

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, 17 Mei 2019

Menyetujui,

Pembimbing

Dr. Veronica Sri Moertini

Ketua Tim Penguji

Anggota Tim Penguji

Kristopher David Harjono, M.T.

Elisati Hulu, M.T.

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 17 Mei 2019

Meterai Rp. 6000

Billy Adiwijaya
NPM: 2015730053

ABSTRAK

Git merupakan perangkat lunak *Version Control Systems*. *Version Control* adalah sistem yang merekam perubahan pada *file* atau sekumpulan *file* dari waktu ke waktu. Perubahan yang terjadi pada repositori dicatat oleh Git dalam bentuk *commit history*. Dengan adanya *commit*, pengembangan suatu proyek perangkat lunak dapat dipantau. Akan tetapi, untuk perangkat lunak yang memiliki banyak *commit*, pemantauan progres dapat memakan waktu yang lama. Oleh karena itu, dibuatlah perangkat lunak yang dapat membangkitkan animasi *timelapse* dari pengembangan proyek perangkat lunak.

Perangkat lunak pembangkit *timelapse* ini hanya berfokus pada pengembangan proyek perangkat lunak berbasis *web*. Animasi *timelapse* dibuat dengan cara menggabungkan *screenshot* halaman *web* pada setiap *commit* menjadi satu *file* bertipe GIF. Penelusuran *commit* dilakukan menggunakan JGit. *Screenshot* halaman *web* pada setiap *commit* diambil menggunakan Selenium WebDriver.

Pengujian dilakukan dengan menggunakan beberapa repositori dari situs web dan WebDriver. Berdasarkan hasil pengujian, program dapat berjalan dengan baik pada beberapa WebDriver. Pada EdgeDriver, tampilan *layout* dari halaman *web* menjadi tidak rapih. Program dapat berjalan dengan baik dan dapat membangkitkan animasi *timelapse* pada beberapa repositori dari situs web. Ukuran *file* hasil animasi bergantung pada banyaknya *commit* pada repositori. Semakin banyak jumlah *commit*, semakin besar ukuran *file*.

Kata-kata kunci: Git, JGit, Selenium WebDriver, *timelapse*, *commit*

ABSTRACT

Git is a Version Control System. Version control is a system that records changes to a file or set of files over time. Changes in a repository are recorded by Git in the form of commit history. Software development can be tracked via commit history. However, it will take time to monitor the progress of software that has many commits. Therefore, software that can generate timelapse animation for software project development is made.

This software only focus on web-based software project development. Timelapse animation can be made by combining screenshots of web page into one GIF file. Commits can be traversed using JGit. Screenshot of web page in every commit can be taken using Selenium WebDriver.

Testing is done by trying to run this software using several websites and WebDriver. The result of software testing shows that software can run well on several WebDriver. In EdgeDriver, layout of the page was broken. Software can run well on several websites. The size of animation file depends on the number of commit in a repository. The more the number of commits made, the bigger the file size of the animation file gets.

Keywords: Git, JGit, Selenium WebDriver, timelapse, commit

Teknik Informatika UNPAR dan diri sendiri

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas seluruh berkat yang diberikan kepada penulis sehingga dapat menyelesaikan tugas akhir dengan judul **Pembangkit Timelapse Pengembangan Proyek Perangkat Lunak Berbasis Web** dengan baik dan tepat waktu. Penulis juga berterima kasih kepada pihak-pihak yang telah memberikan dukungan dan bantuan kepada penulis dalam menyelesaikan tugas akhir ini, yaitu:

1. Keluarga yang selalu memberikan dukungan kepada penulis.
2. Bapak Pascal Alfadian sebagai dosen pembimbing yang telah membimbing penulis hingga dapat menyelesaikan skripsi ini.
3. Bapak Kristopher David Harjono dan Bapak Elisati Hulu sebagai dosen penguji yang telah membantu dalam menguji dan memperbaiki skripsi ini.
4. Teman-teman LKM 2017/2018 yang telah memberikan semangat kepada penulis.
5. Keluarga besar Own Games yang telah memberikan semangat kepada penulis.
6. Pihak-pihak lain yang belum disebutkan, yang berperan dalam penyelesaian tugas akhir ini.

Bandung, Mei 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	2
1.4 Batasan Masalah	2
1.5 Metodologi	2
1.6 Sistematika Pembahasan	3
2 LANDASAN TEORI	5
2.1 Version Control Systems	5
2.1.1 Local Version Control Systems	5
2.1.2 Centralized Version Control Systems	6
2.1.3 Distributed Version Control Systems	7
2.2 Git	7
2.2.1 Cara Kerja Git	7
2.2.2 Operasi Dasar pada Git	10
2.2.3 Git Checkout	12
2.3 JGit	12
2.3.1 Kelas Repository	12
2.3.2 Kelas Ref	13
2.3.3 Kelas FileRepository	13
2.3.4 Kelas Git	13
2.3.5 Kelas CheckoutCommand	15
2.3.6 Kelas ListBranchCommand	15
2.3.7 Enum ListBranchCommand.ListMode	15
2.3.8 Kelas LogCommand	15
2.3.9 Kelas ResetCommand	16
2.3.10 Kelas ResetCommand.ResetType	16
2.3.11 Kelas RevCommit	16
2.3.12 Kelas PersonIdent	17
2.4 Selenium WebDriver	17
2.4.1 Interface WebDriver	18
2.4.2 Interface WebDriver.Options	18
2.4.3 Interface WebDriver.Window	19
2.4.4 Kelas ChromeDriver	19

2.4.5	Kelas FirefoxDriver	19
2.4.6	Kelas OperaDriver	19
2.4.7	Kelas SafariDriver	19
2.4.8	Kelas InternetExplorerDriver	19
2.4.9	Kelas EdgeDriver	19
2.4.10	Kelas HtmlUnitDriver	20
2.4.11	Interface OutputType	20
2.4.12	Interface TakesScreenshot	20
2.5	Apache Commons CLI	20
2.5.1	Interface CommandLineParser	20
2.5.2	Kelas DefaultParser	21
2.5.3	Kelas CommandLine	21
2.5.4	Kelas Options	22
2.5.5	Kelas Option	22
2.5.6	Kelas Option.Builder	22
3	ANALISIS	25
3.1	Analisis Aplikasi Sejenis	25
3.2	Analisis Penggunaan JGit dan Selenium WebDriver	27
3.2.1	Analisis Penggunaan JGit	27
3.2.2	Analisis Penggunaan Selenium WebDriver	28
3.3	Analisis Fitur Aplikasi yang Dibangun	29
3.4	Prapengujian Website Piktora	33
3.4.1	Perbedaan Letak <i>File</i>	36
3.4.2	Permasalahan Konfigurasi <i>Database</i>	36
3.4.3	Permasalahan Migrasi <i>Database</i>	37
4	PERANCANGAN	39
4.1	Perancangan Kelas	39
4.2	Perancangan Antarmuka	42
5	IMPLEMENTASI DAN PENGUJIAN	45
5.1	Implementasi	45
5.1.1	Lingkungan Implementasi	45
5.1.2	Hasil Implementasi	45
5.2	Pengujian	47
5.2.1	Pengujian Fungsional	47
5.2.2	Pengujian Eksperimental	52
6	KESIMPULAN DAN SARAN	63
6.1	Kesimpulan	63
6.2	Saran	63
	DAFTAR REFERENSI	65
	A KODE PROGRAM	67
	B LIBRARY UNTUK MEMBUAT FILE GIF	75
	C KODE PROGRAM PROYEK PIKTORA PADA COMMIT 89000BE (12 JANUARI 2018)	79

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 directory</i> , <i>Staging area</i> , dan Git direktori[1].	9
2.7	<i>Checkout</i> pada <i>commit</i>	12
3.1	Visualisasi dari repositori perangkat lunak menggunakan Gource.	25
3.2	<i>Commit history</i> direpresentasikan sebagai Directed Acyclic Graph.	28
3.3	<i>Browser</i> yang dikontrol oleh ChromeDriver.	29
3.4	<i>Use case diagram</i> perangkat lunak.	31
3.5	Langkah-langkah untuk membangkitkan animasi <i>timelapse</i>	33
4.1	Diagram kelas.	41
4.2	Rancangan tampilan <i>screenshot</i> pada <i>file</i> hasil animasi jika terdapat satu halaman <i>web</i>	43
4.3	Rancangan tampilan <i>screenshot</i> pada <i>file</i> hasil animasi jika terdapat dua halaman <i>web</i>	43
4.4	Rancangan tampilan <i>screenshot</i> pada <i>file</i> hasil animasi jika terdapat tiga halaman <i>web</i>	44
4.5	Rancangan tampilan <i>screenshot</i> pada <i>file</i> hasil animasi jika terdapat empat halaman <i>web</i>	44
5.1	<i>Screenshot</i> halaman <i>web</i> setiap <i>commit</i> pada situs <i>web</i> Piktora.	46
5.2	Salah satu <i>screenshot</i> halaman <i>web</i> yang terdapat pada <i>file</i> hasil animasi. Terdapat judul di bagian pojok kiri bawah.	47
5.3	Salah satu <i>screenshot</i> halaman <i>web</i> yang terdapat pada <i>file</i> hasil animasi. Terdapat logo di bagian pojok kanan bawah.	47
5.4	Salah satu <i>screenshot</i> halaman <i>web</i> pada <i>file</i> hasil animasi jika terdapat satu argumen <code>-capture-url</code>	51
5.5	Salah satu <i>screenshot</i> halaman <i>web</i> hasil animasi jika terdapat dua argumen <code>-capture-url</code>	51
5.6	Salah satu <i>screenshot</i> halaman <i>web</i> hasil animasi jika terdapat tiga argumen <code>-capture-url</code>	51
5.7	Salah satu <i>screenshot</i> halaman <i>web</i> pada <i>file</i> hasil animasi jika terdapat empat argumen <code>-capture-url</code>	52
5.8	Tampilan pada Firefox <i>browser</i> saat dikontrol oleh FirefoxDriver.	53
5.9	Tampilan pada Opera <i>browser</i> saat dikontrol oleh OperaDriver.	54
5.10	Tampilan halaman <i>web</i> pada <i>browser</i> saat dikontrol oleh EdgeDriver.	55
5.11	Tampilan halaman <i>web</i> pada <i>browser</i> saat tidak dikontrol oleh EdgeDriver.	55
5.12	Keadaan awal dari <i>browser</i> saat dikontrol oleh InternetExplorerDriver.	56
5.13	Tampilan pada <i>browser</i> saat dikontrol oleh InternetExplorerDriver.	56
5.14	Sebagian hasil animasi dari situs <i>web</i> Netflix Open Source Software Center.	57
5.15	Sebagian hasil animasi dari situs <i>web</i> Bootstrap.	58

5.16	Sebagian hasil animasi dari situs <i>web</i> IBM Open Source.	59
5.17	Sebagian hasil animasi dari situs <i>web</i> React.	60
5.18	Sebagian hasil animasi dari situs <i>web</i> Yelp.	61

DAFTAR TABEL

3.1	<i>Scenario case</i> membangkitkan animasi <i>timelapse</i>	32
5.1	Tabel pengujian fungsional	48
5.2	Tabel pengujian fungsional	49
5.3	Tabel pengujian fungsional	50
5.4	Tabel hasil pengujian eksperimental pada beberapa situs <i>web</i>	61

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 repositori dicatat oleh Git dalam bentuk *commit history*. Setiap *commit* mengandung informasi mengenai perubahan yang terjadi pada repositori, 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, operasi-operasi 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*, Safari *browser*, Internet Explorer *browser*, dan Microsoft Edge *browser*.

Dengan adanya *commit history* pada Git, pengembangan suatu proyek perangkat lunak dapat dipantau. *Programmer* dapat melihat kembali secara garis besar langkah-langkah yang sudah dilakukan pada *commit* sebelumnya. *Commit history* juga dapat digunakan untuk mengevaluasi pengembangan perangkat lunak. Akan tetapi, untuk perangkat lunak yang memiliki banyak *commit*, pemantauan progres dapat memakan waktu yang lama. Karena itu proses pengembangan perangkat lunak perlu divisualisasikan.

Timelapse berasal dari kata latin *lapsus* yang bermakna "waktu yang berlalu"[4]. Secara umum *timelapse* dalam fotografi menunjukkan waktu yang lebih cepat dibandingkan dengan waktu sebenarnya. *Timelapse* bisa digunakan untuk mengompres waktu. Misalnya pembangunan suatu gedung dalam waktu tiga tahun bisa ditampilkan dalam bentuk *timelapse* yang hanya berdurasi dua menit.

Pada skripsi ini, dibuat sebuah perangkat lunak yang dapat membangun animasi *timelapse* dari pengembangan proyek perangkat lunak berbasis *web*. Yang dibuat animasinya adalah halaman-halaman *web* dari suatu perangkat lunak berbasis *web* yang terekam oleh Git. Perangkat lunak ini dibangun menggunakan bahasa Java. Perangkat lunak ini menggunakan antarmuka terminal/konsol. Masukan perangkat lunak diambil dari argumen *command line*. Perangkat lunak dibuat menggunakan bantuan *library* JGit dan Selenium WebDriver. *Output* yang dihasilkan dari program ini adalah *file* hasil animasi yang bertipe GIF.

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 membangun aplikasi untuk membangkitkan *timelapse* pada pengembangan proyek perangkat lunak berbasis web?
3. Kesimpulan apa yang dapat diambil dari animasi *timelapse* pada beberapa situs *web* Open Source?

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. Membangun aplikasi untuk membangkitkan *timelapse* pada pengembangan proyek perangkat lunak berbasis web.
3. Mendapatkan kesimpulan dari animasi *timelapse* pada beberapa situs *web* Open Source.

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* yang repositorinya terekam oleh Git.
2. Jumlah halaman *web* pada konfigurasi maksimal berjumlah empat halaman.
3. *Setup* perangkat lunak berbasis *web* yang dibuat animasinya harus bisa melalui *terminal* (tanpa GUI).

1.5 Metodologi

Metodologi penelitian yang digunakan dalam skripsi ini adalah sebagai berikut:

1. Melakukan studi literatur tentang Git, JGit, Selenium WebDriver, dan Apache Commons CLI.
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 Version Control Systems, 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*, analisis fitur aplikasi yang dibangun, dan prapengujian.
4. Bab 4: Perancangan, yaitu membahas mengenai perancangan yang dilakukan sebelum melakukan tahapan implementasi. Berisi tentang perancangan kelas dan perancangan antarmuka dari perangkat lunak.
5. Bab 5: Implementasi dan Pengujian, yaitu membahas mengenai implementasi dan pengujian aplikasi yang telah dilakukan. Berisi tentang implementasi dan hasil pengujian perangkat lunak.
6. Bab 6: Kesimpulan dan Saran, yaitu membahas hasil kesimpulan dari keseluruhan penelitian ini dan saran-saran yang dapat diberikan untuk penelitian berikutnya.