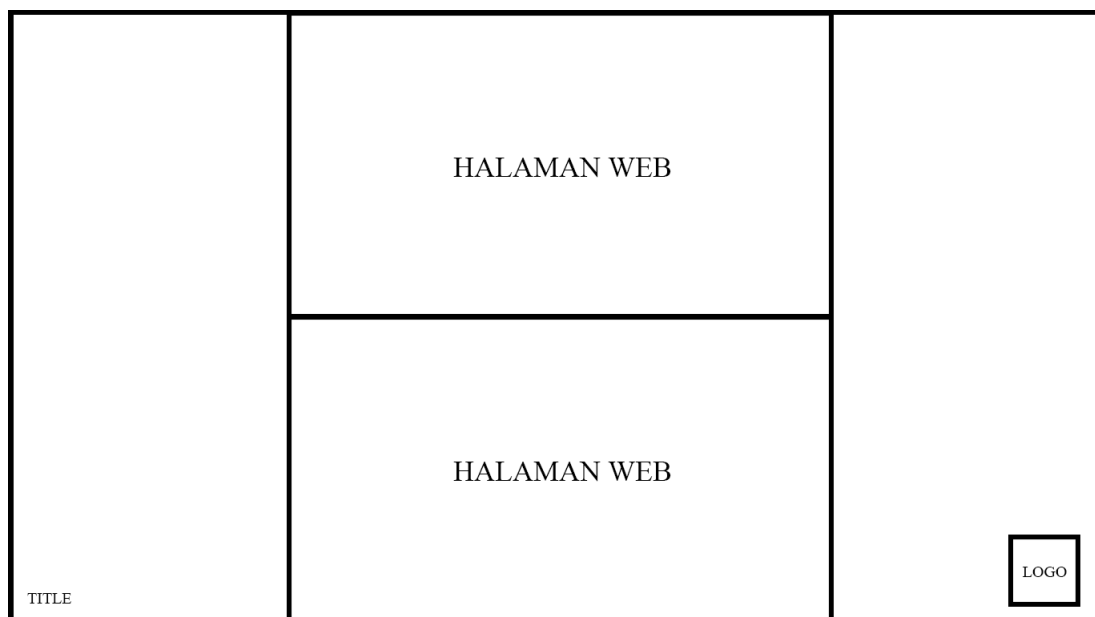
Listing 4.1 menunjukkan status yang akan ditampilkan pada terminal saat program berhasil membangkitkan animasi *timelapse*. Listing 4.2 menunjukkan status yang akan ditampilkan pada terminal saat program gagal membangkitkan animasi *timelapse*. Pada Listing 4.2, baris pertama menunjukkan bahwa program 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:

- **Missing required option: [OPTION NAME]**
Pesan ini muncul jika *user* tidak memasukkan Option yang wajib ada. Option yang wajib dimasukkan adalah `seconds-per-commit`, `project-path`, dan `capture-url`.
- **Missing argument for option: [OPTION NAME]**
Pesan ini muncul jika *user* memasukkan Option tanpa diikuti dengan argumennya. Semua Option yang terdapat pada program ini harus memiliki argumen.
- **Jumlah url yang akan dicapture maksimal 4**
Pesan ini muncul jika jumlah argumen pada Option `capture-url` lebih dari 4.
- **Seconds per commit harus lebih besar dari 0**
Pesan ini muncul jika argumen dari `seconds-per-commit` bernilai lebih kecil dari 0.
- **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.
- **Capture url tidak valid**
Pesan ini muncul jika argumen `capture-url` merupakan alamat URL yang tidak valid.
- **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 7 karakter**
Pesan ini muncul jika panjang Commit ID pada argumen Option `start-commit` tidak sama dengan tujuh karakter.
- **Panjang commit ID akhir harus 7 karakter**
Pesan ini muncul jika panjang Commit ID pada argumen Option `stop-commit` tidak sama dengan tujuh karakter.
- **Commit ID awal tidak ditemukan**
Pesan ini muncul jika Commit ID pada argumen Option `start-commit` tidak ditemukan.
- **Commit ID akhir tidak ditemukan**
Pesan ini muncul jika Commit ID pada argumen Option `stop-commit` tidak ditemukan.
- **Commit ID awal dan akhir tidak boleh sama**
Pesan ini muncul jika argumen pada Option `start-commit` dan Option `stop-commit` bernilai sama.
- **Commit ID awal dan akhir terbalik**
Pesan ini muncul jika nilai argumen pada Option `start-commit` dan Option `stop-commit` tertukar.

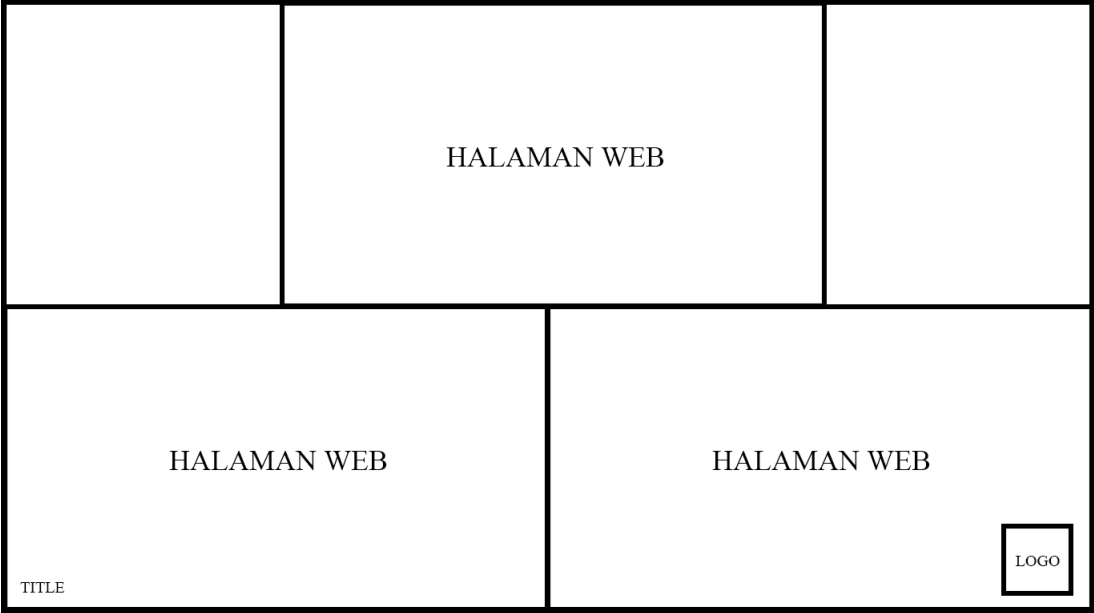
Rancangan *output* dari hasil *file* animasi *timelapse* dapat dilihat pada Gambar 4.2 sampai dengan Gambar 4.5. Gambar 4.2 menunjukkan rancangan *output* jika terdapat satu halaman *web*. Gambar 4.3 menunjukkan rancangan *output* jika terdapat dua halaman *web*. Gambar 4.4 menunjukkan rancangan *output* jika terdapat tiga halaman *web*. Gambar 4.5 menunjukkan rancangan *output* jika terdapat empat halaman *web*.



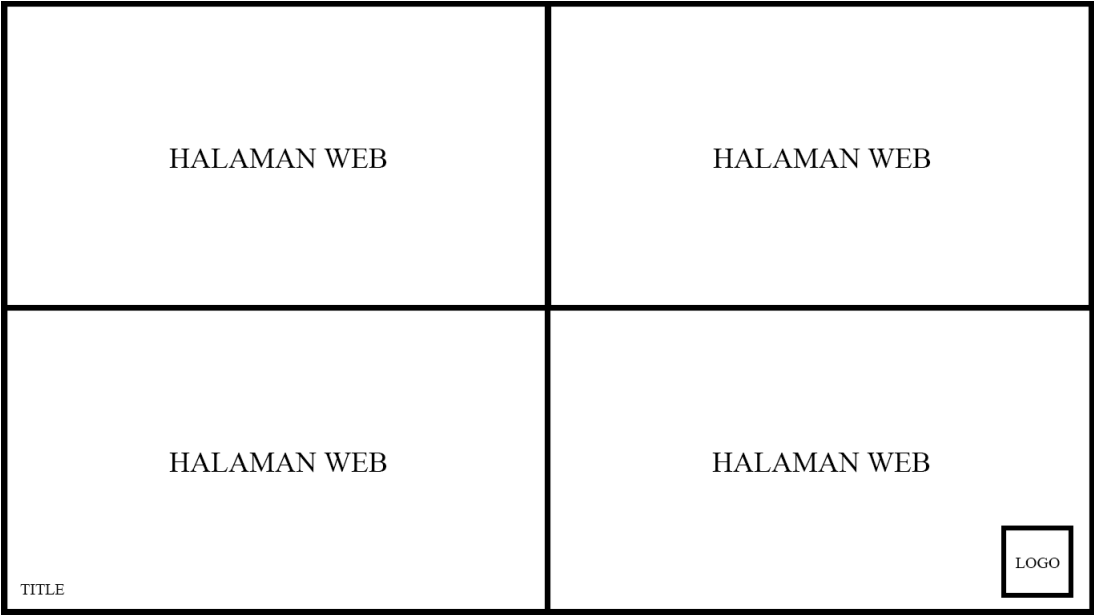
Gambar 4.2: Rancangan *output* jika terdapat satu halaman *web*.



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*.

BAB 5

IMPLEMENTASI DAN PENGUJIAN

Pada bab dijelaskan mengenai implementasi perangkat lunak dan pengujian perangkat lunak. Bagian implementasi berisi tentang lingkungan implementasi dan hasil implementasi.

5.1 Implementasi

5.1.1 Lingkungan Implementasi

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

- Processor: Intel Core i3 4030U
- RAM: 6GB
- Sistem Operasi: Windows 10 pro 64-bit
- Versi Apache HTTP Server: 2.4.29
- Versi MySQL Server: 5.5.5
- Versi Netbeans: 8.1
- Versi Google Chrome: 73.0.3683.86

5.1.2 Hasil Implementasi

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

1. Status pada Terminal

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

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

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

2. *File GIF Hasil Animasi*

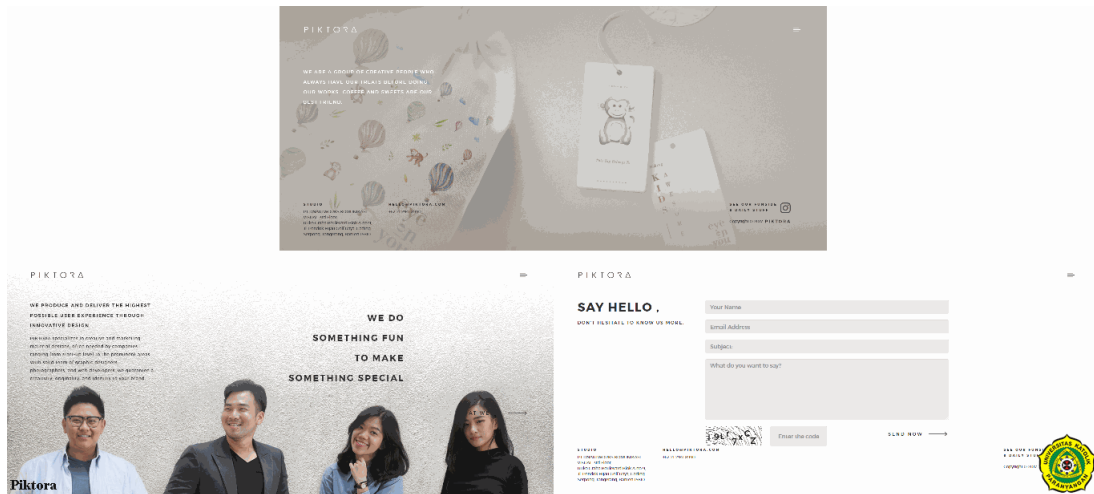
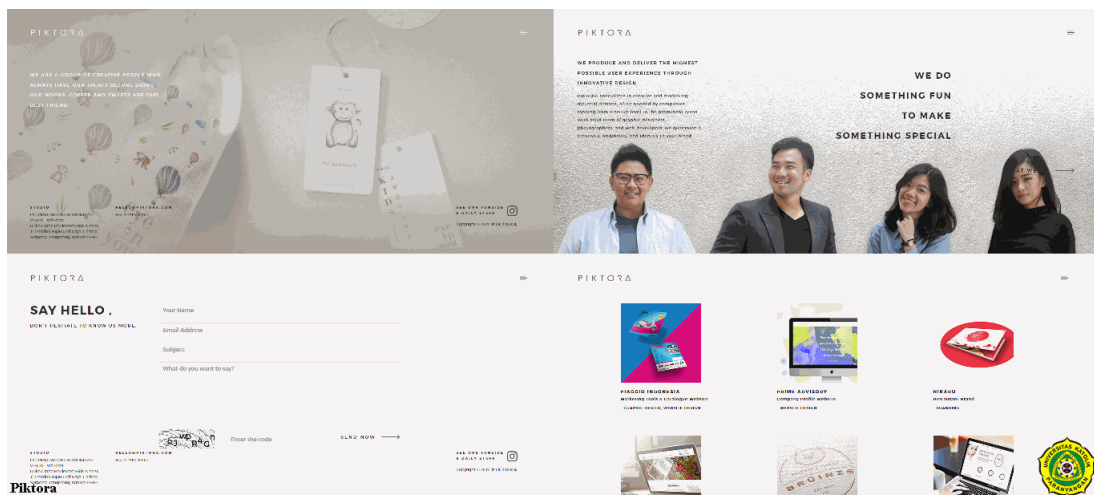
Isi dari *file GIF* hasil animasi dapat dilihat pada Gambar 5.1 - Gambar 5.4. Gambar 5.1 menunjukkan isi dari *file GIF* jika terdapat satu halaman *web*. Gambar 5.2 menunjukkan isi dari *file GIF* jika terdapat dua halaman *web*. Gambar 5.3 menunjukkan isi dari *file GIF* jika terdapat tiga halaman *web*. Gambar 5.4 menunjukkan isi dari *file GIF* jika terdapat empat halaman *web*.



Gambar 5.1: Isi dari *file GIF* jika terdapat satu halaman *web*.



Gambar 5.2: Isi dari *file GIF* jika terdapat dua halaman *web*.

Gambar 5.3: Isi dari *file* GIF jika terdapat tiga halaman *web*.Gambar 5.4: Isi dari *file* GIF jika terdapat empat halaman *web*.

5.2 Pengujian

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

```

76         driver.quit();
77     }
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     this.screenshotFiles.add(((TakesScreenshot) this.drivers[browserIndex]).getScreenshotAs(OutputType.FILE));
88 }
89
90 /**
91  * Method ini berfungsi untuk mengembalikan hasil screenshot.
92  *
93  * @return hasil screenshot berupa List of File.
94  */
95 public List<File> getScreenshotFiles() {
96     return this.screenshotFiles;
97 }
98 }

```

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().longOpt("capture-url").argName("url").desc("link yang akan di
35             capture").build());
36         options.addOption(Option.builder().required().longOpt("seconds-per-commit").argName("seconds").hasArg().desc("durasi satu
37             commit").build());
38         options.addOption(Option.builder().required().longOpt("project-path").argName("path").hasArg().desc("path proyek perangkat
39             lunak").build());
40         options.addOption(Option.builder().longOpt("before-capture").argName("script").hasArg().desc("php script yang dijalankan
41             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
51         this.commandLine = parser.parse(options, args);
52         if (this.commandLine.getOptionValues("capture-url").length > 4) {
53             throw new ParseException("Jumlah url yang akan dicapture maksimal 4");
54         }
55     }
56
57     /**
58      * Method ini berfungsi untuk mengembalikan Option yang sudah diparsing.
59      *
60      * @return Option yang sudah diparsing berupa objek dengan tipe Properties.
61      */
62     public Properties getParsedOptions() {
63         Properties properties = new Properties();
64         for (Option option : this.commandLine.getOptions()) {
65             if (option.getLongOpt().equals("capture-url")) {
66                 String[] values = option.getValues();
67                 String value = values[0];
68
69                 for (int i = 1; i < values.length; i++) {
70                     value = value + ";" + values[i];
71                 }
72                 properties.setProperty(option.getLongOpt(), value);
73             }
74         }
75     }
76 }

```

```

65         } else {
66             properties.setProperty(option.getLongOpt(), option.getValue());
67         }
68     }
69     return properties;
70 }
71 }

```

Listing A.3: Main.java

```

1 package id.ac.unpar.timelapsegenerator;
2
3 import java.io.File;
4 import java.io.IOException;
5 import java.net.HttpURLConnection;
6 import java.net.URL;
7 import java.util.Properties;
8 import javax.imageio.ImageIO;
9 import org.eclipse.jgit.api.errors.GitAPIException;
10
11 /**
12  * @author Billy Adiwijaya
13  */
14 public class Main {
15
16     public static void main(String[] args) {
17         CommandLineOptions commandLineOptions = null;
18         try {
19             commandLineOptions = new CommandLineOptions(args);
20         } catch (Exception e) {
21             System.out.println("Animasi timelapse gagal dibuat");
22             System.out.println(e.getMessage());
23             System.exit(0);
24         }
25         Properties properties = commandLineOptions.getParsedOptions();
26         int numberOfBrowsers = properties.getProperty("capture-url").split(";").length;
27
28         VCS vcs = null;
29         try {
30             vcs = new VCS(properties.getProperty("project-path"));
31
32             if (Double.parseDouble(properties.getProperty("seconds-per-commit")) <= 0) {
33                 throw new Exception("Seconds per commit harus lebih besar dari 0");
34             }
35
36             String[] captureURLs = properties.getProperty("capture-url").split(";");
37             for (String captureURL : captureURLs) {
38                 URL url = new URL(captureURL);
39                 HttpURLConnection http = (HttpURLConnection) url.openConnection();
40                 if (http.getResponseCode() == 404) {
41                     throw new Exception("Capture url tidak valid");
42                 } else if (http.getResponseCode() == 500) {
43                     throw new Exception("Server error");
44                 }
45             }
46
47             if (properties.getProperty("logo") != null) {
48                 File file = new File(properties.getProperty("logo"));
49                 if (!file.exists() || ImageIO.read(new File(properties.getProperty("logo"))) == null) {
50                     throw new Exception("Path gambar tidak valid");
51                 }
52             }
53
54             if (properties.getProperty("start-commit") != null) {
55                 if (properties.getProperty("start-commit").length() != 7) {
56                     throw new Exception("Panjang commit ID awal harus 7 karakter");
57                 }
58                 if (vcs.getCommitIndex(properties.getProperty("start-commit")) == -1) {
59                     throw new Exception("Commit ID awal tidak ditemukan");
60                 }
61             }
62
63             if (properties.getProperty("stop-commit") != null) {
64                 if (properties.getProperty("stop-commit").length() != 7) {
65                     throw new Exception("Panjang commit ID akhir harus 7 karakter");
66                 }
67                 if (vcs.getCommitIndex(properties.getProperty("stop-commit")) == -1) {
68                     throw new Exception("Commit ID akhir tidak ditemukan");
69                 }
70             }
71
72             if (properties.getProperty("start-commit") != null && properties.getProperty("stop-commit") != null) {
73                 if (vcs.getCommitIndex(properties.getProperty("start-commit")) > vcs.getCommitIndex(properties.getProperty("stop-commit"))) {
74                     throw new Exception("Commit ID awal dan akhir terbalik");
75                 } else if (vcs.getCommitIndex(properties.getProperty("start-commit")) == vcs.getCommitIndex(properties.getProperty("stop-commit"))) {
76                     throw new Exception("Commit ID awal dan akhir tidak boleh sama");
77                 }
78             }
79         } catch (NumberFormatException e) {
80             System.out.println("Animasi timelapse gagal dibuat");
81             System.out.println("Seconds per commit harus berupa bilangan riil atau bilangan bulat");
82             System.exit(0);
83         } catch (Exception e) {
84             System.out.println("Animasi timelapse gagal dibuat");
85             System.out.println(e.getMessage());
86             System.exit(0);
87         }
88     }
89 }

```

```

87     }
88
89     BrowserController browserController = new BrowserController(numberOfBrowsers);
90     TimeLapseGenerator timeLapseGenerator = new TimeLapseGenerator();
91     try {
92         timeLapseGenerator.generateTimelapse(properties, vcs, browserController);
93     } catch (IOException | InterruptedException | GitAPIException e) {
94         System.out.println("Animasi timelapse gagal dibuat");
95
96         if (e.getClass().getCanonicalName().equals("java.io.IOException")) {
97             System.out.println("Terminal Command tidak valid");
98         } else {
99             System.out.println(e.getMessage());
100         }
101         System.exit(0);
102     }
103     System.out.println("Animasi timelapse berhasil dibuat");
104 }
105 }
106 }

```

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(10, TimeUnit.SECONDS)) throw new IOException();
66             }
67
68             for (int j = 0; j < captureURL.length; j++) {
69                 browserController.changePage(j, captureURL[j]);
70                 browserController.takeScreenshot(j);
71             }
72
73             vcs.hardReset();
74         }
75         browserController.quit();
76     }
77 }

```



```

75 | vcs.checkoutMaster();
76 |
77 | List<File> screenshotFiles = new ArrayList<>();
78 |
79 | screenshotFiles = browserController.getScreenshotFiles();
80 | BufferedImage[] bufferedImages = new BufferedImage[screenshotFiles.size()];
81 |
82 | for (int i = 0; i < screenshotFiles.size(); i++) {
83 |     bufferedImages[i] = ImageIO.read(screenshotFiles.get(i));
84 | }
85 |
86 | BufferedImage[] bufferedResultImages = new BufferedImage[screenshotFiles.size() / browserController.getNumberOfBrowser()];
87 | switch (browserController.getNumberOfBrowser()) {
88 |     case 1:
89 |         bufferedResultImages = bufferedImages;
90 |         break;
91 |     case 2:
92 |         for (int i = 0, j = 0; i < bufferedResultImages.length; i++, j += 2) {
93 |             bufferedResultImages[i] = new BufferedImage(bufferedImages[0].getWidth(), bufferedImages[0].getHeight(),
94 |                 BufferedImage.TYPE_INT_RGB);
95 |             Graphics2D graphics = bufferedResultImages[i].createGraphics();
96 |             graphics.setBackground(Color.WHITE);
97 |             graphics.fillRect(0, 0, bufferedImages[j].getWidth(), bufferedImages[j].getHeight());
98 |             graphics.drawImage(bufferedImages[j], bufferedImages[j].getWidth() / 4, 0, bufferedImages[j].getWidth() / 2,
99 |                 bufferedImages[j].getHeight() / 2, null);
100 |             graphics.drawImage(bufferedImages[j + 1], bufferedImages[j].getWidth() / 4, bufferedImages[j].getHeight() / 2,
101 |                 bufferedImages[j].getWidth() / 2, bufferedImages[j].getHeight() / 2, null);
102 |             graphics.dispose();
103 |         }
104 |         break;
105 |     case 3:
106 |         for (int i = 0, j = 0; i < bufferedResultImages.length; i++, j += 3) {
107 |             bufferedResultImages[i] = new BufferedImage(bufferedImages[0].getWidth(), bufferedImages[0].getHeight(),
108 |                 BufferedImage.TYPE_INT_RGB);
109 |             Graphics2D graphics = bufferedResultImages[i].createGraphics();
110 |             graphics.setBackground(Color.WHITE);
111 |             graphics.fillRect(0, 0, bufferedImages[j].getWidth(), bufferedImages[j].getHeight());
112 |             graphics.drawImage(bufferedImages[j], bufferedImages[j].getWidth() / 4, 0, bufferedImages[j].getWidth() / 2,
113 |                 bufferedImages[j].getHeight() / 2, null);
114 |             graphics.drawImage(bufferedImages[j + 1], 0, bufferedImages[j].getHeight() / 2, bufferedImages[j].getWidth() /
115 |                 2, bufferedImages[j].getHeight() / 2, null);
116 |             graphics.drawImage(bufferedImages[j + 2], bufferedImages[j].getWidth() / 2, bufferedImages[j].getHeight() / 2,
117 |                 bufferedImages[j].getWidth() / 2, bufferedImages[j].getHeight() / 2, null);
118 |             graphics.dispose();
119 |         }
120 |         break;
121 |     case 4:
122 |         for (int i = 0, j = 0; i < bufferedResultImages.length; i++, j += 4) {
123 |             bufferedResultImages[i] = new BufferedImage(bufferedImages[0].getWidth(), bufferedImages[0].getHeight(),
124 |                 BufferedImage.TYPE_INT_RGB);
125 |             Graphics2D graphics = bufferedResultImages[i].createGraphics();
126 |             graphics.setBackground(Color.WHITE);
127 |             graphics.fillRect(0, 0, bufferedImages[j].getWidth(), bufferedImages[j].getHeight());
128 |             graphics.drawImage(bufferedImages[j], 0, 0, bufferedImages[j].getWidth() / 2, bufferedImages[j].getHeight() /
129 |                 2, null);
130 |             graphics.drawImage(bufferedImages[j + 1], bufferedImages[j].getWidth() / 2, 0, bufferedImages[j].getWidth() /
131 |                 2, bufferedImages[j].getHeight() / 2, null);
132 |             graphics.drawImage(bufferedImages[j + 2], 0, bufferedImages[j].getHeight() / 2, bufferedImages[j].getWidth() /
133 |                 2, bufferedImages[j].getHeight() / 2, null);
134 |             graphics.drawImage(bufferedImages[j + 3], bufferedImages[j].getWidth() / 2, bufferedImages[j].getHeight() / 2,
135 |                 bufferedImages[j].getWidth() / 2, bufferedImages[j].getHeight() / 2, null);
136 |             graphics.dispose();
137 |         }
138 |         break;
139 |     default:
140 |         break;
141 | }
142 |
143 | for (BufferedImage bufferedResultImage : bufferedResultImages) {
144 |     if (properties.getProperty("title") == null && properties.getProperty("logo") == null) {
145 |         break;
146 |     } else {
147 |         if (properties.getProperty("title") != null) {
148 |             Graphics2D graphic = (Graphics2D) bufferedResultImage.getGraphics();
149 |             graphic.setFont(new Font("Times New Roman", Font.BOLD, 18));
150 |             FontMetrics fontMetrics = graphic.getFontMetrics(graphic.getFont());
151 |             graphic.setColor(Color.black);
152 |             graphic.drawString(properties.getProperty("title"), 5, bufferedResultImage.getHeight() - fontMetrics.
153 |                 getDescent() - 5);
154 |             graphic.dispose();
155 |         }
156 |         if (properties.getProperty("logo") != null) {
157 |             BufferedImage logo = ImageIO.read(new File(properties.getProperty("logo")));
158 |             Graphics2D graphic = (Graphics2D) bufferedResultImage.getGraphics();
159 |             graphic.drawImage(logo, bufferedResultImage.getWidth() - logo.getWidth() - 5, bufferedResultImage.getHeight()
160 |                 - logo.getHeight() - 5, null);
161 |             graphic.dispose();
162 |         }
163 |     }
164 | }
165 |
166 | String fileName = String.format("%s.gif", new SimpleDateFormat("yyyy-MM-dd.HH.mm.ss").format(new Date()));
167 | try (ImageOutputStream output = new FileImageOutputStream(new File(fileName))) {
168 |     int frameDelay = (int) (Double.parseDouble(properties.getProperty("seconds-per-commit")) * 1000);
169 |
170 |     GifSequenceWriter writer = new GifSequenceWriter(output, bufferedResultImages[0].getType(), frameDelay, false);
171 |     for (BufferedImage bufferedResultImage : bufferedResultImages) {

```

```

160         writer.writeToSequence(bufferedResultImage);
161     }
162     writer.close();
163 }
164 }
165 }
166 }

```

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.ResetCommand.ResetType;
9 import org.eclipse.jgit.api.errors.GitAPIException;
10 import org.eclipse.jgit.internal.storage.file.FileRepository;
11 import org.eclipse.jgit.lib.Repository;
12 import org.eclipse.jgit.revwalk.RevCommit;
13
14 /**
15  * Kelas ini berfungsi untuk berinteraksi pada proyek perangkat lunak berbasis
16  * web yang terekam oleh Git.
17  *
18  * @author Billy Adiwijaya
19  */
20 public class VCS {
21
22     private final Git git;
23     private final List<String> commitIDs;
24
25     /**
26      * Constructor dari kelas ini. Berfungsi untuk menginisialisasi variabel git
27      * dan mendapatkan seluruh histori commit pada proyek perangkat lunak
28      * berbasis web.
29      *
30      * @param path merupakan path dari proyek perangkat lunak berbasis web.
31      * @throws IOException jika path proyek tidak valid atau repository tidak bisa diakses.
32      * @throws GitAPIException jika terjadi masalah saat melakukan operasi Git Log.
33      */
34     public VCS(String path) throws IOException, GitAPIException {
35         Repository repository = new FileRepository(path);
36         if(repository.getRef("HEAD")==null){
37             throw new IOException("Path proyek tidak valid");
38         }
39         this.git = new Git(repository);
40         Iterable<RevCommit> commits=git.log().call();
41         this.commitIDs = new ArrayList<>();
42         for (RevCommit commit : commits) {
43             this.commitIDs.add(commit.getName().substring(0, 7));
44         }
45         Collections.reverse(commitIDs);
46     }
47
48     /**
49      * Berfungsi untuk melakukan checkout ke commit tertentu.
50      *
51      * @param commitIndex indeks dari variabel commitIDs.
52      * @throws GitAPIException jika terjadi masalah saat melakukan operasi Git Checkout.
53      */
54     public void checkoutCommit(int commitIndex) throws GitAPIException {
55         this.git.checkout().setName(this.commitIDs.get(commitIndex)).call();
56     }
57
58     /**
59      * Berfungsi untuk melakukan checkout ke commit terakhir.
60      *
61      * @throws GitAPIException jika terjadi masalah saat melakukan operasi Git Checkout.
62      */
63     public void checkoutMaster() throws GitAPIException {
64         this.git.checkout().setName("master").call();
65     }
66
67     /**
68      * Berfungsi untuk melakukan operasi Git Reset. Operasi ini menghapus
69      * perubahan pada working tree dan staging area di commit tertentu.
70      *
71      * @throws GitAPIException jika terjadi masalah saat melakukan operasi Git Reset.
72      */
73     public void hardReset() throws GitAPIException {
74         this.git.reset().setMode(ResetType.HARD).call();
75     }
76
77     /**
78      * Berfungsi untuk mendapatkan jumlah commit.
79      *
80      * @return jumlah commit.
81      */
82     public int getNumberOfCommit() {
83         return this.commitIDs.size();
84     }
85
86     /**
87      * Berfungsi untuk mendapatkan index dari variabel commitID.
88      *

```

```
89 |      * @param commitID merupakan Commit ID yang akan dicari indeks nya.  
90 |      * @return indeks dari variabel commitIDs.  
91 |      */  
92 |      public int getCommitIndex(String commitID) {  
93 |          int result = -1;  
94 |          for (int i = 0; i < this.commitIDs.size(); i++) {  
95 |              if (commitID.equals(this.commitIDs.get(i))) {  
96 |                  result = i;  
97 |                  break;  
98 |              }  
99 |          }  
100 |          return result;  
101 |      }  
102 | }
```


LAMPIRAN B

LIBRARY UNTUK MEMBUAT FILE GIF

Listing B.1: GifSequenceWriter.java

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

Listing C.1: auth.php

```
1 <?php
2
3 defined('BASEPATH') OR exit('No direct script access allowed');
4
5 $config['domain'] = 'http://localhost';
6 $config['google-clientid'] = '9911963967-e4apbt2ksb0492p59rtoc8qhcenth6a4e.apps.googleusercontent.com';
7 $config['google-clientsecret'] = 'tw-7N1uNuqdCPRkXRFvqz7xR';
8 $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 |
62 | This option allows you to add a suffix to all URLs generated by CodeIgniter.
63 | For more information please see the user guide:
```

```

64 |
65 | https://codeigniter.com/user_guide/general/urls.html
66 | */
67 | $config['url_suffix'] = '';
68 |
69 | /*
70 | -----
71 | Default Language
72 | -----
73 |
74 | This determines which set of language files should be used. Make sure
75 | there is an available translation if you intend to use something other
76 | than english.
77 |
78 | */
79 | $config['language'] = 'english';
80 |
81 | /*
82 | -----
83 | Default Character Set
84 | -----
85 |
86 | This determines which character set is used by default in various methods
87 | that require a character set to be provided.
88 |
89 | See http://php.net/htmlspecialchars for a list of supported charsets.
90 |
91 | */
92 | $config['charset'] = 'UTF-8';
93 |
94 | /*
95 | -----
96 | Enable/Disable System Hooks
97 | -----
98 |
99 | If you would like to use the 'hooks' feature you must enable it by
100 | setting this variable to TRUE (boolean). See the user guide for details.
101 |
102 | */
103 | $config['enable_hooks'] = FALSE;
104 |
105 | /*
106 | -----
107 | Class Extension Prefix
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 | -----
121 | Composer auto-loading
122 | -----
123 |
124 | Enabling this setting will tell CodeIgniter to look for a Composer
125 | package auto-loader script in application/vendor/autoload.php.
126 |
127 | $config['composer_autoload'] = TRUE;
128 |
129 | Or if you have your vendor/ directory located somewhere else, you
130 | can opt to set a specific path as well:
131 |
132 | $config['composer_autoload'] = '/path/to/vendor/autoload.php';
133 |
134 | For more information about Composer, please visit http://getcomposer.org/
135 |
136 | Note: This will NOT disable or override the CodeIgniter-specific
137 | autoloading (application/config/autoload.php)
138 | */
139 | $config['composer_autoload'] = '../vendor/autoload.php';
140 |
141 | /*
142 | -----
143 | Allowed URL Characters
144 | -----
145 |
146 | This lets you specify which characters are permitted within your URLs.
147 | When someone tries to submit a URL with disallowed characters they will
148 | get a warning message.
149 |
150 | As a security measure you are STRONGLY encouraged to restrict URLs to
151 | as few characters as possible. By default only these are allowed: a-z 0-9~%.:_-
152 |
153 | Leave blank to allow all characters -- but only if you are insane.
154 |
155 | The configured value is actually a regular expression character group
156 | and it will be executed as: ! preg_match('/^[<permitted_uri_chars>]+$'/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 | -----
165 | Enable Query Strings
166 | -----
167 |
168 | By default CodeIgniter uses search-engine friendly segment based URLs:
169 | example.com/who/what/where/
170 |
171 | By default CodeIgniter enables access to the $.GET array. If for some
172 | reason you would like to disable it, set 'allow_get_array' to FALSE.
173 |
174 | You can optionally enable standard query string based URLs:
175 | example.com?who=me&what=something&where=here
176 |
177 | Options are: TRUE or FALSE (boolean)
178 |
179 | The other items let you set the query string 'words' that will
180 | invoke your controllers and its functions:
181 | example.com/index.php?c=controller&m=function
182 |
183 | Please note that some of the helpers won't work as expected when
184 | this feature is enabled, since CodeIgniter is designed primarily to
185 | use segment based URLs.
186 |
187 | */
188 | $config['allow_get_array'] = TRUE;
189 | $config['enable_query_strings'] = FALSE;
190 | $config['controller_trigger'] = 'c';
191 | $config['function_trigger'] = 'm';
192 | $config['directory_trigger'] = 'd';
193 |
194 | /*
195 | -----
196 | Error Logging Threshold
197 | -----
198 |
199 | You can enable error logging by setting a threshold over zero. The
200 | threshold determines what gets logged. Threshold options are:
201 |
202 | 0 = Disables logging, Error logging TURNED OFF
203 | 1 = Error Messages (including PHP errors)
204 | 2 = Debug Messages
205 | 3 = Informational Messages
206 | 4 = All Messages
207 |
208 | You can also pass an array with threshold levels to show individual error types
209 |
210 | array(2) = Debug Messages, without Error Messages
211 |
212 | For a live site you'll usually only enable Errors (1) to be logged otherwise
213 | your log files will fill up very fast.
214 |
215 | */
216 | $config['log_threshold'] = 0;
217 |
218 | /*
219 | -----
220 | Error Logging Directory Path
221 | -----
222 |
223 | Leave this BLANK unless you would like to set something other than the default
224 | application/logs/ directory. Use a full server path with trailing slash.
225 |
226 | */
227 | $config['log_path'] = '';
228 |
229 | /*
230 | -----
231 | Log File Extension
232 | -----
233 |
234 | The default filename extension for log files. The default 'php' allows for
235 | protecting the log files via basic scripting, when they are to be stored
236 | under a publicly accessible directory.
237 |
238 | Note: Leaving it blank will default to 'php'.
239 |
240 | */
241 | $config['log_file_extension'] = '';
242 |
243 | /*
244 | -----
245 | Log File Permissions
246 | -----
247 |
248 | The file system permissions to be applied on newly created log files.
249 |
250 | IMPORTANT: This MUST be an integer (no quotes) and you MUST use octal
251 | integer notation (i.e. 0700, 0644, etc.)
252 |
253 | */
254 | $config['log_file_permissions'] = 0644;
255 |
256 | /*
257 | -----
258 | Date Format for Logs
259 | -----
260 |
261 | Each item that is logged has an associated date. You can use PHP date

```

```

262 |
263 */
264 $config['log_date_format'] = 'Y-m-d H:i:s';
265
266 /*
267 |-----
268 | Error Views Directory Path
269 |-----
270 |
271 | Leave this BLANK unless you would like to set something other than the default
272 | application/views/errors/ directory. Use a full server path with trailing slash.
273 |
274 */
275 $config['error_views_path'] = '';
276
277 /*
278 |-----
279 | Cache Directory Path
280 |-----
281 |
282 | Leave this BLANK unless you would like to set something other than the default
283 | application/cache/ directory. Use a full server path with trailing slash.
284 |
285 */
286 $config['cache_path'] = '';
287
288 /*
289 |-----
290 | Cache Include Query String
291 |-----
292 |
293 | Whether to take the URL query string into consideration when generating
294 | output cache files. Valid options are:
295 |
296 | FALSE      = Disabled
297 | TRUE       = Enabled, take all query parameters into account.
298 |              Please be aware that this may result in numerous cache
299 |              files generated for the same page over and over again.
300 | array('q') = Enabled, but only take into account the specified list
301 |              of query parameters.
302 |
303 */
304 $config['cache_query_string'] = FALSE;
305
306 /*
307 |-----
308 | Encryption Key
309 |-----
310 |
311 | If you use the Encryption class, you must set an encryption key.
312 | See the user guide for more info.
313 |
314 | https://codeigniter.com/user\_guide/libraries/encryption.html
315 |
316 */
317 $config['encryption_key'] = '';
318
319 /*
320 |-----
321 | Session Variables
322 |-----
323 |
324 | 'sess_driver'
325 |
326 | The storage driver to use: files, database, redis, memcached
327 |
328 | 'sess_cookie_name'
329 |
330 | The session cookie name, must contain only [0-9a-z_-] characters
331 |
332 | 'sess_expiration'
333 |
334 | The number of SECONDS you want the session to last.
335 | Setting to 0 (zero) means expire when the browser is closed.
336 |
337 | 'sess_save_path'
338 |
339 | The location to save sessions to, driver dependent.
340 |
341 | For the 'files' driver, it's a path to a writable directory.
342 | WARNING: Only absolute paths are supported!
343 |
344 | For the 'database' driver, it's a table name.
345 | Please read up the manual for the format with other session drivers.
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 | -----
380 | Cookie Related Variables
381 | -----
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 | -----
401 | Standardize newlines
402 | -----
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 | -----
415 | Global XSS Filtering
416 | -----
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 | -----
429 | Cross Site Request Forgery
430 | -----
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 | */
443 | $config['csrf_protection'] = FALSE;
444 | $config['csrf_token_name'] = 'csrf_test_name';
445 | $config['csrf_cookie_name'] = 'csrf_cookie_name';
446 | $config['csrf_expire'] = 7200;
447 | $config['csrf_regenerate'] = TRUE;
448 | $config['csrf_exclude_uris'] = array();
449 |
450 | /*
451 | -----
452 | Output Compression
453 | -----
454 |
455 | Enables Gzip output compression for faster page loads. When enabled,
456 | the output class will test whether your server supports Gzip.
457 | Even if it does, however, not all browsers support compression
458 | so enable only if you are reasonably sure your visitors can handle it.
459 |
460 | Only used if zlib.output_compression is turned off in your php.ini.
461 | Please do not use it together with httpd-level output compression.

```

```

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 |$config['compress_output'] = FALSE;
469 |
470 |/*
471 |-----
472 | Master Time Reference
473 |-----
474 |
475 | Options are 'local' or any PHP supported timezone. This preference tells
476 | the system whether to use your server's local time as the master 'now'
477 | reference, or convert it to the configured one timezone. See the 'date
478 | helper' page of the user guide for information regarding date handling.
479 |
480 |*/
481 |$config['time_reference'] = 'local';
482 |
483 |/*
484 |-----
485 | Rewrite PHP Short Tags
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 |$config['rewrite_short_tags'] = FALSE;
496 |
497 |/*
498 |-----
499 | Reverse Proxy IPs
500 |-----
501 |
502 | If your server is behind a reverse proxy, you must whitelist the proxy
503 | IP addresses from which CodeIgniter should trust headers such as
504 | HTTP_X_FORWARDED_FOR and HTTP_CLIENT_IP in order to properly identify
505 | the visitor's IP address.
506 |
507 | You can use both an array or a comma-separated list of proxy addresses,
508 | as well as specifying whole subnets. Here are a few examples:
509 |
510 | Comma-separated: '10.0.1.200,192.168.5.0/24'
511 | Array:           array('10.0.1.200', '192.168.5.0/24')
512 |*/
513 |$config['proxy_ips'] = '';

```

Listing C.3: database.php

```

1  <?php
2  defined('BASEPATH') OR exit('No direct script access allowed');
3
4  /*
5  |-----
6  | DATABASE CONNECTIVITY SETTINGS
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 |
24 |           Currently supported:
25 |           cubrid, ibase, mssql, mysql, mysqli, oci8,
26 |           odbc, pdo, postgre, sqlite, sqlite3, sqlsrv
27 | ['dbprefix']      You can add an optional prefix, which will be added
28 |                   to the table name when using the Query Builder class
29 | ['pconnect']      TRUE/FALSE - Whether to use a persistent connection
30 | ['db_debug']      TRUE/FALSE - Whether database errors should be displayed.
31 | ['cache_on']      TRUE/FALSE - Enables/disables query caching
32 | ['cachedir']      The path to the folder where cache files should be stored
33 | ['char_set']      The character set used in communicating with the database
34 | ['dbcollat']      The character collation used in communicating with the database
35 |                   NOTE: For MySQL and MySQLi databases, this setting is only used
36 |                   as a backup if your server is running PHP < 5.2.3 or MySQL < 5.0.7
37 |                   (and in table creation queries made with DB Forge).
38 |                   There is an incompatibility in PHP with mysql_real_escape_string() which
39 |                   can make your site vulnerable to SQL injection if you are using a
40 |                   multi-byte character set and are running versions lower than these.
41 |                   Sites using Latin-1 or UTF-8 database character set and collation are unaffected.
42 | ['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/mysqli' 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 variable 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 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 class Migrate extends CI_Controller {
5
6     /**
7      * Jika dibuka di http://localhost/migrate akan mengupdate ke db terbaru.
8      * Jika ada data sendiri, mohon dibackup dulu.
9      */
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 GuzzleHttp\Client(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     }
20 }

```

```

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 }
26 }

```

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

```

```

        companies ranging from start-up level to the prominent areas. With solid team of graphic designers,
        photographers, and web developers, we guarantee a creativity, originality, and identity to your brand.</p>
158     </div>
159     <div class="special-description">
160         <h4>We Do<br>Something Fun<br>To Make<br>Something Special</h4>
161     </div>
162     <div class="content-navigation content-next">
163         <p>WHAT WE DO </p>
164     </div>
165 </div>
166 <div class="content-2">
167     <div class="container-fluid">
168         <div class="row">
169             <div class="col-xs-12 col-sm-6 col-md-3">
170                 <h5>BRANDING</h5>
171                 <p class="description">It all started with a discovery session where we will chat about your vision
                    and expectation for your creative needs. Just mention and describe what you have in mind, or we
                    can even brainstorm and gather ideas together; then we'll explore them from scratches. From
                    there, leave the rest to us, we'll translate your creative ideas into a workable concept with a
                    strong impression.</p>
172             </div>
173             <div class="col-xs-12 col-sm-6 col-md-3">
174                 <h5>GRAPHIC DESIGN</h5>
175                 <p class="description">Let us know what your target market is. We can design every single marketing
                    material that you need. We'll act as your artwork consultant to narrow down your needs in order
                    to achieve your goals. We always take a closer look in every detail, to make sure that the
                    appearance of your brand looks consistent and meaningful.</p>
176             </div>
177             <div class="col-xs-12 col-sm-6 col-md-3">
178                 <h5>WEBSITE DESIGN</h5>
179                 <p class="description">We believe that a good design comes with a good user experience on how people
                    access a brand in the virtual world. Piktora specializes in developing a brand, while also
                    collaborating with the current trend. Remember, a website is not only a tool that tells your
                    audience about what your brand does, but also creates an impressive display in today's modern
                    industry, as well as an affective user experience from the very beginning.</p>
180             </div>
181         </div>
182     </div>
183     <div class="content-navigation content-prev">
184         <p> BACK TO OUR STORY</p>
185     </div>
186 </div>
187 </div>
188 </div>
189
190 <?php $this->load->view('templates/scripts', array('extra_js' => array('slick.min'))); ?>
191 <script>
192     $(document).ready(function(){
193         $(".content").slick({
194             swipe : false,
195             arrows: false
196         });
197         $(".content-next").click(function(event) {
198             $(".content").slick("slickNext");
199         });
200         $(".content-prev").click(function(event) {
201             $(".content").slick("slickPrev");
202         });
203     });
204 </script>
205 </body>
206 </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:#eeeeee9;
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">Don't 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="

```

```

132         required"></textarea>
133     </div>
134     <div class="form-group captcha-container">
135         <div class="col-xs-6 col-md-3 captcha-code">
136             <?= $captcha; ?>
137         </div>
138         <div class="col-xs-6 col-md-3 captcha-input">
139             <input type="text" name="captcha" class="form-control" placeholder="Enter the code" required="required"
140                 ">
141         </div>
142         <div class="col-xs-6 col-md-3 col-md-offset-3">
143             <button type="submit" class="btn btn-default send-button">Send Now </button>
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                         <h5><?= $project['title'] ?></h5>
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:#ffffffa;
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 </div>
132 </div>
133 <?php $this->load->view('templates/scripts'); ?>
134 <script>
135     $(document).ready(function() {
136         function setContentHeight(){
137             var height = $(window).height()-$(".navigation").height()-50;
138             $(".container-content-img").css( "height", (height + 'px'));
139             $(".container-content-text").css( "height", (height + 'px'));
140         }
141         if($( window ).width())>=768){
142             setContentHeight();
143         }
144         $( window ).resize(function() {
145             if($( window ).width())>=768){
146                 setContentHeight();
147             }
148         });
149     });
150 </script>
151 </body>
152 </html>

```

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,
                Gading Serpong, Tangerang, Banten 15810.</p>
9         </div>
10        <div class="col-xs-12 col-sm-3 col-md-2">
11            <h5>hello@piktora.com</h5>
12            <p>+62 21 2941 8990</p>
13        </div>
14        <div class="col-xs-12 col-sm-3 col-sm-offset-3 col-md-2 col-md-offset-6">
15            <table class="daily-stuff-container">
16                <tbody>
17                    <tr>
18                        <td class="daily-stuff"><h5>See Our Funside<br>& Daily Stuff</h5></td>
19                        <td class="social-media"><a target="_blank" href="https://www.instagram.com/piktorastudio/"></a></td>
20                    </tr>
21                </tbody>

```

```

22         </table>
23         <p>Copyright &copy; 2017&nbsp;&nbsp;&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"></a></td>
11             <td class="navigation-right"></td>
12         </tr>
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-->

```