

TUGAS AKHIR

SISTEM INFORMASI RIWAYAT MAHASISWA



SAMUEL HERMAN

NPM: 2010730013

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2015**

FINAL PROJECT

STUDENT INFORMATION HISTORY RECORDING SYSTEM



SAMUEL HERMAN

NPM: 2010730013

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2015**

ABSTRAK

Penelitian ini akan membangun perangkat lunak Sistem Informasi Riwayat Mahasiswa (SIRM). SIRM dibangun dengan tujuan untuk membantu para dosen dalam mengingat riwayat mahasiswa. SIRM dibangun menggunakan empat teknologi, yaitu Google Authentication, Markdown, StrapdownJS, dan Zurb Foundation. Google Authentication akan digunakan untuk membangun fungsi login yang akan membatasi lingkup pengguna. Markdown akan digunakan dalam format penulisan riwayat mahasiswa. StrapdownJS akan digunakan untuk menampilkan teks yang ditulis menggunakan sintaks Markdown ke HTML. Zurb Foundation akan digunakan untuk membuat User Interface (UI) SIRM.

Penelitian ini akan mengembangkan CAPTCHA yang memiliki kualitas yang baik. Kualitas yang baik akan didapat apabila CAPTCHA yang dikembangkan dapat diselesaikan oleh manusia dan tidak dapat diselesaikan oleh komputer. Gambar CAPTCHA yang dibangkitkan akan menggunakan tiga efek distorsi untuk meningkatkan kualitas CAPTCHA. Efek distorsi pertama adalah merotasi karakter yang akan diacak dari 0° sampai 360° . Efek distorsi kedua adalah membalik karakter secara horisontal maupun vertikal. Dan efek distorsi yang terakhir adalah menambahkan garis yang ditambahkan pada gambar secara acak. Ketiga efek distorsi tersebut akan digabungkan sehingga menghasilkan CAPTCHA dengan kualitas yang baik.

Kata-kata kunci: Google Authentication, Markdown, StrapdownJS, Zurb Foundation

ABSTRACT

This research will develop a CAPTCHA that has good quality. Good quality will be obtained if the CAPTCHA which is developed can be solved by humans and cannot be solved by a computer. A picture captcha raised will use of three effects of distortion to improve the quality of captcha. The first distortion effects is a rotation of characters that will be scrambled from 0° to 360° . The second distortion effects to flip characters to the horizontal as well as vertical. And the effects of distortion that last one is add a line that is added in figure at random. The third effect of the distortion will be combined so as to generate a CAPTCHA with a good quality.

Keywords: Google Authentication, Markdown, StrapdownJS, Zurb Foundation

DAFTAR ISI

DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiv
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 Penelitian	2
1.6 Sistematika Pembahasan	3
2 LANDASAN TEORI	5
2.1 Google Authentication [1]	5
2.1.1 Langkah Dasar	5
2.1.2 Skenario Google Authentication	6
2.1.3 Masa Habis Berlaku Token	11
2.1.4 Lingkup Otorisasi [2]	11
2.2 Markdown	14
2.2.1 Apa itu Markdown? [3]	14
2.2.2 Sintaks yang Berguna [3]	14
2.2.3 GitHub Flavored Markdown [4]	20
2.3 StrapdownJS [5]	23
2.4 Zurb Foundation [6]	24
2.4.1 Kompatibilitas	25
2.4.2 Apa Saja yang Dapat Dibuak Dengan Foundation?	25
3 ANALISIS	35
3.1 Analisis Google Authentication	35
3.1.1 Langkah Dasar Penggunaan OAuth 2.0	35
3.1.2 Skenario Aplikasi	40
3.2 Analisis Markdown	41
3.3 Analisis StrapdownJS	44
3.4 Analisis Zurb Foundation	45
3.5 Analisis Kebutuhan Perangkat Lunak	46
3.5.1 Use Case Diagram	46
3.5.2 Skenario	47
3.5.3 Entity Relationship Diagram (ERD)	50
3.5.4 Data Flow Diagram	50
4 PERANCANGAN	53

4.1	Perancangan Tampilan Web Yang Digunakan	53
4.1.1	Tampilan Halaman Awal	53
4.1.2	Tampilan Web Pilih Mahasiswa	53
4.1.3	Tampilan Web Info Mahasiswa	54
4.1.4	Tampilan Web Edit Mahasiswa	54
4.1.5	Tampilan Web Lihat Histori	55
4.1.6	Tampilan Web Lihat Versi Ini	55
4.1.7	Tampilan Web Entri Baru	56
4.2	Perancangan Modul	56
4.2.1	Modul Login	57
4.2.2	Modul Pilih Mahasiswa	57
4.2.3	Modul Info Mahasiswa	57
4.2.4	Modul Edit Mahasiswa	57
4.2.5	Modul Lihat Histori	57
4.2.6	Modul Entri Baru	58
4.3	Perancangan Tabel Sistem Informasi Riwayat Mahasiswa	58
4.3.1	Perancangan Tabel Info Mahasiswa	58
4.3.2	Perancangan Tabel Histori	58
4.4	Diagram Sekuens	59
5	IMPLEMENTASI DAN PENGUJIAN PERANGKAT LUNAK	61
5.1	Implementasi Perangkat Lunak	61
5.1.1	Lingkungan Implementasi Perangkat Keras	61
5.1.2	Lingkungan Implementasi Perangkat Lunak	61
5.1.3	Hasil Implementasi Perangkat Lunak	62
5.2	Pengujian Perangkat Lunak	62
5.2.1	Lingkungan Pengujian Perangkat Keras	62
5.2.2	Lingkungan Pengujian Perangkat Lunak	62
5.2.3	Pengujian Fungsional	63
5.2.4	Hasil Pengujian Fungsional	70
5.2.5	Pengujian Eksperimental	70
6	KESIMPULAN DAN SARAN	75
6.1	Kesimpulan	75
6.2	Saran	75
	DAFTAR REFERENSI	77
	A KODE PROGRAM	79
	B THE SOURCE CODE	87

DAFTAR GAMBAR

2.1	Gambar Skenario Aplikasi Layanan Web	7
2.2	Gambar Skenario Aplikasi yang Terinstal	8
2.3	Gambar Skenario Aplikasi Sisi Klien (JavaScript)	9
2.4	Gambar Skenario Aplikasi Pada Perangkat Dengan Masukan Yang Terbatas	10
2.5	Gambar Skenario Layanan Akun	11
2.6	Gambar Markdown Cetak Tebal dan Cetak Miring	14
2.7	Gambar Markdown Judul Bab	15
2.8	Gambar Markdown Batas Baris Dengan Tiga Spasi	15
2.9	Gambar Markdown Batas Baris Tanpa Tiga Spasi	16
2.10	Gambar Markdown Paragraf	16
2.11	Gambar Markdown Menampilkan Gambar	17
2.12	Gambar Markdown Link	17
2.13	Gambar Markdown Kode	18
2.14	Gambar Markdown Kutipan	18
2.15	Gambar Markdown Garis Horisontal	19
2.16	Gambar Markdown Daftar Tidak Berurutan	19
2.17	Gambar Markdown Daftar Berurutan	20
2.18	Gambar GFM Garis Bawah	20
2.19	Gambar GFM Taut Otomatis URL	21
2.20	Gambar GFM Tanda Coret	21
2.21	Gambar GFM Blok Kode	21
2.22	Gambar GFM Penandaan Sintaks	22
2.23	Gambar GFM Tabel 1	22
2.24	Gambar GFM Tabel 2	22
2.25	Gambar GFM Tabel 3	23
2.26	Gambar GFM Tabel 4	23
2.27	Gambar GFM Tabel 5	23
2.28	Gambar Contoh Grid	26
2.29	Gambar Grid Utama	27
2.30	Gambar Grid Kecil	27
2.31	Gambar Grid Menengah	27
2.32	Gambar Grid Tingkat Lanjut	28
2.33	Gambar Grid Dengan Offset	28
2.34	Gambar Grid Dengan Baris yang Tidak Lengkap	29
2.35	Gambar Grid Dengan Baris yang Mengempis/Tidak Mengempis	29
2.36	Gambar Grid Dengan Kolom yang Berpusat	30
2.37	Gambar Grid Dengan Sumber yang Terurut	30
2.38	Gambar Contoh Tombol	31
2.39	Gambar Tombol Dasar	31
2.40	Gambar Ukuran dan Bentuk Tombol	32
2.41	Gambar Warna-warna Tombol	32
2.42	Gambar Aksesibilitas Tombol	33

2.43	Gambar Tabel Foundation	33
2.44	Gambar Contoh Navigasi	34
2.45	Gambar Contoh Plugins	34
3.1	Google Developers Console	37
3.2	Membuat Proyek Baru	37
3.3	Menu Credentials	38
3.4	Membuat Client ID yang Baru	38
3.5	Tipe Aplikasi	39
3.6	Pengisian Tipe Aplikasi	39
3.7	Client ID	40
3.8	Izin Pihak Pengguna	40
3.9	Skenario Aplikasi SIRM	41
3.10	Output Sintaks Cetak Tebal dan Cetak Miring	42
3.11	Output Sintaks Judul Bab	42
3.12	Output Sintaks Batas Baris	43
3.13	Output Sintaks Paragraf	43
3.14	Output Sintaks Link	43
3.15	Output Sintaks Daftar	43
3.16	Output Keterangan Mahasiswa	44
3.17	Tampilan pilihmahasiswa.php Dengan Zurb Foundation	46
3.18	Use Case Diagram Sistem Informasi Riwayat Mahasiswa	46
3.19	Entity Relationship Diagram	50
3.20	Data Context Diagram Sistem Informasi Riwayat Mahasiswa	51
3.21	Data Flow Diagram Sistem Informasi Riwayat Mahasiswa	51
4.1	Desain Antarmuka Halaman Awal	53
4.2	Desain Antarmuka Pilih Mahasiswa	54
4.3	Desain Antarmuka Info Mahasiswa	54
4.4	Desain Antarmuka Edit Mahasiswa	55
4.5	Desain Antarmuka Lihat Histori	55
4.6	Desain Antarmuka Lihat Versi Ini	56
4.7	Desain Antarmuka Entri Baru	56
4.8	Diagram Sekuens Bagian Satu	59
4.9	Diagram Sekuens Bagian Dua	60
4.10	Diagram Sekuens Bagian Tiga	60
5.1	Membuka Halaman index.php	64
5.2	Login Dengan Email yang Diakhiri "@student.unpar.ac.id"	65
5.3	Konfirmasi Email yang Dikelola oleh student.unpar.ac.id	65
5.4	CAS UNPAR	65
5.5	Izin Akses Dari Pihak Pengguna	66
5.6	Login Dengan Email yang Diakhiri "@gmail.com"	66
5.7	Alert Email yang Digunakan Tidak Dapat Mengakses SIRM	66
5.8	Memilih Mahasiswa	67
5.9	Melihat Info Mahasiswa	67
5.10	Mengedit Info Mahasiswa	67
5.11	Melihat Histori	68
5.12	Keterangan Versi Pertama	68
5.13	Keterangan Versi Kedua	68
5.14	Template Entri Baru	69
5.15	Membuat Entri Baru	69

5.16 Entri Baru Berhasil Dibuat 69

DAFTAR TABEL

2.1	Tabel 2-1 Daftar Pengujian Zurb Foundation	25
3.1	Tabel 3-1 Skenario Login	47
3.2	Tabel 3-2 Skenario Pilih Mahasiswa	48
3.3	Tabel 3-3 Skenario Melihat Info Mahasiswa	48
3.4	Tabel 3-4 Skenario Edit Mahasiswa	49
3.5	Tabel 3-5 Skenario Lihat Histori	49
3.6	Tabel 3-6 Skenario Membuat Entri Baru	49
4.1	Tabel 4-1 Modul Login	57
4.2	Tabel 4-2 Modul Pilih Mahasiswa	57
4.3	Tabel 4-3 Modul Info Mahasiswa	57
4.4	Tabel 4-4 Modul Edit Mahasiswa	58
4.5	Tabel 4-5 Modul Lihat Histori	58
4.6	Tabel 4-6 Modul Entri Baru	58
4.7	Tabel 4-7 Rancangan Tabel Info Mahasiswa	59
4.8	Tabel 4-8 Rancangan Tabel Histori	59
5.1	Tabel 5-1 Hasil Pengujian Fungsional	70
5.2	Tabel 5-2 Catatan Waktu Pengujian Eksperimental Kelompok Pertama	71
5.3	Tabel 5-3 Catatan Waktu Pengujian Eksperimental Kelompok Kedua	72
5.4	Tabel 5-4 Perbandingan Rata-rata Waktu Pengujian Eksperimental	73

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Saat ini jumlah dosen dan jumlah mahasiswa menjadi pemasalahan, disebabkan minimnya jumlah dosen. Kurangnya tenaga dosen mengakibatkan seorang dosen harus menjadi dosen wali atau dosen pembimbing banyak mahasiswa dalam satu waktu. Kesulitan yang dimiliki oleh setiap dosen adalah kesulitan dalam mengingat perkembangan setiap mahasiswa serta sejarah setiap mahasiswa.

Maka dari itu berdasarkan jabaran masalah diatas, baik untuk dibuat sebuah perangkat lunak yang mencatat riwayat setiap mahasiswa. Dimana semua dosen yang telah terautentikasi dapat berkontribusi untuk memantau perkembangan setiap mahasiswa. Lalu setiap aksi yang dilakukan pada mahasiswa baik aksi *edit* maupun aksi *view* dicatat sehingga dapat dilihat historinya. Dan yang terakhir setiap perubahan dicatat revisinya sehingga dapat dipantau apa saja yang telah dirubah.

Untuk membangun aplikasi tersebut, teknologi yang digunakan adalah Google Authentication, Markdown Syntax, StrapdownJS, Zurb Foundation, PHP, dan MySQL. Google Authentication akan digunakan untuk mengautentikasi setiap dosen pada saat login. Kemudian semua format penulisan akan menggunakan Markdown Syntax. Lalu untuk menampilkan penulisan dalam format Markdown Syntax ke halaman website menggunakan StrapdownJS. Perangkat lunak ini menggunakan Zurb Foundation untuk membuat tampilan antarmuka. Yang terakhir untuk kebutuhan fungsional dan basis data akan menggunakan PHP dan MySQL.

1.2 Rumusan Masalah

Berdasarkan latar belakang maka dapat dirumuskan permasalahan sebagai berikut:

- Bagaimana mengautentikasi pengguna menggunakan Google Authentication?
- Bagaimana menggunakan teks dengan format Markdown?
- Bagaimana menampilkan teks dengan format Markdown ke halaman website?
- Bagaimana merancang antarmuka Sistem Informasi Riwayat Mahasiswa menggunakan Zurb Foundation?
- Bagaimana mengimplementasikan Sistem Informasi Riwayat Mahasiswa yang telah dirancang kedalam *script* PHP?

1.3 Tujuan

Berdasarkan rumusan masalah yang ditulis dalam sub bab 2, tujuan utama yang ingin dicapai melalui penelitian ini adalah:

- Mengautentikasi pengguna menggunakan Google Authentication.
- Menggunakan teks dengan format Markdown Syntax.
- Menampilkan teks dengan format Markdown Syntax ke halaman website.
- Merancang antarmuka Sistem Informasi Riwayat Mahasiswa menggunakan Zurb Foundation.
- Mengimplementasikan Sistem Informasi Riwayat Mahasiswa yang telah dirancang ke dalam *script* PHP.

1.4 Batasan Masalah

Dalam penelitian ini ditetapkan batasan-batasan yang akan menjadi pedoman dalam pelaksanaan penelitian:

- Perangkat lunak akan memiliki 6 fitur yaitu: Login, Pilih mahasiswa, Melihat info mahasiswa, Edit mahasiswa, Lihat histori, dan Membuat entri baru.
- Untuk fitur login hanya untuk dosen yang diakhiri dengan @unpar.ac.id dan *username* bukan angka semua.
- Untuk fitur pilih mahasiswa, pengguna dapat memilih mahasiswa yang ingin dilihat atau dirubah dan pengguna juga bisa menekan tombol "Add" untuk menambah mahasiswa baru.
- Untuk fitur melihat info mahasiswa, pengguna dapat melihat info terkini dari mahasiswa dan aksi ini dicatat dalam log untuk alasan penjagaan privasi.
- Untuk fitur edit mahasiswa, pengguna dapat mengubah info mahasiswa dan aksi ini juga dicatat dalam log.
- Untuk fitur lihat histori, pengguna dapat melihat histori setiap aksi perubahan atau aksi *view*.
- Untuk fitur membuat entri baru, saat membuat entri baru akan dibuatkan *template* sehingga kedepannya isi info setiap mahasiswa seragam.

1.5 Metodologi Penelitian

Metodologi yang digunakan untuk menyusun penelitian:

- Melakukan studi pustaka mengenai teknologi yang akan digunakan untuk membangun Sistem Informasi Riwayat Mahasiswa.

- 1 • Menganalisis cara kerja teknologi yang akan digunakan untuk membangun Sistem
- 2 Informasi Riwayat Mahasiswa.
- 3 • Merancang Sistem Informasi Riwayat Mahasiswa yang akan dibuat.
- 4 • Melakukan implementasi untuk Sistem Informasi Riwayat Mahasiswa yang telah di-
- 5 rancang ke dalam PHP.
- 6 • Melakukan pengujian perangkat lunak yang telah diimplementasikan.

7 1.6 Sistematika Pembahasan

8 Sistematika pembahasan dalam penelitian ini adalah sebagai berikut:

- 9 • Bab I Pendahuluan
- 10 Bab ini menjelaskan latar belakang permasalahan, rumusan masalah, tujuan, batasan
- 11 masalah, metodologi penelitian, dan sistematika pembahasan.
- 12 • Bab II Dasar Teori
- 13 Bab ini menjelaskan teori-teori dasar mengenai Google Authentication, Markdown
- 14 Syntax, StrapdownJS, Zurb Foundation, PHP, dan MySQL yang menjadi referensi
- 15 utama dalam pelaksanaan penelitian.
- 16 • Bab III Analisis
- 17 Bab ini berisi analisis mengenai Google Authentication, Markdown Syntax, Strapdo-
- 18 wnJS, Zurb Foundation, PHP, dan MySQL yang akan digunakan pada penelitian ini.
- 19 • Bab IV Perancangan
- 20 Bab ini berisi perancangan Sistem Informasi Riwayat Mahasiswa yang akan dibuat.
- 21 • Bab V Implementasi dan Pengujian
- 22 Bab ini berisi pengimplementasian dan pengujian Sistem Informasi Riwayat Mahasis-
- 23 wa.
- 24 • Bab VI Kesimpulan dan Saran
- 25 Bab ini berisi kesimpulan dari hasil penelitian dan saran untuk pengembangan lebih
- 26 lanjut.

BAB 2

LANDASAN TEORI

Bab ini terdiri atas empat bagian, yaitu Google Authentication, Markdown Syntax, StrapdownJS dan Zurb Foundation. Empat bagian tersebut akan membahas mengenai dasar-dasar teori mengenai Google Authentication, Markdown Syntax, StrapdownJS dan Zurb Foundation yang akan digunakan dalam penelitian ini untuk membangun perangkat lunak Sistem Informasi Riwayat Mahasiswa.

2.1 Google Authentication [1]

API Google menggunakan protokol OAuth 2.0 untuk otentikasi dan otorisasi. OAuth 2.0 adalah protokol yang relatif sederhana. Untuk memulainya cukup dengan mendapatkan kepercayaan OAuth 2.0 dari Google Developers Console¹. Maka aplikasi akan meminta suatu token akses dari Google Authorization Server, ekstrak token akses yang merupakan jawaban dari server, dan mengirim token akses ke Google API yang akan diakses.

Sub bab berikut memberikan gambaran skenario otorisasi OAuth 2.0 yang merupakan dukung dari Google. Rincian tentang cara menggunakan OAuth 2.0 untuk otentikasi (yaitu *sign-in*), dapat dilihat pada OpenID Connect².

2.1.1 Langkah Dasar

Semua aplikasi akan mengikuti pola dasar ketika mengakses Google API menggunakan OAuth 2.0. Terdapat empat langkah yang harus diikuti :

1. Mendapatkan kepercayaan OAuth 2.0 dari Google Developers Console
Berkunjung ke Google Developers Console untuk mendapatkan kepercayaan OAuth 2.0 seperti klien id dan kerahasiaan klien yang keduanya dikenal oleh Google dan aplikasi yang dibuat. Set nilai-nilai yang bervariasi sesuai dengan jenis aplikasi apa yang sedang dibuat. Misalnya, sebuah aplikasi JavaScript tidak memerlukan sebuah rahasia, tapi apakah aplikasi layanan web memerlukannya.
2. Memperoleh token akses dari Google Authorization Server
Sebelum aplikasi dapat mengakses data privat dengan menggunakan Google API, terlebih dahulu diperlukan token akses untuk mengakses API tersebut. Satu token akses dapat memberikan berbagai tingkat akses ke beberapa API. Izin token akses merupakan parameter untuk variabel ruang lingkup yang mengontrol sumber daya dan operasi.

¹<https://console.developers.google.com/>

²<https://developers.google.com/accounts/docs/OpenIDConnect>

Selama ada permintaan untuk token akses, maka aplikasi akan mengirimkan satu atau lebih nilai pada parameter ruang lingkup.

Ada beberapa cara dan variasi untuk melakukan permintaan tersebut berdasarkan aplikasi yang dibangun. Contohnya aplikasi JavaScript mungkin meminta token akses menggunakan mesin pencari yang mengarah kembali ke Google, namun aplikasi yang dibangun dan diimplementasi pada perangkat tidak memiliki fitur mesin pencari maka akan menggunakan layanan web. Beberapa permintaan memerlukan tahap otentikasi dimana pengguna diharuskan login menggunakan akun Google mereka. Setelah login pengguna akan ditanya apakah pengguna akan memberi izin untuk aplikasi yang telah melakukan permintaan tersebut. Proses ini disebut izin dari pihak pengguna. Jika pengguna memberi izin, maka Google Authorization Server akan mengirimkan aplikasi tersebut sebuah token akses. Jika pengguna tidak memberi izin, maka server akan menunjukkan respon yang menyatakan error.

3. Kirim token akses ke API

Setelah aplikasi mendapat token akses, lalu aplikasi akan mengirimkan token akses ke Google API melalui otorisasi yang terletak pada header HTTP. Sangat mungkin untuk mengirimkan token sebagai parameter permintaan URI dalam tipe data *string*, namun langkah ini tidak direkomendasikan karena parameter URI akan berakhir pada file log yang tidak aman. Juga merupakan hal yang baik karena menghindari menciptakan nama parameter URI yang tidak perlu. Token akses hanya berlaku untuk set operasi dan sumber daya yang dijelaskan pada lingkup permintaan token. Sebagai contoh, jika token akses dikeluarkan untuk Google+ API, hal tersebut tidak memberikan akses untuk Google Contact API. Namun token akses untuk Google+ API dapat dikirim beberapa kali untuk operasi yang serupa.

4. Memperbaharui token akses jika diperlukan

Token akses memiliki daya tahan yang terbatas. Jika aplikasi yang dibangun membutuhkan akses ke Google API melebihi masa aktif token akses, maka dapat memperbaharui token akses tersebut. Hal ini memungkinkan untuk mendapatkan token akses yang baru.

2.1.2 Skenario Google Authentication

Terdapat lima skenario yang dapat digunakan untuk Google Authentication yaitu Skenario Aplikasi Web Server, Skenario Aplikasi yang Terinstal, Skenario Aplikasi Sisi Klien (JavaScript), Skenario Aplikasi Pada Perangkat Dengan Masukan Yang Terbatas, dan Skenario Layanan Akun. Untuk penjelasan lebih lanjut dapat dilihat pada sub-sub-bab berikut.

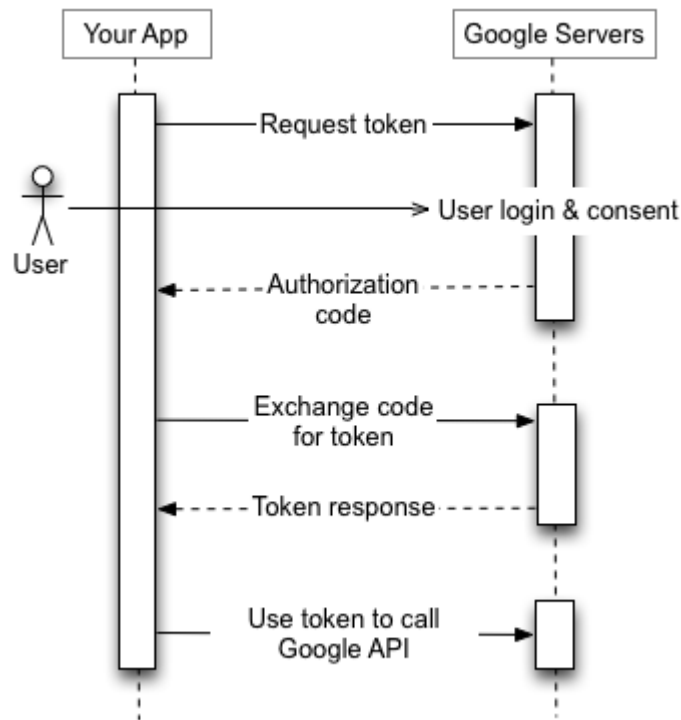
2.1.2.1 Skenario Aplikasi Layanan Web

Google OAuth 2.0 mendukung aplikasi Layanan Web yang menggunakan bahasa dan kerangka kerja seperti PHP, Java, Python, Ruby, dan ASP.NET.

Urutan otorisasi dimulai ketika aplikasi mengarahkan mesin pencari ke URL Google; URL tersebut termasuk parameter permintaan yang menunjukkan jenis akses yang diminta.

1 Google menangani otentikasi pengguna, pemilihan sesi, dan izin dari pihak pengguna. Ha-
 2 silnya adalah sebuah kode otorisasi, dimana aplikasi dapat bertukar untuk token akses dan
 3 memperbaharui token akses.

4 Aplikasi harus menyimpan pembaharuan token akses untuk penggunaan kedepannya
 5 dan menggunakan token akses untuk mengakses Google API. Setelah masa token akses
 6 berakhir, maka aplikasi akan memperbaharui token akses untuk mendapatkan yang baru.
 7 Untuk gambaran skenario dapat dilihat pada Gambar 2.1.



Gambar 2.1: Skenario Aplikasi Layanan Web

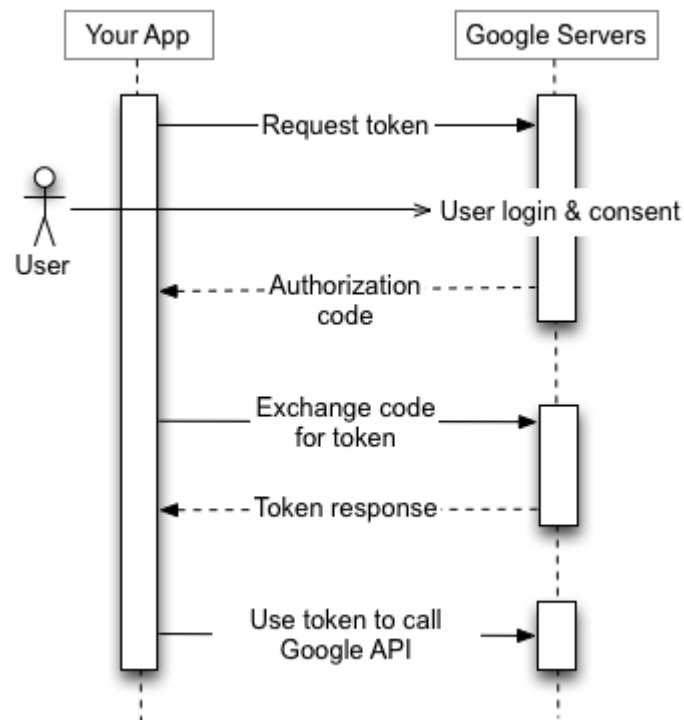
8 2.1.2.2 Skenario Aplikasi yang Terinstal

9 Google OAuth 2.0 mendukung aplikasi yang diinstal pada perangkat seperti komputer, per-
 10angkat *mobile*, dan tablet. Ketika membuat klien id melalui Google Developers Console,
 11 menentukan aplikasi yang terinstal kemudian pilih Android, Chrome, iOS, atau "*Other*"
 12 sebagai jenis aplikasi.

13 Hasil proses klien id dan kerahasiaan klien dalam beberapa kasus dimasukkan dalam
 14 kode sumber aplikasi. (Dalam konteks ini, kerahasiaan klien jelas tidak diperlakukan sebagai
 15 rahasia.)

16 Urutan otorisasi dimulai ketika aplikasi mengarahkan mesin pencari ke URL Google;
 17 URL termasuk parameter permintaan yang menunjukkan jenis akses yang diminta. Google
 18 menangani otentikasi pengguna, pemilihan sesi, dan izin pengguna. Hasilnya adalah sebuah
 19 kode otorisasi yang dapat bertukar untuk token akses dan memperbaharui token.

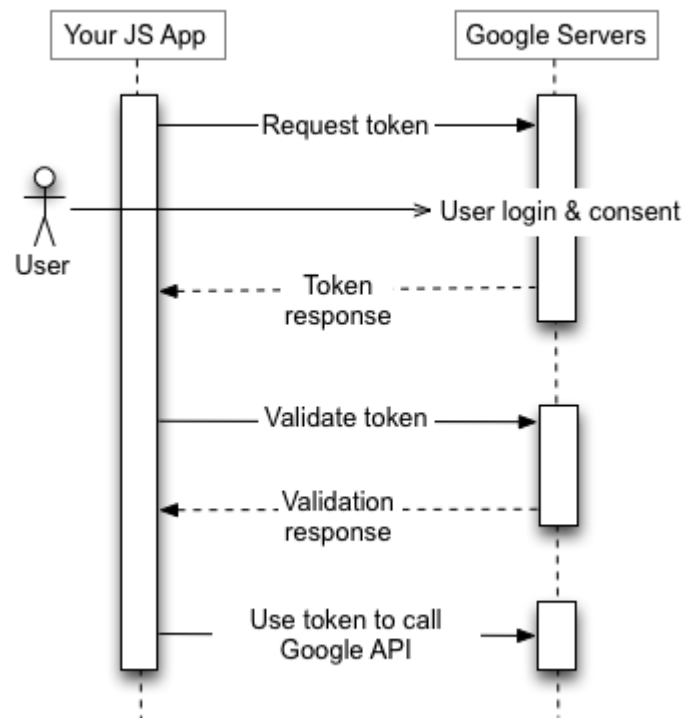
20 Aplikasi harus menyimpan token yang diperbaharui untuk penggunaan masa depan dan
 21 menggunakan token akses untuk mengakses API Google. Setelah masa token akses berakhir,
 22 maka aplikasi akan memperbaharui token untuk mendapatkan yang baru. Untuk gambar
 23 skenario dapat dilihat pada Gambar 2.2.



Gambar 2.2: Skenario Aplikasi yang Terinstal

1 2.1.2.3 Skenario Aplikasi Sisi Klien (JavaScript)

2 Google OAuth 2.0 mendukung aplikasi JavaScript yang berjalan di mesin pencari. Urutan
3 otorisasi dimulai ketika aplikasi mengarahkan mesin pencari ke URL Google; URL termasuk
4 parameter permintaan yang menunjukkan jenis akses yang diminta. Google menangani
5 otentikasi pengguna, pemilihan sesi, dan izin pengguna. Hasilnya adalah token akses dimana
6 klien harus memvalidasi sebelum memasukkannya ke dalam permintaan Google API. Ketika
7 masa token berakhir, aplikasi mengulangi proses. Untuk gambar skenario dapat dilihat pada
8 Gambar 2.3.

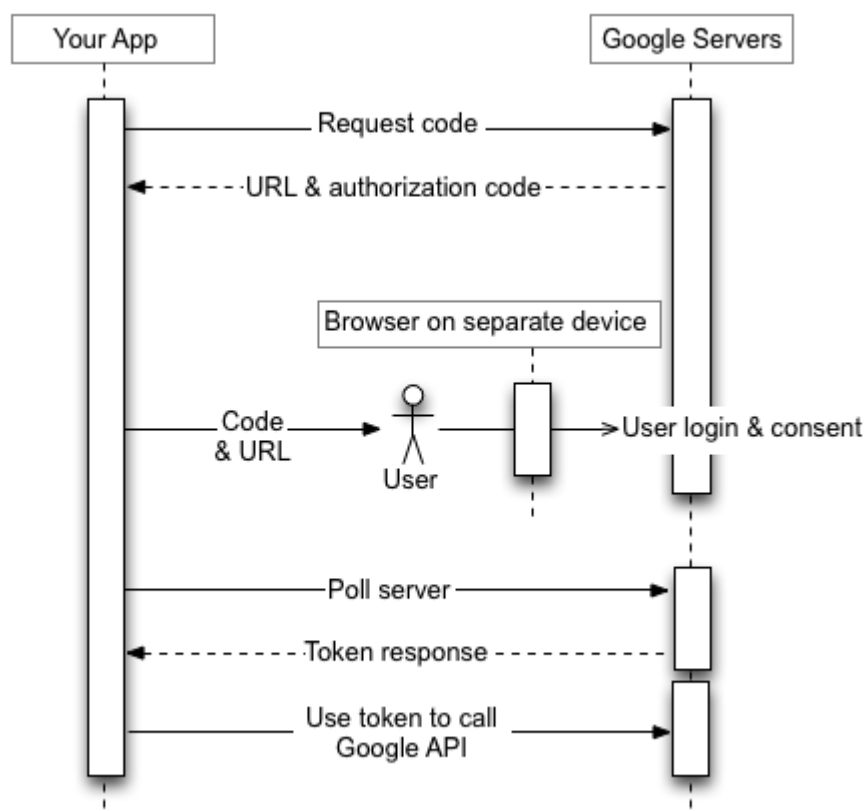


Gambar 2.3: Skenario Aplikasi Sisi Klien (JavaScript)

1 2.1.2.4 Skenario Aplikasi Pada Perangkat Dengan Masukan Yang Terbatas

2 Google OAuth 2.0 mendukung aplikasi yang berjalan pada perangkat dengan masukan yang
 3 terbatas seperti konsol game, kamera video, dan printer. Urutan otorisasi dimulai dengan
 4 aplikasi membuat permintaan layanan web ke URL Google untuk kode otorisasi. Tang-
 5 gapan berisi beberapa parameter, termasuk URL dan kode bahwa aplikasi menunjukkan
 6 kepada pengguna. Pengguna memperoleh URL dan kode dari perangkat, kemudian beralih
 7 ke perangkat terpisah atau komputer dengan kemampuan masukan yang lebih. Pengguna
 8 membuka mesin pencari, menavigasi ke URL tertentu, melakukan *log in*, dan memasukan
 9 kode.

10 Sementara itu, aplikasi jajak pendapat dari URL Google pada interval tertentu. Setelah
 11 pengguna menyetujui akses, respon dari server Google berisi token akses dan memperbaharui
 12 token. Aplikasi harus menyimpan token yang baru untuk penggunaan masa depan dan
 13 menggunakan token akses untuk mengakses Google API. Setelah masa token akses berakhir,
 14 maka aplikasi akan memperbaharui token untuk mendapatkan yang baru. Untuk gambar
 15 skenario dapat dilihat pada Gambar 2.4.



Gambar 2.4: Skenario Aplikasi Pada Perangkat Dengan Masukan Yang Terbatas

2.1.2.5 Skenario Layanan Akun

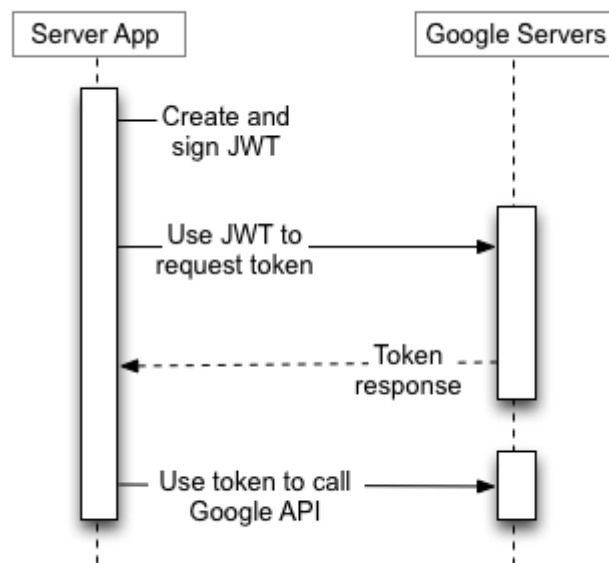
Google API seperti Prediction API dan Google Cloud Storage dapat bertindak atas nama aplikasi yang dibuat tanpa mengakses informasi pengguna. Dalam situasi ini aplikasi perlu membuktikan identitasnya sendiri ke API, tapi tidak diperlukan izin dari pihak pengguna. Demikian pula, dalam skenario perusahaan, aplikasi dapat meminta akses didelegasikan ke beberapa sumber daya.

Untuk jenis interaksi antara server memerlukan layanan akun, dimana akun tersebut terdapat pada aplikasi yang dibuat, bukan individu ke pengguna akhir. Aplikasi memanggil Google API atas nama layanan akun, dan izin dari pihak pengguna tidak diperlukan. (Dalam skenario tanpa layanan akun, aplikasi memanggil Google API atas nama pengguna akhir, dan izin dari pihak pengguna kadang-kadang diperlukan.)

Catatan: skenario layanan akun ini membutuhkan aplikasi untuk membuat dan tanda kriptografi JSON Web Token (JWTs). Sangat disarankan untuk menggunakan perpustakaan untuk melakukan tugas-tugas ini. Jika menulis kode ini tanpa menggunakan perpustakaan secara abstrak tanda penciptaan dan penandatanganan, mungkin membuat kesalahan yang akan memiliki dampak yang parah pada keamanan aplikasi yang dibangun.

Kredensial layanan akun, yang diperoleh dari Google Developers Console, termasuk alamat email yang dihasilkan yang unik, klien id, dan setidaknya satu pasang kunci publik / privat. Menggunakan klien id dan satu kunci privat untuk membuat JWT ditandatangani dan membangun permintaan token akses dalam format yang sesuai. Aplikasi kemudian mengirimkan permintaan token ke Google OAuth 2.0 Authorization Server, yang mengem-

1 balikan token akses. Aplikasi menggunakan token untuk mengakses API Google. Ketika
 2 masa token berakhir, aplikasi mengulangi proses. Untuk gambar skenario dapat dilihat
 3 pada Gambar 2.5.



Gambar 2.5: Skenario Layanan Akun

4 2.1.3 Masa Habis Berlaku Token

5 Kode token harus ditulis untuk mengantisipasi kemungkinan bahwa token yang diberikan
 6 mungkin tidak lagi bekerja suatu saat. Token mungkin berhenti bekerja untuk beberapa
 7 alasan di bawah ini:

- 8 • Pengguna telah mencabut akses.
- 9 • Token tidak digunakan selama enam bulan.
- 10 • Akun pengguna telah melampaui jumlah tertentu permintaan token.

11 Saat ini batas untuk setiap akun Google adalah 25 token. Jika pengguna akun telah
 12 memiliki 25 token, permintaan otentikasi untuk token ke-26 akan berhasil tapi token yang
 13 paling tua atau token ke-1 akan dibuat tidak berlaku tanpa sepengetahuan pengguna. Jika
 14 perlu untuk mengotorisasi beberapa program, mesin, atau perangkat, salah satu solusi ada-
 15 lah untuk membatasi jumlah klien dimana harus mengotorisasi per pengguna akun antara
 16 15 atau 20. Jika Anda adalah admin Google Apps, Anda dapat membuat admin tambahan
 17 untuk mengizinkan beberapa klien.

18 2.1.4 Lingkup Otorisasi [2]

19 Lingkup disini merupakan sebuah *string* yang memungkinkan akses ke sumber daya ter-
 20 tentu, misalnya akses ke data pengguna. Dengan memasukkan lingkup tertentu pada saat
 21 permintaan otorisasi, kemudian mendapatkan izin sesuai dengan teks yang akan ditampilkan
 22 ke pengguna. Setelah mendapat persetujuan dari pihak pengguna untuk izin atas lingkup

tersebut, maka Google mengirimkan token untuk aplikasi yang mengidentifikasi untuk memberikan otorisasi khusus. Dengan kata lain, lingkup dan token menentukan apa saja data pengguna yang diberi izin oleh pengguna untuk diakses.

Sebuah aplikasi yang dibuat tanpa permintaan otentikasi (tidak ada lingkup yang diminta) hanya dapat mengakses data pengguna yang umum di Google+. Contoh, jika sebuah aplikasi mencari postingan publik, respon dari pencarian akan menampilkan id pengguna yang telah diposting secara publik dan aplikasi dapat mengakses nama dan URL foto pengguna yang dimana keduanya selalu diposting secara publik. Dapat juga mengakses tanggal ulang tahun atau jenis kelamin pengguna jika pengguna telah memposting secara publik. Untuk daftar lingkup otorisasi dapat dilihat pada sub-sub-bab berikut.

2.1.4.1 Lingkup Profil

| `profile`

Lingkup ini merupakan lingkup dasar dimana lingkup ini melakukan beberapa hal seperti berikut:

- Meminta agar aplikasi diberikan akses ke informasi profil dasar bagi pengguna yang terotentikasi.
- Memungkinkan aplikasi untuk mengetahui siapa pengguna yang dikonfirmasi dengan mengganti id pengguna dengan "*me*" yang mewakili pengguna yang telah terotentikasi disetiap permintaan yang dilakukan.
- Memungkinkan aplikasi diakses melalui aplikasi android.

| `https://www.googleapis.com/auth/plus.login`

Lingkup login disarankan untuk aplikasi yang menyediakan akses ke fitur sosial. Lingkup ini secara implisit mencakup lingkup profil dan juga meminta aplikasi diberikan akses ke:

- Rentang usia pengguna yang telah terotentikasi.
- Daftar teman yang telah diberikan akses oleh pengguna.
- Metode untuk membaca, menulis dan menghapus kegiatan app ke Google atas nama pengguna.

Lingkup ini juga memungkinkan lintas platform dengan pendaftaran tunggal.

2.1.4.2 Lingkup Email

| `email`

Lingkup ini meminta agar aplikasi diberikan akses ke:

- Alamat email Google dari pengguna. Mengakses alamat email dengan memanggil `people.get` yang akan mengeluarkan array email atau dengan memanggil

| `people.getOpenIdConnect`

yang akan mengeluarkan email dengan format OIDC (OpenID Connect).

1 • Nama domain Google Apps jika ada yang dimiliki pengguna. Nama domain dikem-
 2 balikan sebagai kepemilikan domain dari `people.get` atau properti `hd` dari `getOpenI-`
 3 `dConnect`.

4 • Lingkup email ini setara dan menggantikan lingkup di bawah ini.

5 | `https://www.googleapis.com/auth/userinfo.email`

6 | `https://www.googleapis.com/auth/plus.profile.emails.read`

7 Lingkup ini meminta aplikasi agar diberikan akses ke:

8 • Alamat email Google pengguna yang telah diverifikasi di *profile* Google+. Mengakses
 9 email dengan memanggil `people.get` dan mengembalikan hasil email dalam array.

10 • Nama domain yang telah didaftarkan di Google jika pengguna memiliki fitur tersebut.

11 2.1.4.3 Lingkup yang lain

12 | `openid`

13 Lingkup `openid` menginformasikan server otorisasi bahwa klien membuat permintaan Ope-
 14 nID Connect dan meminta akses ke id pengguna yang terotentikasi tersebut. Lingkup ini
 15 harus disertakan lingkup OpenId Connect.

16 Metode `getOpenIdConnect` mengembalikan profil pengguna dengan format OIDC meng-
 17 ikuti jalur permintaan HTTP:

18 | `https://www.googleapis.com/plus/v1/people/me/openIdConnect`

19 Untuk keperluan login menggunakan lingkup profil atau lingkup

20 | `https://www.googleapis.com/auth/plus.login`

21 karena lingkup

22 | `https://www.googleapis.com/auth/plus.me`

23 tidak dianjurkan sebagai lingkup login dikarenakan pengguna yang belum upgrade ke Goo-
 24 gle+ tidak akan mengembalikan nama atau alamat email pengguna.

25 Lingkup ini melakukan hal berikut:

26 • Memungkinkan aplikasi untuk mengetahui siapa pengguna yang dikonfirmasi dengan
 27 mengganti id pengguna dengan "*me*" yang mewakili pengguna yang telah teroten-
 28 tikasi disertai permintaan yang dilakukan.

29 2.1.4.4 Lingkup yang tidak dipakai lagi

30 | `https://www.googleapis.com/auth/userinfo.profile`

31 Ganti dengan lingkup yang setara yaitu lingkup profil. Lingkup ini setara dengan lingkup
 32 profil dan meminta akses data yang sama.

33 Catatan: lingkup ini tidak dipakai lagi namun tetap dipertahankan dan terus tersedia
 34 untuk kompatibilitas.

35 | `https://www.googleapis.com/auth/userinfo.email`

1 Ganti dengan lingkup yang setara yaitu lingkup email. Lingkup ini meminta akses ke alamat
 2 email akun Google pengguna. Google menghasilkan token baru dengan lingkup ini untuk
 3 titik akhir `people.get`. Lingkup ini juga meminta akses dari pengguna ke titik akhir `userinfo`
 4 untuk kompatibilitas.

5 Lihat juga lingkup terkait:

6 | `https://www.googleapis.com/auth/plus.profile.emails.read`

7 Catatan: lingkup ini tidak dipakai lagi namun tetap dipertahankan dan terus tersedia
 8 untuk kompatibilitas.

9 2.2 Markdown

10 2.2.1 Apa itu Markdown? [3]

11 John Gruber pembuat Markdown, memperkenalkan Markdown sebagai alat konversi sebuah
 12 teks untuk ditampilkan ke HTML untuk para penulis website. Markdown memungkinkan
 13 penulis mudah untuk membaca dan mudah untuk menulis sebuah teks biasa, lalu merubah
 14 teks tersebut secara struktural yang valid dengan XHTML atau HTML. Markdown memiliki
 15 beberapa sintaks yang sederhana sebagai peraturan dalam menulis, hal tersebut membuat
 16 mudah dalam konversi ke HTML dengan banyak perangkat lunak yang mendukung. Untuk
 17 contoh, jika menulis `**hello**` pada Markdown dan konversi ke HTML menggunakan teks
 18 editor yang mendukung, teks tersebut akan menjadi `hello` yang akan
 19 terlihat **hello**.

20 2.2.2 Sintaks yang Berguna [3]

21 Terdapat beberapa sintaks untuk penggunaan cetak tebal, cetak miring, judul sub bab, batas
 22 garis, paragraf, gambar, link, kode, kutipan, garis horisontal, dan list. Untuk penjelasan
 23 lebih lanjut dapat dilihat pada sub sub bab di bawah ini.

24 2.2.2.1 Cetak Tebal dan Cetak Miring

25 Markdown memperlakukan karakter bintang (*) sebagai penekanan. Teks yang dibungkus
 26 dengan satu karakter * maka hasil teks akan cetak miring, dan teks yang dibungkus dengan
 27 dua karakter * maka hasil teks akan cetak tebal. Berikut contoh penggunaan sintaks untuk
 28 cetak tebal dan cetak miring.

```
29 | *hello* untuk cetak miring
30 |
31 | **hello** untuk cetak tebal
```

32 Sintaks tersebut akan menghasilkan kode HTML sebagai berikut. Untuk hasil pada HTML
 33 dapat dilihat pada Gambar 2.6.

```
34 | <p><em>hello </em> untuk cetak miring</p>
35 |
36 | <p><strong>hello </strong> untuk cetak tebal</p>
```

hello untuk cetak miring

hello untuk cetak tebal

Gambar 2.6: Markdown Cetak Tebal dan Cetak Miring

2.2.2.2 Judul Bab

Markdown memperlakukan karakter hash (#) sebagai indikator dari bab. Gunakan beberapa karakter hash untuk bab. Selalu gunakan spasi antara karakter hash dengan teks yang akan digunakan. Jumlah # yang digunakan akan menentukan ukuran judul bab. Berikut contoh penggunaan sintaks untuk judul bab.

```
# Judul Bab
## Judul Sub Bab
### Judul Sub Sub Bab
#### Tingkat ke 4
##### Tingkat ke 5
##### Tingkat ke 6
```

Sintaks tersebut akan menghasilkan kode HTML sebagai berikut. Untuk hasil pada HTML dapat dilihat pada Gambar 2.6.

```
<h1>Judul Bab</h1>
<h2>Judul Sub Bab</h2>
<h3>Judul Sub Sub Bab</h3>
<h4>Tingkat ke 4</h4>
<h5>Tingkat ke 5</h5>
<h6>Tingkat ke 6</h6>
```

Judul Bab
Judul Sub Bab
Judul Sub Sub Bab
Tingkat ke 4
Tingkat ke 5
Tingkat ke 6

Gambar 2.7: Markdown Judul Bab

2.2.2.3 Batas Baris

Untuk menyisipkan satu baris baru dalam dokumen, mengakhiri baris dengan dua atau lebih spasi lalu tekan 'Enter'. Berikut contoh penggunaan sintaks untuk batas baris.

```
Baris ini dengan
batas baris
```

Contoh di atas, setelah kata 'dengan' diakhiri dengan tiga spasi lalu tekan 'Enter'. Sintaks tersebut akan menghasilkan kode HTML sebagai berikut. Untuk hasil pada HTML dapat dilihat pada Gambar 2.8.

```
<p>Baris ini dengan <br />
batas baris</p>
```

Baris ini dengan
batas baris

Gambar 2.8: Markdown Batas Baris Dengan Tiga Spasi

```

1 | Baris ini tanpa
2 | batas baris

```

Contoh di atas, setelah kata 'tanpa' diakhiri tanpa spasi langsung tekan 'Enter'. Sintaks tersebut akan menghasilkan kode HTML sebagai berikut. Untuk hasil pada HTML dapat dilihat pada Gambar 2.9.

```

6 | <p>Baris ini tanpa
7 | batas baris</p>

```

Baris ini tanpa batas baris

Gambar 2.9: Markdown Batas Baris Tanpa Tiga Spasi

2.2.2.4 Paragraf

Untuk menyisipkan paragraf baru, cukup menyisipkan satu baris kosong. Berikut contoh penggunaan sintaks untuk paragraf.

```

11 | Ini kalimat pertama. Ini kalimat berikutnya. Ini kalimat terakhir.
12 |
13 | Ini paragraf baru.

```

Contoh diatas baris pertama adalah peragraf kesatu. Setelah itu ada satu baris kosong. Kalimat kedua merupakan paragraf kedua. Sintaks tersebut akan menghasilkan kode HTML sebagai berikut. Untuk hasil pada HTML dapat dilihat pada Gambar 2.9.

```

17 | <p>Ini kalimat pertama. Ini kalimat berikutnya. Ini kalimat terakhir.</p>
18 |
19 | <p>Ini paragraf baru.</p>

```

Ini kalimat pertama. Ini kalimat berikutnya. Ini kalimat terakhir.

Ini paragraf baru.

Gambar 2.10: Markdown Paragraf

2.2.2.5 Gambar

Untuk menyisipkan gambar pada dokumen Markdown, gunakan sintaks berikut

```

22 | ![teks](/url_gambar "judul_gambar")

```

Contoh penggunaan sintaks gambar:

```

24 | ![logo](https://www.google.com/logos/doodles/2014/googles-16th-birthday-4613606054297600-hp.gif "
25 | Google")

```

Url gambar dapat diganti dengan path yang mengarah pada file gambar yang akan digunakan. Sintaks tersebut akan menghasilkan kode HTML sebagai berikut. Untuk hasil pada HTML dapat dilihat pada Gambar 2.11.

```

29 | <p></p>

```



Gambar 2.11: Markdown Menampilkan Gambar

2.2.2.6 Link

Untuk menyisipkan hyperlink pada dokumen Markdown, gunakan sintaks berikut

```
[link_teks](/tujuan_url "judul opsional")
```

Contoh penggunaan sintaks link:

```
[my_website](http://browsernative.com "Click Here")
```

Sintaks tersebut akan menghasilkan kode HTML sebagai berikut. Untuk hasil pada HTML dapat dilihat pada Gambar 2.12.

```
<p><a href="http://browsernative.com" title="Click Here">my_website</a></p>
```

[my_website](#)



Gambar 2.12: Markdown Link

2.2.2.7 Kode

Untuk menyisipkan kode pada sebuah baris pada dokumen Markdown, gunakan karakter kutip belakang ('). Lampirkan kode yang ingin disisipkan dalam karakter kutip belakang. Untuk menyisipkan blok kode gunakan tiga kutip belakang (``). Setiap baris kode harus diawali dengan empat spasi.

```
Definisi dari 'initLabels()' dapat dilihat di bawah ini"
'''
    function initLabels(){
        function setLabels(elementId, messageId){
            var label = document.querySelector('label[for=' + elementId + ']');
            label.textContent = chrome.i18n.getMessage(messageId);
        }
    }
'''
Paragraf lain .
```

'initLabels()' akan menjadi kode pada satu baris kalimat. Sedangkan fungsi yang ada didalam `` akan menjadi blok kode. Sintaks tersebut akan menghasilkan kode HTML sebagai berikut. Untuk hasil pada HTML dapat dilihat pada Gambar 2.13.

```
<p>Definisi dari 'initLabels()' dapat dilihat di bawah ini"</p>
<p>'''
Paragraf lain .
```

```

1      function initLabels(){
2          function setLabels(elementId, messageId){
3              var label = document.querySelector('label[for=' + elementId + ']');
4              label.textContent = chrome.i18n.getMessage(messageId);
5          }
6      }
7      '''</p>
8
9      <p>Paragraf lain.</p>

```

Definisi dari 'initLabels()' dapat dilihat di bawah ini"

```

''' function initLabels(){ function setLabels(elementId, messageId){ var label = document.querySelector('label[for=' + elementId + ']'); label.textContent =
chrome.i18n.getMessage(messageId);} } '''

```

Paragraflain.

Gambar 2.13: Markdown Kode

2.2.2.8 Kutipan

Untuk menyisipkan kutipan pada dokumen Markdown, gunakan tanda lebih besar (>) pada awal kupitan.

```

13      Ini merupakan kutipan **favorit** saya:
14
15      > The weak can never forgive.
16      > Forgiveness is the attribute of the strong.

```

Sintaks tersebut akan menghasilkan kode HTML sebagai berikut. Untuk hasil pada HTML dapat dilihat pada Gambar 2.14.

```

19      <p>Ini merupakan kutipan <strong>favorit </strong> saya:</p>
20
21      <blockquote>
22          <p>The weak can never forgive.
23          Forgiveness is the attribute of the strong.</p>
24      </blockquote>

```

Ini merupakan kutipan **favorit** saya:

The weak can never forgive. Forgiveness is the attribute of the strong.

Gambar 2.14: Markdown Kutipan

2.2.2.9 Garis Horisontal

Menyisipkan garis horisontal pada dokumen Markdown, gunakan tiga atau lebih tanda hubung (-) dalam baris baru. Maka akan muncul sebagai garis horisontal pada keluaran HTML.

```

28      # Bagian Satu
29
30      The quick brown fox jumps over the lazy dog. The quick brpwn fox jump over the lazy dog.
31
32      ———
33
34      Last Edited on *25th Dec 2014*

```

Pada contoh di atas — akan menjadi garis horisontal. Sintaks tersebut akan menghasilkan kode HTML sebagai berikut. Untuk hasil pada HTML dapat dilihat pada Gambar 2.15.

```

37      <h1>Bagian Satu</h1>
38
39      <p>The quick brown fox jumps over the lazy dog. The quick brpwn fox jump over the lazy dog.</p>
40
41      <hr />
42
43      <p>Last Edited on <em>25th Dec 2014</em></p>

```


Bagian Satu

The quick brown fox jumps over the lazy dog. The quick brpwn fox jump over the lazy dog.

Last Edited on *25th Dec 2014*

Gambar 2.15: Markdown Garis Horisontal

2.2.2.10 Daftar

Terdapat dua macam daftar yang dapat dibuat. Dua macam daftar tersebut dapat dilihat dibawah ini.

1. Daftar tidak berurutan

Untuk membuat daftar tidak berurutan dapat menggunakan simbol bintang *, simbol tambah +, maupun tanda hubung - sebelum daftar item yang ingin dimasukkan.

Untuk contoh penggunaan dapat dilihat di bawah ini.

```

8      * Item
9      * Item
10     * Item
11
12     + Item
13     + Item
14     + Item
15
16     - Item
17     - Item
18     - Item

```

Sintaks tersebut akan menghasilkan kode HTML sebagai berikut. Untuk hasil pada HTML dapat dilihat pada Gambar 2.16.

```

21     <ul>
22     <li>Item</li>
23     <li>Item</li>
24     <li><p>Item</p></li>
25     <li><p>Item</p></li>
26     <li>Item</li>
27     <li><p>Item</p></li>
28     <li><p>Item</p></li>
29     <li>Item</li>
30     <li>Item</li>
31     </ul>

```

- Item
- Item
- Item
- Item
- Item
- Item
- Item
- Item
- Item

Gambar 2.16: Daftar Tidak Berurutan

2. Daftar berurutan

Untuk membuat daftar berurutan dapat menggunakan nomor sebelum daftar item yang ingin dimasukkan. Untuk contoh penggunaan dapat dilihat di bawah ini.

```
1. Item 1
2. Item 2
3. Item 3
```

Sintaks tersebut akan menghasilkan kode HTML sebagai berikut. Untuk hasil pada HTML dapat dilihat pada Gambar 2.17.

```
<ol>
<li>Item 1</li>
<li>Item 2</li>
<li>Item 3</li>
</ol>
```

```
1. Item 1
2. Item 2
3. Item 3
```

Gambar 2.17: Markdown Daftar Berurutan

2.2.3 GitHub Flavored Markdown [4]

GitHub menggunakan "GitHub Flavored Markdown" atau disingkat menjadi GFM. GFM berbeda dengan Standar Markdwon (SM) dalam beberapa bagian yang cukup signifikan dan ada beberapa sintaks tambahan. Beberapa hal yang berbeda dari SM dapat dilihat di bawah ini.

2.2.3.1 Beberapa Garis Bawah Pada Kalimat

Pada Markdown kata yang berada diantara garis bawah akan dirubah menjadi cetak miring, namun pada GFM garis bawah tidak memiliki pasangan maka garis bawah tersebut diabaikan sehingga akan tetap tampil sebagai karakter garis bawah.

```
wow_great_stuff
do_this_and_do_that_and_another_thing.
```

Hal tersebut memungkinkan untuk merender kode dan nama dengan benar. Untuk menekankan sebagian kata dapat menggunakan tanda bintang (*). Untuk hasil pada HTML dapat dilihat pada Gambar 2.18.

```
wow_great_stuff do_this_and_do_that_and_another_thing.
```

Gambar 2.18: GFM Garis Bawah

2.2.3.2 Taut Otomatis URL

GFM membuat standar untuk taut otomatis URL. Tanpa mengatur teks untuk link sebuah URL, cukup menyisipkan URL dan URL tersebut akan menjadi taut otomatis yang mengarah ke URL tersebut.

```
http://example.com
```

- 1 Link di atas kan menjadi taut secara otomatis ke URL tersebut. Untuk hasil pada HTML
- 2 dapat dilihat pada Gambar 2.19.

<http://www.google.com>

Gambar 2.19: GFM Taut Otomatis URL

3 2.2.3.3 Tanda Coret

- 4 GFM menambahkan sintaks untuk membuat teks dicoret, yang dihilangkan dari SM. Untuk
- 5 membuat teks dicoret gunakan dua karakter tilde (~) antara kata yang akan dicoret.

6 | ~Mistaken text.~

- 7 Untuk hasil pada HTML dapat dilihat pada Gambar 2.20.

~~Mistaken text.~~

Gambar 2.20: GFM Tanda Coret

8 2.2.3.4 Blok Kode

- 9 SM mengkonfersi blok kode dengan diawali empat spasi untuk setiap baris yang berada
- 10 dalam blok kode. GFM juga mendukung blok kode namun cukup membungkus kode dengan
- 11 tiga kutip belakang (``) tanpa harus memperhatikan empat spasi untuk awalan kode.

```
12 | Here's an example:
13 |
14 | ```
15 | function test() {
16 |     console.log("notice the blank line before this function?");
17 | }
18 | ```
```

- 19 Untuk hasil pada HTML dapat dilihat pada Gambar 2.21.

Here's an example:

```
function test() {
  console.log("notice the blank line before this function?");
}
```

Gambar 2.21: GFM Blok Kode

20 2.2.3.5 Penandaan Sintaks

- 21 Blok kode dapat dilanjutkan dengan menambah sintaks. Dalam blok yang ditandai tam-
- 22 bahkan sebuah indentifikasi bahasa apa yang digunakan. Misalnya penandaan code sintaks
- 23 Ruby.

```
24 | ```ruby
25 | require 'redcarpet'
26 | markdown = Redcarpet.new("Hello World!")
27 | puts markdown.to_html
28 | ```
```

- 1 Untuk hasil pada HTML dapat dilihat pada Gambar 2.22.

```
require 'redcarpet'
markdown = Redcarpet.new("Hello World!")
puts markdown.to_html
```

Gambar 2.22: GFM Penandaan Sintaks

2 2.2.3.6 Tabel

- 3 Dapat membuat tabel dengan menyusun daftar kata dan membagi dengan tanda hubung
4 (-) untuk baris pertama. Kemudian memisahkan kolom dengan pipa |. Untuk contoh dapat
5 dilihat di bawah ini.

```
6 | First Header | Second Header
7 |-----|-----|
8 | Content Cell | Content Cell
9 | Content Cell | Content Cell
```

- 10 Untuk hasil pada HTML dapat dilihat pada Gambar 2.23.

First Header	Second Header
Content Cell	Content Cell
Content Cell	Content Cell

Gambar 2.23: GFM Tabel 1

- 11 Untuk tujuan estetika, dapat juga menambahkan pipa pada setiap ujung tabel. Untuk
12 contoh dapat dilihat di bawah ini.

```
13 | | First Header | Second Header |
14 | |-----|-----|
15 | | Content Cell | Content Cell |
16 | | Content Cell | Content Cell |
```

- 17 Untuk hasil pada HTML dapat dilihat pada Gambar 2.24.

First Header	Second Header
Content Cell	Content Cell
Content Cell	Content Cell

Gambar 2.24: GFM Tabel 2

- 18 Untuk membagi dengan tanda hubung (-) juga tidak perlu menyesuaikan panjang judul
19 maupun daftar kata. Untuk contoh dapat dilihat di bawah ini.

```
20 | Name | Description |
21 |-----|-----|
22 | Help | Display the help window. |
23 | Close | Closes a window |
```

- 24 Untuk hasil pada HTML dapat dilihat pada Gambar 2.25.

Name	Description
Help	Display the help window.
Close	Closes a window

Gambar 2.25: GFM Tabel 3

Selain itu dapat memasukan berbagai sintaks Markdown kedalam tabel seperti link, cetak tebal, cetak miring atau teks yang dicoret. Untuk contoh dapat dilihat di bawah ini.

```

3 | Name | Description |
4 | -----|-----|
5 | Help | ~Display the~ help window. |
6 | Close | _Closes_ a window |

```

Untuk hasil pada HTML dapat dilihat pada Gambar 2.26.

Name	Description
Help	Display the help window.
Close	<i>Closes</i> a window

Gambar 2.26: GFM Tabel 4

Dengan memasukan tanda titik dua pada baris header. Dapat menentukan baris kiri, baris tengah, maupun baris kanan. Untuk contoh dapat dilihat di bawah ini.

```

10 | Left-Aligned | Center Aligned | Right Aligned |
11 | :-----: | :-----: | :-----: |
12 | col 3 is | some wordy text | $1600 |
13 | col 2 is | centered | $12 |
14 | zebra stripes | are neat | $1 |

```

Untuk hasil pada HTML dapat dilihat pada Gambar 2.27.

Left-Aligned	Center Aligned	Right Aligned
col 3 is	some wordy text	\$1600
col 2 is	centered	\$12
zebra stripes	are neat	\$1

Gambar 2.27: GFM Tabel 5

2.3 StrapdownJS [5]

Strapdown.js membuat lebih sederhana untuk membuat dokumen Markdown yang elegan. Tidak diperlukan kompilasi dari sisi server. Gunakan strapdown.js untuk mendokumentasikan proyek dengan cepat, membuat tutorial, membuat halaman utama sebuah website. Contoh website yang menggunakan strapdown.js adalah <http://strapdownjs.com/>.

Untuk penggunaan StrapdownJS gunakan skrip strapdown.js dengan cara langsung mengarahkan ke website maupun menggunakan path yang mengarahkan dimana file strapdown.js berada.

- Cara pertama:

```
<script src="http://strapdownjs.com/v/0.2/strapdown.js"></script>
```

- Cara kedua:

```
<script src="v/0.2/strapdown.js"></script>
```

Peringatan untuk cara kedua harus terlebih dahulu mengunduh file strapdown.js.

Lalu buat tag `<xml>` untuk membuat area untuk menulis dengan sintaks Markdown.

```
<xmp theme="united" style="display:none;">
</xmp>
```

Menulis dengan Markdown dapat dilakukan diantara tag `<xml>` seperti di atas.

Untuk contoh lengkap penggunaan strapdown.js dapat dilihat pada template HTML dibawah ini dan taruh pada file server statis untuk mencobanya:

```
<!DOCTYPE html>
<html>
<title>Hello Strapdown</title>
<xmp theme="united" style="display:none;">
# Markdown text goes in here

## Chapter 1

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

## Chapter 2

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
</xmp>
<script src="http://strapdownjs.com/v/0.2/strapdown.js"></script>
</html>
```

Strapdown.js juga memiliki beberapa fitur :

1. Ramah dengan mesin pencari
2. Kompatibel dengan berbagai browser (Sudah diuji dengan ponsel menggunakan Safari, IE 8/9, Firefox, Chrome)
3. Github menggunakan Markdown (Tabel, Syntax, Headline)
4. Dapat menggunakan tema

2.4 Zurb Foundation [6]

Zurb Foundation merupakan alat bantu dalam membuat aplikasi baru maupun membuat website yang responsif. Jutaan desainer dan teknisi menggunakan Foundation sebagai bagian

Tabel 2.1: Daftar Pengujian Zurb Foundation¹

Browser/OS	The Grid	Layout/UI	JS
Chrome	✓	✓	✓
Firefox	✓	✓	✓
Safari	✓	✓	✓
IE10	✓	✓	✓
IE11	✓	✓	✓
IE9	✓	✓	✓
IE8	✗	✗	✗
IE7	✗	✗	✗
iOS (iPhone)	✓	✓	✓
iOS (iPad)	✓	✓	✓
Android 2, 4 (Phone)	✓	✓	✓
Android 2, 4 (Tablet)	✓	✓	✓
Windows Phone 7+	✓	✓	✓
Surface	✓	✓	✓

1 dari alur kerja mereka. Zurb Foundation adalah *framework* pertama yang memperkenalkan
 2 konsep responsif, semantik, mobile dan parsial. Zurb Foundation juga kompatibel dengan
 3 kebanyakan mesin pencari dan perangkat. Maka dari itu Zurb Foundation merupakan pilihan
 4 profesional bagi para desainer dan teknisi.

5 2.4.1 Kompatibilitas

6 Zurb Foundation dirancang dan diuji pada berbagai browser dan perangkat. Daftar peng-
 7 ujian pada berbagai browser dan perangkat dapat dilihat pada Tabel 2.1.

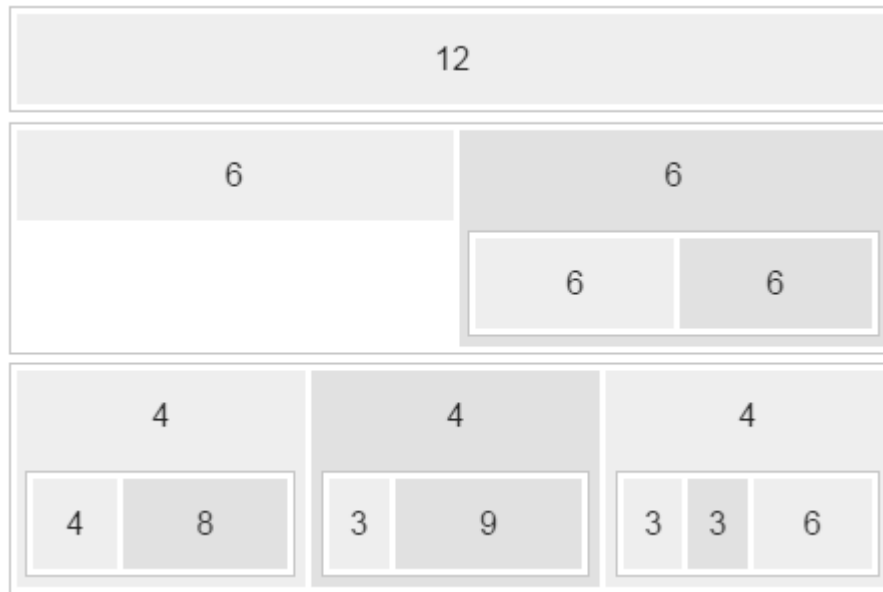
8 2.4.2 Apa Saja yang Dapat Dibuat Dengan Foundation?

9 Foundation memiliki banyak komponen dan struktur untuk membantu membangun sebuah
 10 situs responsif. Untuk komponen Foundation dapat melihat beberapa gambar dibawah ini :

11 1 Grid

12 Grid bekerja pada hampir semua perangkat dan memiliki dukungan untuk menjadi satu
 13 kesatuan, sumber pemesanan, offset dan perangkat presentasi. Hal tersebut sedikit mu-
 14 dah dengan waktu yang singkat dapat menciptakan tata letak yang kompleks seperti ini.
 15 Untuk contoh grid dapat dilihat pada Gambar 2.28.

¹<http://foundation.zurb.com/docs/compatibility.html>



Gambar 2.28: Contoh Grid

- Grid Utama

Mulailah membuat elemen dengan kelas secara berturut-turut. Ini akan membuat blok horisontal yang berisi kolom vertikal. Kemudian tambahkan beberapa div dengan kelas kolom pada baris tersebut. Dapat menggunakan *column* atau *columns* karena hanya berbeda tata bahasa. Tentukan lebar dari setiap kolom dengan menggunakan kelas *small-#*, *medium-#*, dan *large-#*. Foundation merupakan yang pertama dalam mengembangkan kode untuk layar kecil dan untuk perangkat dengan layar lebih besar akan mendapat bagian dengan gaya yang sama. Berikut kode HTML untuk membuat grid utama dan dapat dilihat hasilnya pada Gambar 2.29.

```

10  <!-- no. 1 -->
11  <div class="row">
12    <div class="small-2 large-4 columns">2</div>
13    <div class="small-4 large-4 columns">4</div>
14    <div class="small-6 large-4 columns">6</div>
15  </div>
16  <!-- no. 2 -->
17  <div class="row">
18    <div class="large-3 columns">full</div>
19    <div class="large-6 columns">full</div>
20    <div class="large-3 columns">full</div>
21  </div>
22  <!-- no. 3 -->
23  <div class="row">
24    <div class="small-6 large-2 columns">6</div>
25    <div class="small-6 large-8 columns">6</div>
26    <div class="small-12 large-2 columns">full</div>
27  </div>
28  <!-- no. 4 -->
29  <div class="row">
30    <div class="small-3 columns">3</div>
31    <div class="small-9 columns">9</div>
32  </div>
33  <!-- no. 5 -->
34  <div class="row">
35    <div class="large-4 columns">full</div>
36    <div class="large-8 columns">full</div>
37  </div>
38  <!-- no. 6 -->
39  <div class="row">
40    <div class="small-6 large-5 columns">6</div>
41    <div class="small-6 large-7 columns">6</div>
42  </div>
43  <!-- no. 7 -->
44  <div class="row">
45    <div class="large-6 columns">full</div>

```



```

1      <div class="large-6 columns">full </div>
2    </div>

```



Gambar 2.29: Grid Utama

• Grid Kecil

Menampilkan grid kecil ke layar besar lebih mudah dibandingkan memaksa menampilkan grid besar ke layar kecil. Berikut kode HTML untuk membuat grid kecil dan dapat dilihat hasilnya pada Gambar 2.30.

```

7      <!-- no. 1 -->
8      <div class="row">
9        <div class="small-2 columns">2 columns</div>
10       <div class="small-10 columns">10 columns</div>
11     </div>
12     <!-- no. 2 -->
13     <div class="row">
14       <div class="small-3 columns">3 columns</div>
15       <div class="small-9 columns">9 columns</div>
16     </div>

```



Gambar 2.30: Grid Kecil

• Grid Menengah

Layar berukuran sedang akan mendapatkan bagian dari gaya yang kecil, kecuali ditentukan tata letak yang berbeda dengan menggunakan grid menengah. Berikut kode HTML untuk membuat grid menengah dan dapat dilihat hasilnya pada Gambar 2.31.

```

22     <!-- no. 1 -->
23     <div class="row">
24       <div class="medium-2 columns">2 columns</div>
25       <div class="medium-10 columns">10 columns</div>
26     </div>
27     <!-- no. 2 -->
28     <div class="row">
29       <div class="medium-3 columns">3 columns</div>
30       <div class="medium-9 columns">9 columns</div>
31     </div>

```



Gambar 2.31: Grid Menengah

• Grid Tingkat Lanjut

Dapat membuat grid bersarang dengan menggunakan kode HTML berikut. Untuk hasilnya dapat dilihat pada Gambar 2.32.

```

4      <div class="row">
5          <div class="small-8 columns">8
6              <div class="row">
7                  <div class="small-8 columns">8 Nested
8                      <div class="row">
9                          <div class="small-8 columns">8 Nested Again</div>
10                         <div class="small-4 columns">4</div>
11                     </div>
12                 </div>
13             <div class="small-4 columns">4</div>
14         </div>
15     </div>
16     <div class="small-4 columns">4</div>
17 </div>

```



Gambar 2.32: Grid Tingkat Lanjut

• Grid Dengan Offset

Memindahkan blok hingga 11 kolom ke kanan dengan menggunakan kelas *large-offset-1* dan *small-offset-3*. Berikut contoh penggunaan pada kode HTML dan dapat dilihat hasilnya pada Gambar 2.33.

```

22     <!-- no. 1 -->
23     <div class="row">
24         <div class="large-1 columns">1</div>
25         <div class="large-11 columns">11</div>
26     </div>
27     <!-- no. 2 -->
28     <div class="row">
29         <div class="large-1 columns">1</div>
30         <div class="large-10 large-offset-1 columns">10, offset 1</div>
31     </div>
32     <!-- no. 3 -->
33     <div class="row">
34         <div class="large-1 columns">1</div>
35         <div class="large-9 large-offset-2 columns">9, offset 2</div>
36     </div>
37     <!-- no. 4 -->
38     <div class="row">
39         <div class="large-1 columns">1</div>
40         <div class="large-8 large-offset-3 columns">8, offset 3</div>
41     </div>

```



Gambar 2.33: Grid Dengan Offset

• Grid Dengan Baris yang Tidak Lengkap

Untuk mengatasi perbedaan pada beberapa mesin pencari, Foundation akan memindahkan kolom terakhir berturut-turut ke kanan sehingga sejajar dengan tepi. Jika jumlah kolom tidak mencapai 12 maka pada kolom terakhir perlu ditandai dengan

1 kelas *end*. Berikut contoh penggunaan pada kode HTML dan dapat dilihat hasilnya
 2 pada Gambar 2.34.

```

3      <!-- no. 1 -->
4      <div class="row">
5          <div class="medium-3 columns">3</div>
6          <div class="medium-3 columns">3</div>
7          <div class="medium-3 columns">3</div>
8      </div>
9      <!-- no. 2 -->
10     <div class="row">
11         <div class="medium-3 columns">3</div>
12         <div class="medium-3 columns">3</div>
13         <div class="medium-3 columns end">3 end</div>
14     </div>

```

1.	3	3		3
2.	3	3	3 end	

Gambar 2.34: Grid Dengan Baris yang Tidak Lengkap

15 • Grid Dengan Baris yang Mengempis/Tidak Mengempis

16 Kelas ini memungkinkan untuk menghapus jarak antara kolom. Ada saat tidak
 17 setiap kueri memiliki sifat *collapsed* atau *uncollapsed*, cukup menambahkan kelas
 18 yang mengatur hal tersebut. Misal tidak menunjukkan jarak antara kolom pada
 19 kolom ukuran kecil dan menambahkan jarak antara kolom pada kolom ukuran
 20 menengah. Berikut contoh penggunaan pada kode HTML dan dapat dilihat hasilnya
 21 pada Gambar 2.35.

```

22     <div class="row medium-uncollapse large-collapse">
23         <div class="small-6 columns">
24             Removes gutter at large media query
25         </div>
26         <div class="small-6 columns">
27             Removes gutter at large media query
28         </div>
29     </div>

```

On a large screen, I have no gutters!	On a large screen, I have no gutters!
---------------------------------------	---------------------------------------

Gambar 2.35: Grid Dengan Baris yang Mengempis/Tidak Mengempis

30 • Grid Dengan Kolom yang Berpusat

31 Membuat kolom di tengah cukup dengan menambahkan kelas *small-centered* pada
 32 kolom tersebut. Tampilan pada layar besar akan mendapat bagian seperti tampilan
 33 pada layar kecil, tapi dapat menggunakan kelas *large-centered* untuk memusatkan di
 34 tampilan pada layar besar. Untuk tidak memusatkan di tampilan pada layar besar
 35 dapat menggunakan *large-uncentered*. Berikut contoh penggunaan pada kode HTML
 36 dan dapat dilihat hasilnya pada Gambar 2.36.

```

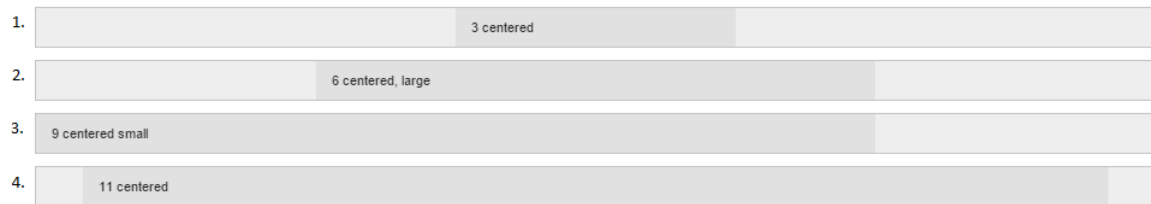
37     <!-- no. 1 -->
38     <div class="row">
39         <div class="small-3 small-centered columns">3 centered</div>
40     </div>
41     <!-- no. 2 -->
42     <div class="row">
43         <div class="small-6 large-centered columns">6 centered</div>
44     </div>
45     <!-- no. 3 -->
46     <div class="row">
47         <div class="small-9 small-centered large-uncentered columns">9 centered</div>

```

```

1      </div>
2      <!-- no. 4 -->
3      <div class="row">
4          <div class="small-11 small-centered columns">11 centered</div>
5      </div>

```



Gambar 2.36: Grid Dengan Kolom yang Berpusat

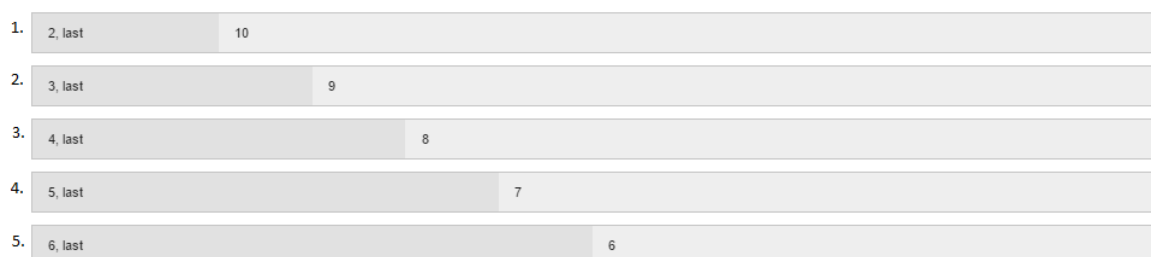
- Grid Dengan Sumber yang Terurut

Dengan menggunakan kelas ini memungkinkan untuk memindahkan kolom sesuai dengan titik yang ditentukan. Jika memiliki sub menu di bawah menu utama pada layar kecil, maka memiliki pilihan untuk posisi sub navigasi di kanan atau di kiri untuk tampilan pada layar besar. Berikut contoh penggunaan pada kode HTML dan dapat dilihat hasilnya pada Gambar 2.37.

```

12      <!-- no. 1 -->
13      <div class="row">
14          <div class="small-10 small-push-2 columns">10</div>
15          <div class="small-2 small-pull-10 columns">2, last</div>
16      </div>
17      <!-- no. 2 -->
18      <div class="row">
19          <div class="large-9 large-push-3 columns">9</div>
20          <div class="large-3 large-pull-9 columns">3, last</div>
21      </div>
22      <!-- no. 3 -->
23      <div class="row">
24          <div class="large-8 large-push-4 columns">8</div>
25          <div class="large-4 large-pull-8 columns">4, last</div>
26      </div>
27      <!-- no. 4 -->
28      <div class="row">
29          <div class="small-5 small-push-7 medium-7 medium-push-5 columns">7</div>
30          <div class="small-7 small-pull-5 medium-5 medium-pull-7 columns">5, last</div>
31      </div>
32      <!-- no. 5 -->
33      <div class="row">
34          <div class="medium-6 medium-push-6 columns">6</div>
35          <div class="medium-6 medium-pull-6 columns">6, last</div>
36      </div>

```

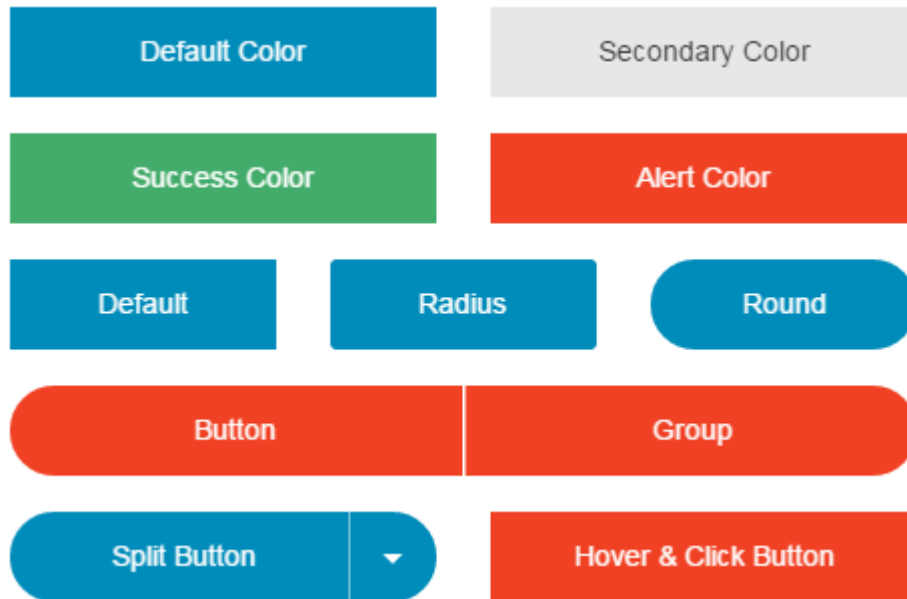


Gambar 2.37: Grid Dengan Sumber yang Terurut

2 Tombol

Mengklik tombol dengan material yang bagus merupakan hal yang mengagumkan. Mengklik tombol juga menghubungkan pengguna dengan berbagai aksi. Ada beberapa gaya tombol yang ringan untuk ukuran, presentasi, dan warna untuk menyesuaikan tombol

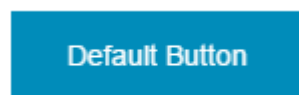
- 1 Anda sendiri semudah menambahkan kelas. Untuk contoh macam-macam tombol dapat
 2 dilihat pada Gambar 2.38.



Gambar 2.38: Contoh Tombol

- 3 • Tombol Utama
- 4 Dapat membuat tombol utama dengan kode HTML sebagai berikut dan dapat dilihat
 5 hasilnya pada Gambar 2.39.

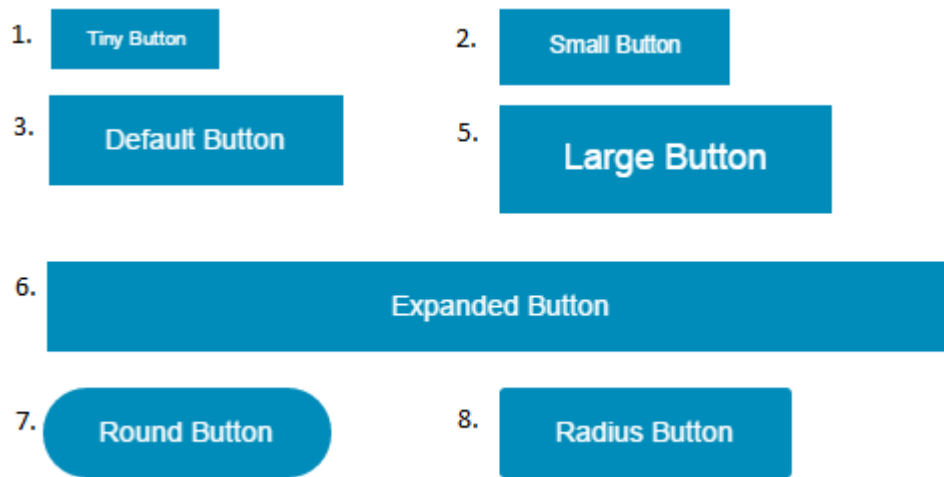
```
6 | <a href="#" class="button">Default Button</a>
```



Gambar 2.39: Tombol Dasar

- 7 • Mengubah Ukuran Tombol
- 8 Terdapat kelas tambahan untuk merubah bentuk dan ukuran tombol, berikut kode
 9 HTML yang dapat digunakan dan dapat dilihat hasilnya pada Gambar 2.40.

```
10 | <!-- Kelas untuk merubah ukuran -->
11 | 1. <a href="#" class="button tiny">Tiny Button</a>
12 | 2. <a href="#" class="button small">Small Button</a>
13 | 3. <a href="#" class="button">Default Button</a>
14 | 4. <a href="#" class="button disabled">Disabled Button</a>
15 | 5. <a href="#" class="button large">Large Button</a>
16 | 6. <a href="#" class="button expand">Expanded Button</a>
17 | <!-- Kelas untuk merubah bentuk -->
18 | 7. <a href="#" class="button round">Round Button</a>
19 | 8. <a href="#" class="button radius">Radius Button</a>
```

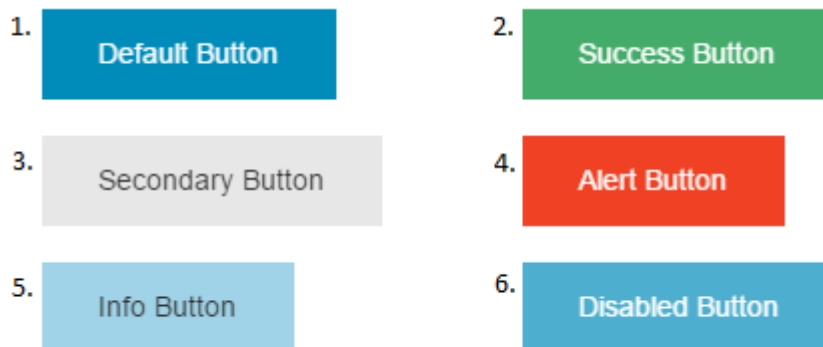


Gambar 2.40: Ukuran dan Bentuk Tombol

- Warna Tombol

Terdapat kelas tambahan untuk merubah warna tombol, berikut kode HTML yang dapat digunakan dan dapat dilihat hasilnya pada Gambar 2.41.

```
<!-- Kelas untuk merubah warna -->
1. <a href="#" class="button">Default Button</a>
2. <a href="#" class="button success">Success Button</a>
3. <a href="#" class="button secondary">Secondary Button</a>
4. <a href="#" class="button alert">Alert Button</a>
5. <a href="#" class="button info">Info Button</a>
6. <a href="#" class="button disabled">Disabled Button</a>
```



Gambar 2.41: Warna-warna Tombol

- Aksesibilitas

Gunakan kode HTML dibawah ini untuk membuat tombol lebih mudah diakses. Dapat menggunakan atribut `aria-label='submit form'` untuk memberikan petunjuk jika tidak ada keterangan pada tombol. Jika tombol tidak memiliki `` maka cukup menambahkan `tabindex="0"` pada `div` atau `span` untuk membuat tombol tersebut dapat difokuskan.

```
1. <a role="button" href="#" class="button">Default Button</a>
2. <a role="button" aria-label="submit form" href="#" class="button">Submit</a>
3. <div role="button" tabindex="0" class="button">Default Button</div>
```



Gambar 2.42: Aksesibilitas Tombol

3 Tabel

Dapat membuat tabel dengan menggunakan markup yang minim. Berikut contoh penggunaan tabel pada kode HTML dan dapat dilihat hasilnya pada Gambar 2.43.

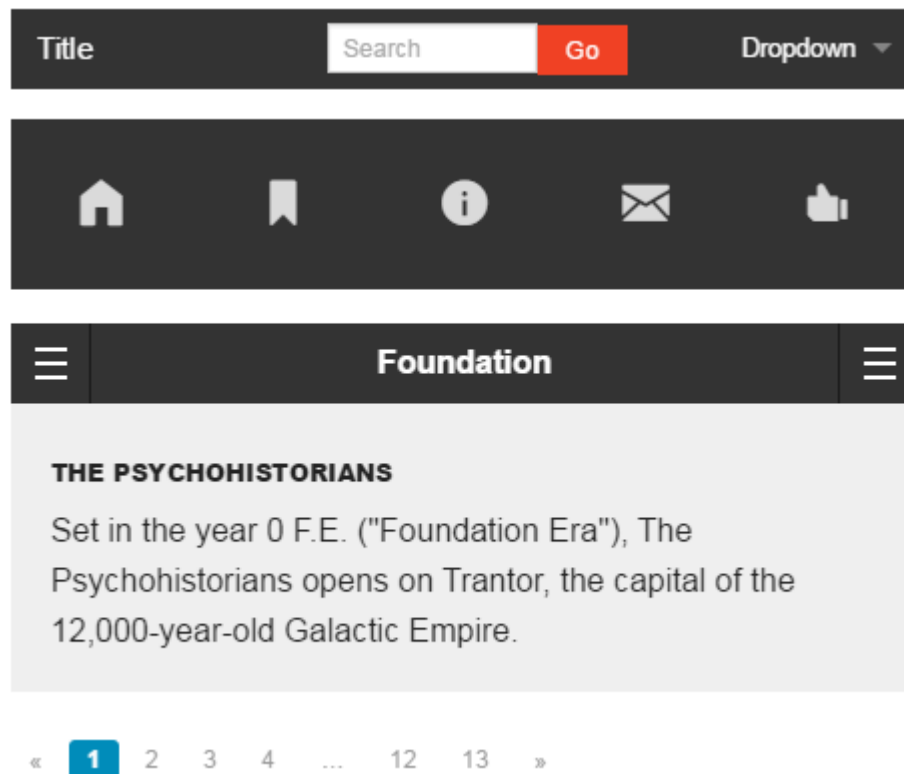
```
4 <table>
5 <thead>
6 <tr>
7 <th width="200">Table Header</th>
8 <th>Table Header</th>
9 <th width="150">Table Header</th>
10 <th width="150">Table Header</th>
11 </tr>
12 </thead>
13 <tbody>
14 <tr>
15 <td>Content Goes Here</td>
16 <td>This is longer content Donec id elit non mi porta gravida at eget metus.</td>
17 <td>Content Goes Here</td>
18 <td>Content Goes Here</td>
19 </tr>
20 <tr>
21 <td>Content Goes Here</td>
22 <td>This is longer Content Goes Here Donec id elit non mi porta gravida at eget metus
23 .</td>
24 <td>Content Goes Here</td>
25 <td>Content Goes Here</td>
26 </tr>
27 <tr>
28 <td>Content Goes Here</td>
29 <td>This is longer Content Goes Here Donec id elit non mi porta gravida at eget metus
30 .</td>
31 <td>Content Goes Here</td>
32 <td>Content Goes Here</td>
33 </tr>
34 </tbody>
35 </table>
```

Table Header	Table Header	Table Header	Table Header
Content Goes Here	This is longer content Donec id elit non mi porta gravida at eget metus.	Content Goes Here	Content Goes Here
Content Goes Here	This is longer Content Goes Here Donec id elit non mi porta gravida at eget metus.	Content Goes Here	Content Goes Here
Content Goes Here	This is longer Content Goes Here Donec id elit non mi porta gravida at eget metus.	Content Goes Here	Content Goes Here

Gambar 2.43: Tabel Foundation

4 Navigasi

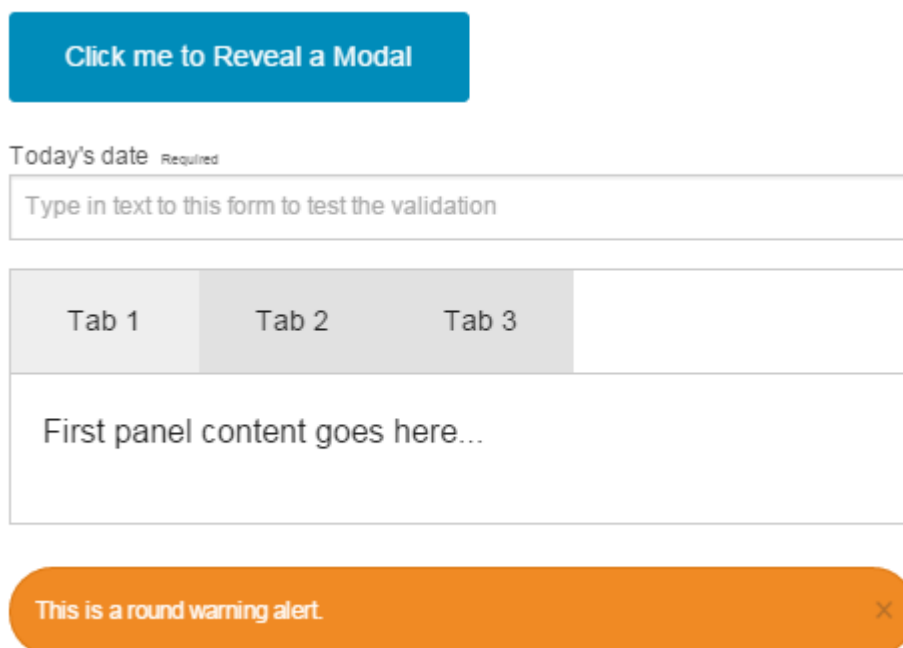
Orang yang mengakses harus bisa berkeliling melihat menu-menu yang ada. Gaya navigasi pada Foundation meliputi : bar bagian atas yang kuat dengan menu dropdown; tombol; bar pencari; ikon bar yang keren; implementasi kanvas yang lepas dari keluhan; dan sekelompok navigasi lainnya. Untuk contoh macam-macam navigasi dapat dilihat pada Gambar 2.44.



Gambar 2.44: Contoh Navigasi

5 Plugins

- Sudah meliputi banyak plugin javascript yang ditulis untuk modal dasar *pop-up*; menambatkan formulir validasi yang diperlukan; membuat tab konten; tanda peringatan; dan masih banyak lagi. Untuk contoh macam-macam *plugin* dapat dilihat pada Gambar 2.45.



Gambar 2.45: Contoh Plugins

BAB 3

ANALISIS

Bab ini terdiri atas lima bagian, yaitu Analisis Google Authentication, Analisis Markdown, Analisis StrapdownJS, Analisis Zurb dan Analisis Berorientasi Objek. Bagian Analisis Google Authentication berisi penjelasan analisis Google Authentication yang akan digunakan pada penelitian ini. Bagian Analisis Markdown berisi penjelasan analisis Markdown yang akan digunakan pada penelitian ini. Bagian Analisis StrapdownJS berisi penjelasan analisis StrapdownJS yang akan digunakan pada penelitian ini. Bagian Analisis Zurb Foundation berisi penjelasan analisis Zurb Foundation yang akan digunakan pada penelitian ini. Sedangkan bagian Analisis Kebutuhan Perangkat Lunak berisi use case diagram, skenario, entity relationship diagram, dan data flow diagram perangkat lunak yang akan dibangun.

3.1 Analisis Google Authentication

Pada penelitian ini untuk otentikasi fitur login akan menggunakan teknologi Google authentication atau dikenal OAuth 2.0. Untuk langkah-langkah penggunaan OAuth 2.0 dapat dilihat pada sub bab berikutnya.

3.1.1 Langkah Dasar Penggunaan OAuth 2.0

Berdasarkan langkah dasar yang terdapat pada bab 2, maka terdapat empat langkah yang akan diikuti untuk menggunakan OAuth 2.0 pada penelitian ini. Empat langkah yang diikuti:

1. Mendapatkan kepercayaan OAuth 2.0 dari Google Developers Console

(a) Mengunjungi Google Developers Console. Agar lebih jelas dapat dilihat pada Gambar 3.1.

(b) Buat sebuah proyek baru. Dapat dilihat pada Gambar 3.2.

(c) Masuk ke proyek yang telah dibuat dan masuk ke menu 'Credentials'. Agar lebih jelas dapat dilihat pada Gambar 3.3.

(d) Membuat client ID yang baru. Dapat dilihat pada Gambar 3.4.

(e) Pilih tipe aplikasi sesuai aplikasi yang dibangun, pada penelitian ini menggunakan tipe aplikasi web karena aplikasi yang akan dibangun berbasis web. Agar lebih jelas dapat dilihat pada Gambar 3.5.

(f) Isi bagian AUTHORIZED JAVASCRIPT ORIGINS (merupakan path dimana javascript otorisasi akan dijalankan) pada penelitian ini bagian AUTHORIZED JAVASCRIPT ORIGINS akan diisi dengan `http://localhost/` karena aplikasi yang akan dibangun pada penelitian ini terletak pada localhost dan AUTHORIZED REDIRECT URIS (merupakan pengarah jika otorisasi sudah berhasil) pada penelitian ini bagian AUTHORIZED REDIRECT URIS akan diisi dengan

```
| http://localhost/oauth.php
```

karena setelah menjalankan aplikasi dan berhasil melakukan otorisasi maka yang halaman pertama yang akan dituju adalah `oauth.php` untuk pembatasan user. Agar lebih jelas dapat dilihat pada Gambar 3.6.

(g) Setelah langkah-langkah diatas terpenuhi maka akan mendapatkan client id dan client secret. Client id dan client secret yang didapat dapat dilihat di bawah ini.

```
| Client id :
| 568951368854-ufmbistn0pcaq0khubafo1a133orfgve.apps.googleusercontent.com
| Client secret :
| -cSZ-AUmeQ9PaWWry_IpiBBi
```

Agar lebih jelas dapat dilihat pada Gambar 3.7.

2. Memperoleh token akses dari Google Authorization Server

Untuk memperoleh token akses akan menggunakan izin dari pihak pengguna. Jadi pada saat melakukan login, pengguna diharuskan login menggunakan akun Google sendiri. Setelah login pengguna akan ditanya dan akan memberi respon untuk memberi izin atau tidak pada aplikasi yang telah melakukan permintaan tersebut. Untuk gambar izin dari pihak pengguna dapat dilihat pada Gambar 3.8.

3. Kirim token ke API

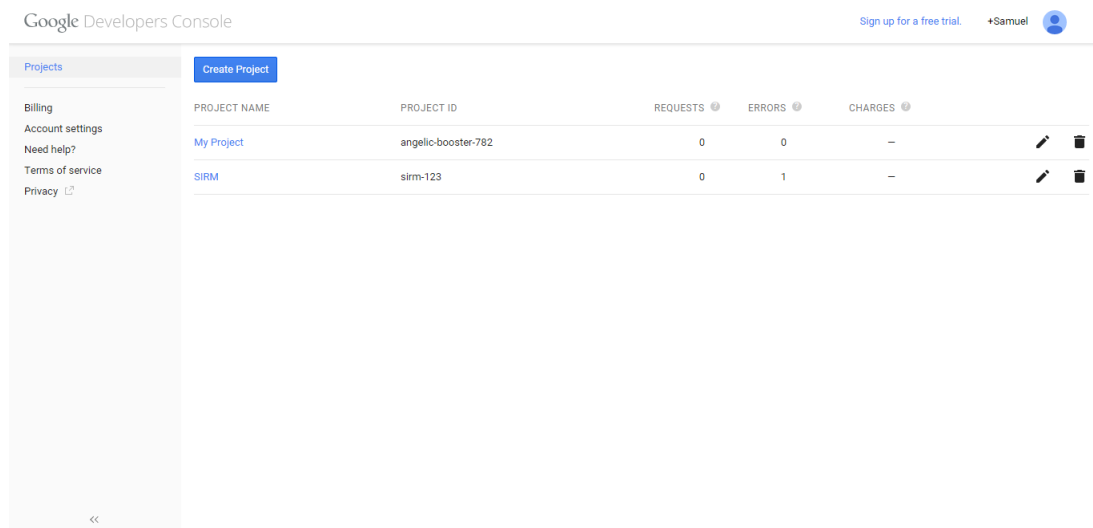
Setelah mendapatkan token akses, maka untuk mengirimkan ke API diperlukan scope. Karena sesuai dengan landasan teori, jika token akses dikeluarkan untuk Google+ API maka token akses tersebut tidak berlaku untuk mengakses Google Contact API. Scope yang akan digunakan pada penelitian ini adalah:

```
| https://www.googleapis.com/auth/plus.login
| dan
| email
```

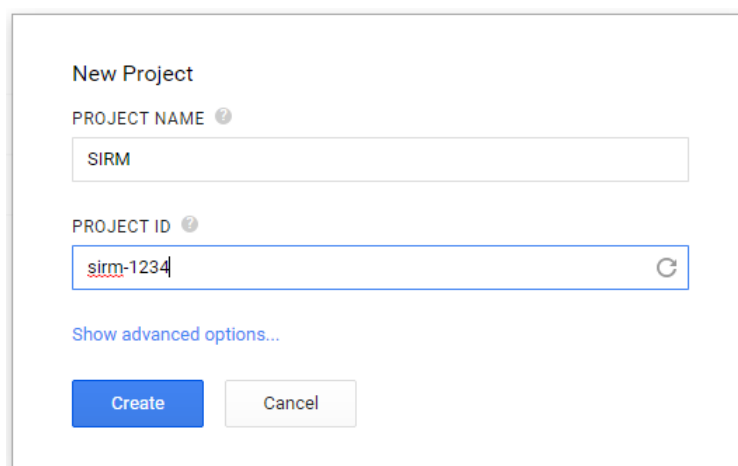
karena pada penelitian ini dibutuhkan email pengguna dan nama pengguna, keduanya itu dapat diperoleh dengan menggunakan kedua scope tersebut. Scope pertama untuk mendapatkan info dari pengguna dan scope kedua untuk mendapatkan email dari pengguna.

4. Memperbaharui token akses jika diperlukan

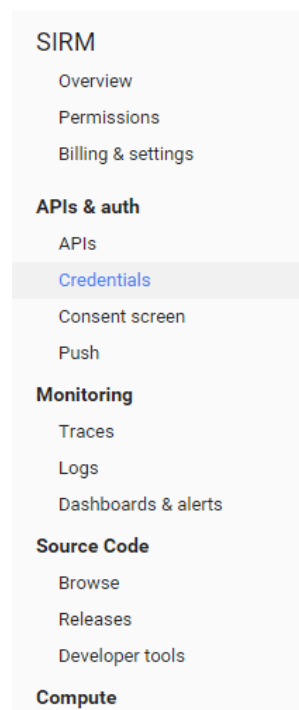
Pada penelitian ini tidak akan menggunakan tahap memperbaharui token akses karena token akses hanya digunakan selama penelitian ini berlangsung.



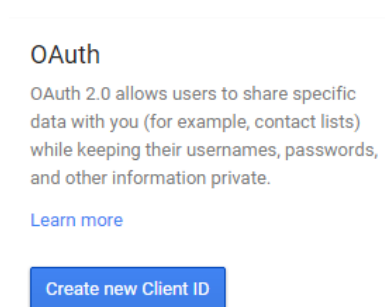
Gambar 3.1: Google Developers Console



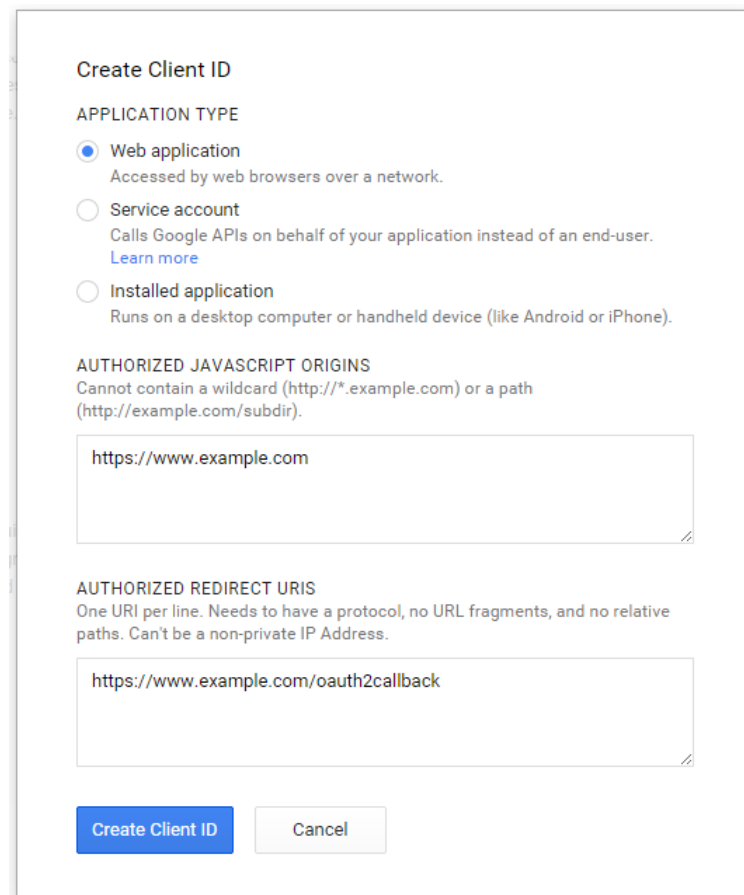
Gambar 3.2: Membuat Proyek Baru



Gambar 3.3: Menu Credentials



Gambar 3.4: Membuat Client ID yang Baru



Create Client ID

APPLICATION TYPE

☒ **Web application**
Accessed by web browsers over a network.

☐ **Service account**
Calls Google APIs on behalf of your application instead of an end-user. [Learn more](#)

☐ **Installed application**
Runs on a desktop computer or handheld device (like Android or iPhone).

AUTHORIZED JAVASCRIPT ORIGINS
Cannot contain a wildcard (`http://*.example.com`) or a path (`http://example.com/subdir`).

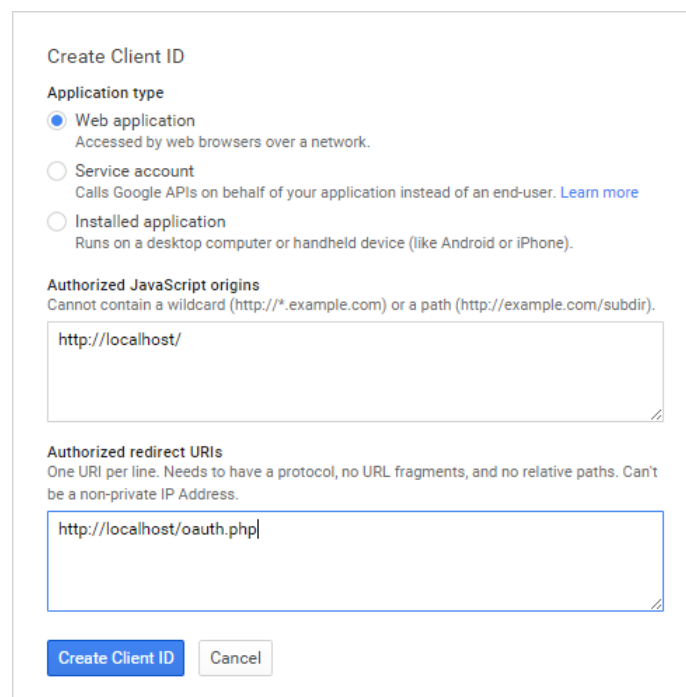
`https://www.example.com`

AUTHORIZED REDIRECT URIS
One URI per line. Needs to have a protocol, no URL fragments, and no relative paths. Can't be a non-private IP Address.

`https://www.example.com/oauth2callback`

Create Client ID **Cancel**

Gambar 3.5: Tipe Aplikasi



Create Client ID

Application type

☒ **Web application**
Accessed by web browsers over a network.

☐ **Service account**
Calls Google APIs on behalf of your application instead of an end-user. [Learn more](#)

☐ **Installed application**
Runs on a desktop computer or handheld device (like Android or iPhone).

Authorized JavaScript origins
Cannot contain a wildcard (`http://*.example.com`) or a path (`http://example.com/subdir`).

`http://localhost/`

Authorized redirect URIs
One URI per line. Needs to have a protocol, no URL fragments, and no relative paths. Can't be a non-private IP Address.

`http://localhost/oauth.php`

Create Client ID **Cancel**

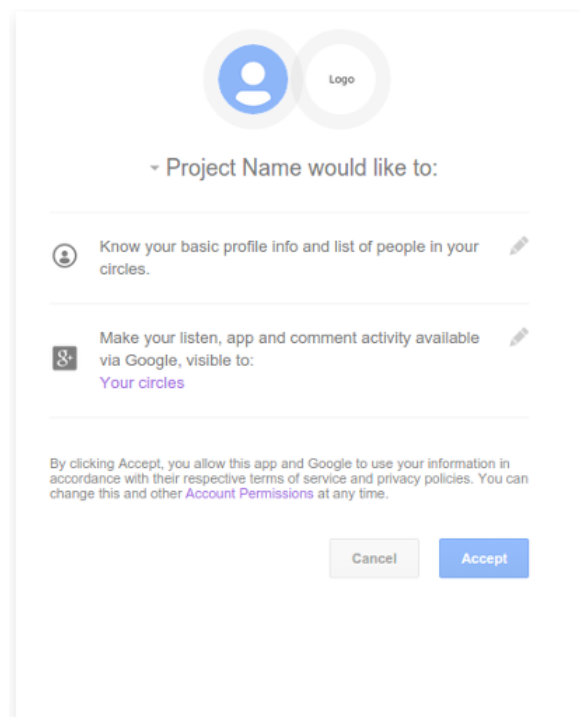
Gambar 3.6: Pengisian Tipe Aplikasi

Client ID for web application

CLIENT ID	568951368854-ufmbistn0pcaq0khubafo1a133orfgve.apps.googleusercontent.com
EMAIL ADDRESS	568951368854-ufmbistn0pcaq0khubafo1a133orfgve@developer.gserviceaccount.com
CLIENT SECRET	-cSZ-AUmeQ9PaWWry_lpiBBI
REDIRECT URIS	http://localhost/pilihmahasiswa.php
JAVASCRIPT ORIGINS	http://localhost/

Edit settings
Reset secret
Download JSON
Delete

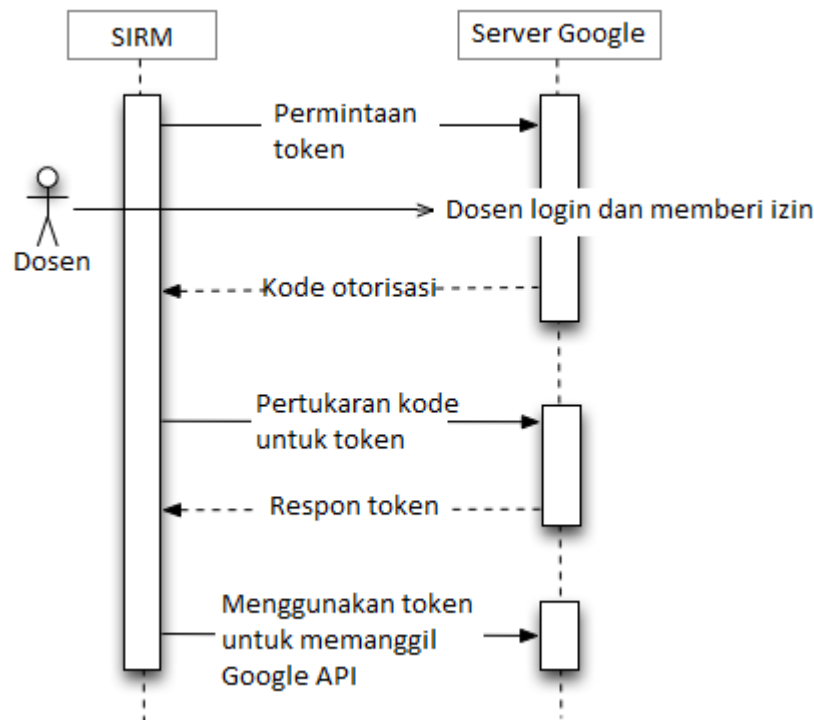
Gambar 3.7: Client ID



Gambar 3.8: Izin Pihak Pengguna

1 3.1.2 Skenario Aplikasi

2 Berdasarkan landasan teori skenario yang ada pada bab 2 dan berdasarkan perangkat lu-
3 nak yang akan dibangun, maka skenario yang akan digunakan pada penelitian ini adalah
4 skenario aplikasi web server. Aplikasi SIRM akan melakukan permintaan token ke Server
5 Google. Dosen sebagai pengguna akan melakukan login dan memberikan izin. Server Google
6 akan memberikan balasan berupa kode otorisasi. Kemudian aplikasi akan menukarkan kode
7 tersebut untuk mendapatkan token akses. Server Google memberikan token akses sebagai
8 respon penukaran kode otorisasi dengan token akses. Setelah aplikasi mendapatkan token
9 akses, maka aplikasi dapat memanggil Google API dengan menggunakan token akses. Untuk
10 skenario aplikasi SIRM dapat dilihat pada Gambar 3.9.



Gambar 3.9: Skenario Aplikasi SIRM

3.2 Analisis Markdown

Sintaks Markdown yang akan digunakan sesuai dengan landasan teori pada Bab 2. Sintaks Markdown akan digunakan pada bagian keterangan mahasiswa agar seragam dalam penulisan. Keterangan mahasiswa yang akan ditampilkan antara lain; NPM, nama, deskripsi umum, catatan. Maka dari itu sintaks Markdown yang akan digunakan adalah Cetak Tebal dan Cetak Miring, Judul Bab, Batas Baris, Paragraf, Link, dan Daftar.

- Sintaks Cetak Tebal dan Cetak Miring

Sintaks ini akan digunakan untuk memberikan penekanan pada satu kata dalam satu kalimat. Berikut penggunaan sintaks dan hasilnya dapat dilihat pada Gambar 3.10.

```
| **NPM** - *2010730013*
```

- Sintaks Judul Bab

Sintaks ini akan digunakan untuk menampilkan judul setiap bagian (NPM, nama, umum, dan catatan). Berikut penggunaan sintaks dan hasilnya dapat dilihat pada Gambar 3.11.

```
| # Judul 1
| ## Judul 2
| ### Judul 3
| #### Judul 4
| ##### Judul 5
| ##### Judul 6
```

- Sintaks Batas Baris

Sintaks ini digunakan pada penulisan paragraf jika diperlukan untuk mengakhiri sebuah baris atau ingin membuat baris baru. Berikut penggunaan sintaks dan hasilnya dapat dilihat pada Gambar 3.12.

```

1 | Baris ini dengan
2 | batas baris
3 |
4 | Baris ini tanpa
5 | batas baris

```

• Sintaks Paragraf

Sintaks ini akan digunakan untuk menulis deskripsi umum mahasiswa. Berikut penggunaan sintaks dan hasilnya dapat dilihat pada Gambar 3.13.

```

9 | Samuel adalah seorang mahasiswa yang periang namun terkadang sulit diatur. Dia aktif di
10 | himpunan sebagai ketua divisi pelayanan masyarakat.
11 |
12 | Grady adalah seorang mahasiswa yang memiliki jiwa pemimpin. Dia aktif di UKM sebagai ketua
13 | divisi logistik.

```

• Link

Sintaks ini akan digunakan untuk menampilkan website mahasiswa jika mahasiswa yang bersangkutan memiliki sebuah website maupun blog. Berikut penggunaan sintaks dan hasilnya dapat dilihat pada Gambar 3.14.

```

18 | Yang bersangkutan memiliki blog di [http://bletack.blogspot.com/](http://bletack.blogspot.
19 | com/).

```

• Daftar

Sintaks ini akan digunakan untuk menampilkan daftar catatan. Berikut penggunaan sintaks dan hasilnya dapat dilihat pada Gambar 3.15.

```

23 | * 9 Oktober 2014, bimbingan skripsi
24 | * 3 Oktober 2014, bimbingan skripsi
25 | * 1 September 2014, perwalian
26 | * 1 September 2014, pertama kali dibuat

```

NPM - 2010730013

Gambar 3.10: Output Sintaks Cetak Tebal dan Cetak Miring

Judul 1

Judul 2

Judul 3

Judul 4

Judul 5

Judul 6

Gambar 3.11: Output Sintaks Judul Bab

Baris ini dengan
batas baris

Baris ini tanpa batas baris

Gambar 3.12: Output Sintaks Batas Baris

Samuel adalah seorang mahasiswa yang periang namun terkadang sulit diatur. Dia aktif di himpunan sebagai ketua divisi pelayanan masyarakat.

Grady adalah seorang mahasiswa yang memiliki jiwa pemimpin. Dia aktif di UKM sebagai ketua divisi logistik.

Gambar 3.13: Output Sintaks Paragraf

Yang bersangkutan memiliki blog di <http://bletack.blogspot.com/>.

Gambar 3.14: Output Sintaks Link

- 9 Oktober 2014, bimbingan skripsi
- 3 Oktober 2014, bimbingan skripsi
- 1 September 2014, perwalian
- 1 September 2014, pertama kali dibuat

Gambar 3.15: Output Sintaks Daftar

1 Berikut penggunaan sintaks Markdown secara keseluruhan untuk bagian keterangan ma-
2 hasiswa. Berikut penggunaan sintaks dan hasilnya dapat dilihat pada Gambar

```

3  ### NPM
4
5  2010730013
6
7  ### Nama
8
9  Samuel
10
11 ### Umum
12
13 Samuel adalah seorang mahasiswa yang periang namun terkadang sulit diatur. Dia aktif di himpunan
14 sebagai ketua divisi pelayanan masyarakat. Yang bersangkutan memiliki blog di [http://bletack.
15 blogspot.com/](http://bletack.blogspot.com/).
16
17 ### Catatan
18
19 * 9 Oktober 2014, bimbingan skripsi
20 * 3 Oktober 2014, bimbingan skripsi
21 * 1 September 2014, perwalian
22 * 1 September 2014, pertama kali dibuat

```

NPM

2010730013

Nama

Samuel

Umum

Samuel adalah seorang mahasiswa yang periang namun terkadang sulit diatur. Dia aktif di himpunan sebagai ketua divisi pelayanan masyarakat. Yang bersangkutan memiliki blog di <http://bletack.blogspot.com/>.

Catatan

- 9 Oktober 2014, bimbingan skripsi
- 3 Oktober 2014, bimbingan skripsi
- 1 September 2014, perwalian
- 1 September 2014, pertama kali dibuat

Gambar 3.16: Output Keterangan Mahasiswa

1 3.3 Analisis StrapdownJS

2 StrapdownJS digunakan untuk menampilkan sintaks Markdown ke halaman HTML. Pada
 3 penelitian ini strapdown.js terlebih dahulu diunduh dan untuk menggunakannya menggu-
 4 nakan path seperti di bawah ini.

```
5 | <script src="js/0.2/strapdown.js"></script>
```

6 Skrip tersebut disisipkan pada skrip infomahasiswa.php yang berfungsi untuk menampilkan
 7 info mahasiswa yang dimana info tersebut ditulis menggunakan sintaks Markdown. Berikut
 8 skrip infomahasiswa.php yang menggunakan strapdown.js.

```
9 <!doctype html>
10 <html class="no-js" lang="en">
11   <head>
12     <meta charset="utf-8" />
13     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
14     <title>SIRM | Welcome</title>
15     <link rel="stylesheet" href="css/foundation.css" />
16     <script src="js/vendor/modernizr.js"></script>
17   </head>
18   <body>
19     <div class="row">
20       <h5>Anda melihat catatan mahasiswa ini sebagai test@unpar.ac.id.</h5>
21     </div>
22     <div class="row">
23       <ul class="button-group">
24         <li><a href="editmahasiswa.php" class="button">Edit</a></li>
25         <li><a href="lihathistori.php" class="button">Lihat Histori</a></li>
26       </ul>
27     </div>
28     <hr/>
29 <xmp style="display:none;">
30 ### NPM
31
32 2010730013
33
34 ### Nama
35
36 Samuel
37
38 ### Umum
39
40 Samuel adalah seorang mahasiswa yang periang namun terkadang sulit diatur. Dia aktif di himpunan
41 sebagai ketua divisi pelayanan masyarakat. Yang bersangkutan memiliki blog di [http://bletack.
42 blogspot.com/](http://bletack.blogspot.com/).
43
44 ### Catatan
45
46 * 9 Oktober 2014, bimbingan skripsi
47 * 3 Oktober 2014, bimbingan skripsi
48 * 1 September 2014, perwalian
49 * 1 September 2014, pertama kali dibuat
50
```

```

1 </xmp>
2 <script src="js/0.2/strapdown.js"></script>
3 </body>
4 </html>

```

5 Untuk baris 17 sampai baris 37 akan diambil dari database.

6 3.4 Analisis Zurb Foundation

7 Zurb Foundation digunakan untuk membuat tampilan antarmuka aplikasi yang akan diba-
8 ngun. Sesuai landasan teori pada bab 2, pada aplikasi ini menggunakan tiga bagian yaitu
9 Grid, Tabel dan Tombol. Grid digunakan untuk mengatur pembagian tata letak kompleks
10 sehingga terlihat rapih. Tabel digunakan untuk menampilkan data yang berasal dari da-
11 tabase. Tombol digunakan untuk merubah tombol yang biasa menjadi lebih enak untuk
12 dilihat. Berikut sintaks penggunaan Grid, Tabel, dan Tombol pada pilihmahasiswa.php dan
13 untuk gambar dapat dilihat pada Gambar 3.17.

```

14 <!doctype html>
15 <html class="no-js" lang="en">
16   <head>
17     <meta charset="utf-8" />
18     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
19     <title>SIRM | Welcome</title>
20     <link rel="stylesheet" href="css/foundation.css" />
21     <script src="js/vendor/modernizr.js"></script>
22   </head>
23   <body>
24     <?php
25       session_start();
26     ?>
27     <div class="row">
28       <h3>Pilih NPM yang ingin dicari / tambah baru.</h3>
29       <ul class="button-group">
30         <li><a href="new.php" class="button">Add</a></li>
31         <li><a href="index.php?logout" class="button">Logout</a></li>
32       </ul>
33     </div>
34     <div class="row">
35     <?php
36       include_once "configDatabase.php";
37       $hasil = mysql_query("SELECT * FROM info_mahasiswa", $id_mysql);
38       if(! $hasil){
39         die("Permintaan gagal");
40       }
41
42
43       echo "<table>
44       <thead>
45       <tr>
46         <th width='250'>NPM</th>
47         <th width='500'>Nama</th>
48         <th width='250'>Last Update</th>
49       </tr>
50       </thead>";
51
52       while($row = mysql_fetch_array($hasil))
53       {
54         echo "<tr>";
55         //echo "<td><input class='button1' type='submit' name='submit' value='". $row['npm'] .'></td>";
56         echo "<td><a href='view.php?npm='". $row['npm'] .'>" . $row['npm'] . "</a></td>";
57         echo "<td>" . $row['nama'] . "</td>";
58         echo "<td>" . $row['pemburuan_terakhir'] . "</td>";
59         echo "</tr>";
60       }
61       echo "</table>";
62
63       mysql_close($id_mysql);
64     ?>
65   </div>
66 </body>
67 </html>

```

Pilih NPM yang ingin dicari / tambah baru.

Add	Logout
-----	--------

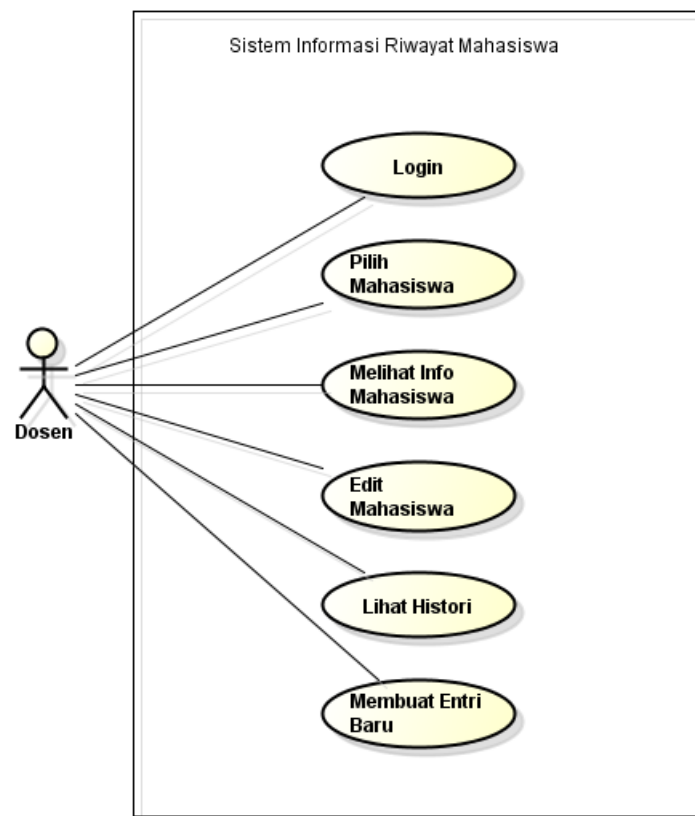
NPM	Nama	Last Update
2010120031	Kenneth Natanael	2015-04-30 14:29:51
2010730001	Andri Agustian	2015-04-29 15:09:33
2010730005	Grady Ireneus	2015-04-30 14:31:19
2010730012	Kevin PL	2015-04-24 00:55:57
2010730013	Samuel Herman	2015-04-24 00:23:34

Gambar 3.17: Tampilan pilihmahasiswa.php Dengan Zurb Foundation

1 3.5 Analisis Kebutuhan Perangkat Lunak

2 Pembahasan use case diagram dan skenario yang akan digunakan pada penelitian.

3 3.5.1 Use Case Diagram



Gambar 3.18: Use Case Diagram Sistem Informasi Riwayat Mahasiswa

4 Use case diagram merupakan pemodelan yang menunjukkan kegiatan apa saja yang dapat
 5 dilakukan pengguna dan kegiatan yang dilakukan sistem. Berikut adalah deskripsi dari use
 6 case pada Gambar 3.18.

- 7 • Login

Tabel 3.1: Skenario Login

Nama	Login		
Aktor	Pengguna		
Deskripsi	Melakukan login via Google OAuth		
Kondisi Awal	Masih berada pada login.php		
Kondisi Akhir	Sudah berada pada pilihmahasiswa.php		
Skenario Utama	No	Aksi Aktor	Reaksi Sistem
	1	Pengguna melakukan login	Server akan mengirimkan pertanyaan untuk izin
	2	Pengguna memberikan izin	Aplilasi mendapatkan otorisasi kode
Eksepsi	Pengguna harus memiliki email yang diakhiri @unpar.ac.id dan username bukan angka semua		

1 Use case ini memungkinkan pengguna untuk login via Google OAuth.

2 • Pilih Mahasiswa

3 Use case ini memungkinkan pengguna untuk memilih mahasiswa yang ingin dilihat
4 infonya. Selain itu pengguna juga bisa menekan tombol "Add" untuk menambah
5 entri baru.

6 • Lihat Info Mahasiswa

7 Use case ini memungkinkan pengguna untuk melihat info mahasiswa. Selain itu peng-
8 guna bisa menekan tombol "Edit" untuk mengedit info mahasiswa dan pengguna juga
9 bisa menekan tombol "Lihat Histori" untuk melihat histori.

10 • Edit Mahasiswa

11 Use case ini memungkinkan pengguna untuk mengedit info mahasiswa yang sudah ada.

12 • Lihat Histori

13 Use case ini memungkinkan pengguna untuk melihat histori untuk setiap perubahan
14 dan aksi view.

15 • Membuat Entri Baru

16 Use case ini memungkinkan pengguna untuk membuat entri baru dengan memasukan
17 inputan pada form yang telah disediakan.

18 3.5.2 Skenario

19 3.5.2.1 Login

20 Untuk use case Login, skenarionya dapat dilihat pada Tabel 3.1.

21 3.5.2.2 Pilih Mahasiswa

22 Untuk use case Pilih Mahasiswa, skenarionya dapat dilihat pada Tabel 3.2.

Tabel 3.2: Skenario Pilih Mahasiswa

Nama	Pilih Mahasiswa		
Aktor	Pengguna		
Deskripsi	Pengguna dapat memilih dan mencari mahasiswa berdasarkan NPM		
Kondisi Awal	Sebuah form dengan tabel yang berisi data mahasiswa		
Kondisi Akhir	Salah satu mahasiswa terpilih		
Skenario Utama	No	Aksi Aktor	Reaksi Sistem
	1	Pengguna mencari mahasiswa berdasarkan NPM	Sistem seleksi mahasiswa berdasarkan NPM
	2	Pengguna mengklik NPM mahasiswa yang dipilih	Pindah ke halaman info-mahasiswa.php
Eksepsi	-		

Tabel 3.3: Skenario Melihat Info Mahasiswa

Nama	Melihat Info Mahasiswa		
Aktor	Pengguna		
Deskripsi	Melihat info mahasiswa yang telah dipilih pada pilihmahasiswa.php		
Kondisi Awal	Menampilkan info yang dimiliki mahasiswa		
Kondisi Akhir	Jika pengguna mengklik "Edit" maka pindah ke editmahasiswa.php. Jika pengguna mengklik "Lihat Histori" maka pindah ke lihathistori.php		
Skenario Utama	No	Aksi Aktor	Reaksi Sistem
	1	Pengguna melihat info mahasiswa	Sistem menampilkan info mahasiswa
Eksepsi	-		

1 3.5.2.3 Melihat Info Mahasiswa

2 Untuk use case Melihat Info Mahasiswa, skenarionya dapat dilihat pada Tabel 3.3.

3 3.5.2.4 Edit Mahasiswa

4 Untuk use case Edit Mahasiswa, skenarionya dapat dilihat pada Tabel 3.4.

5 3.5.2.5 Lihat Histori

6 Untuk use case Lihat Histori, skenarionya dapat dilihat pada Tabel 3.5.

7 3.5.2.6 Membuat Entri Baru

8 Untuk use case Membuat Entri Baru, skenarionya dapat dilihat pada Tabel 3.6.

Tabel 3.4: Skenario Edit Mahasiswa

Nama	Edit Mahasiswa		
Aktor	Pengguna		
Deskripsi	Mengedit info mahasiswa yang sudah ada di database		
Kondisi Awal	Menampilkan form dengan data yang sudah ada pada database		
Kondisi Akhir	Form dengan data yang telah diedit		
Skenario Utama	No	Aksi Aktor	Reaksi Sistem
	1	Pengguna mengedit data yang sudah ada	Sistem menampilkan data yang sudah ada
	2	Pengguna menyimpan perubahan	Sistem akan merekan perubahan ke dalam database
Eksepsi	-		

Tabel 3.5: Skenario Lihat Histori

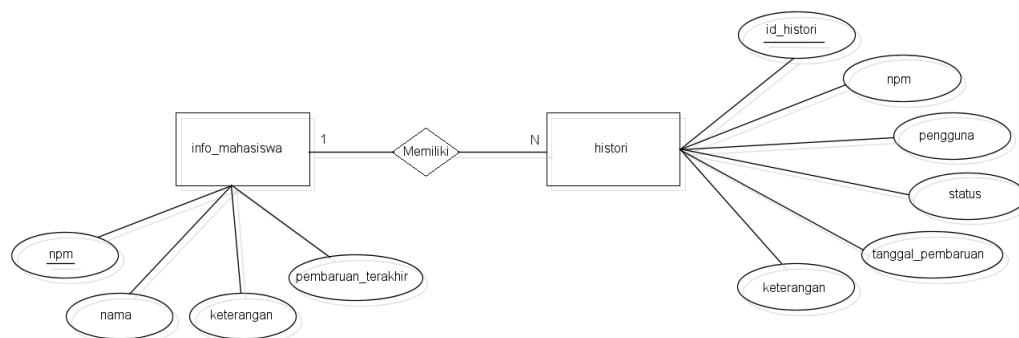
Nama	Lihat Histori		
Aktor	Pengguna		
Deskripsi	Melihat histori perubahan dan aksi melihat yang dilakukan pengguna		
Kondisi Awal	Menampilkan log histori perubahan dan aksi melihat		
Kondisi Akhir	Terus bertambah sesuai aksi yang dilakukan		
Skenario Utama	No	Aksi Aktor	Reaksi Sistem
	1	Pengguna melihat log histori	Sistem akan menampilkan log histori
Eksepsi	-		

Tabel 3.6: Skenario Membuat Entri Baru

Nama	Membuat Entri Baru		
Aktor	Pengguna		
Deskripsi	Membuat entri baru yang belum ada pada database		
Kondisi Awal	Menampilkan form untuk menambah entri baru		
Kondisi Akhir	Input pada form akan dimasukan kedalam database		
Skenario Utama	No	Aksi Aktor	Reaksi Sistem
	1	Pengguna mengisi form entri baru	Sistem menampilkan form entri baru
	2	Pengguna menyimpan inputan dari form entri baru	Sistem akan merekan inputan pengguna ke dalam database
Eksepsi	-		

3.5.3 Entity Relationship Diagram (ERD)

Entity Relationship Diagram Sistem Informasi Riwayat Mahasiswa dapat dilihat pada Gambar 3.19.



Gambar 3.19: Entity Relationship Diagram

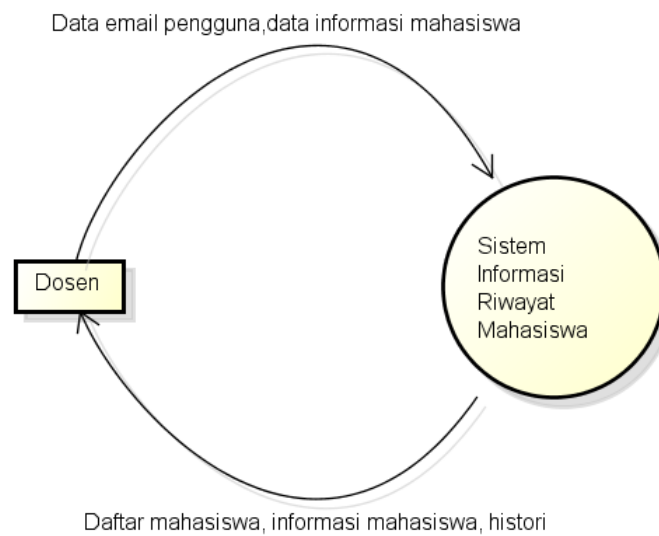
Pada ERD Sistem Informasi Riwayat Mahasiswa terdapat dua entitas yaitu info mahasiswa dan histori. Entitas info mahasiswa dengan relasi memiliki terbuang dengan entitas histori.

3.5.4 Data Flow Diagram

Data Flow Diagram Sistem Informasi Riwayat Mahasiswa digambarkan dalam dua level, yaitu Data Context Diagram / Data Flow Diagram Level 0, Data Flow Diagram Level 1.

1 Data Context Diagram

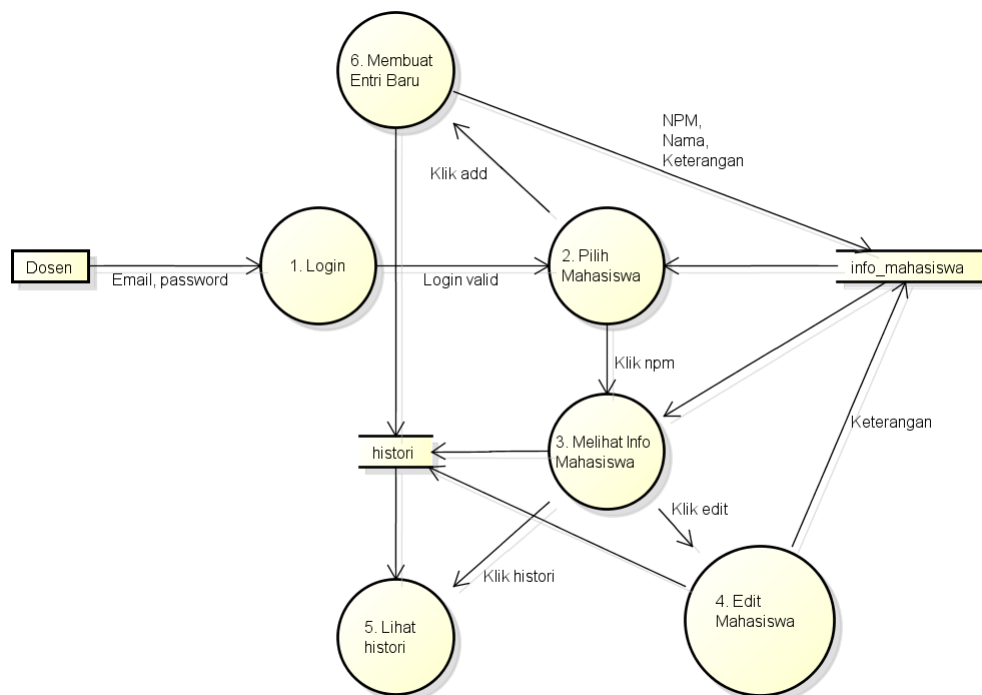
Data Context Diagram Sistem Informasi Riwayat Mahasiswa dapat dilihat pada Gambar 3.20.



Gambar 3.20: Data Context Diagram Sistem Informasi Riwayat Mahasiswa

2 Data Flow Diagram Level 1

Data Flow Diagram Level 1 Sistem Informasi Riwayat Mahasiswa dapat dilihat pada Gambar 3.21.



Gambar 3.21: Data Flow Diagram Sistem Informasi Riwayat Mahasiswa

Untuk spesifikasi proses dapat dilihat di bawah ini.

- 1 • No dan Nama Proses : P1 Login
2 Deskripsi : Proses akses ke perangkat lunak
3 Data Input : Email, password
4 Data Output : Login valid
5 Proses : Mengecek email yang digunakan diakhiri @unpar.ac.id atau tidak. Jika ya
6 maka sistem akan mengarahkan ke halaman utama akan tetapi jika tidak maka
7 sistem akan menampilkan pesan login gagal.
- 8 • No dan Nama Proses : P2 Pilih Mahasiswa
9 Deskripsi : Proses memilih mahasiswa
10 Data Input : -
11 Data Output : Tabel daftar mahasiswa
12 Proses : pengguna memilih mahasiswa yang ingin dipilih dari tabel daftar mahasiswa.
- 13 • No dan Nama Proses : P3 Melihat Info Mahasiswa
14 Deskripsi : Proses melihat info mahasiswa
15 Data Input : -
16 Data Output : Keterangan mahasiswa
17 Proses : pengguna mendapatkan info mahasiswa dari mahasiswa yang telah pengguna
18 pilih pada proses 2. Proses ini dicatat pada histori sebagai aksi melihat.
- 19 • No dan Nama Proses : P4 Edit Mahasiswa
20 Deskripsi : Proses mengedit info mahasiswa
21 Data Input : Keterangan mahasiswa terbaru
22 Data Output : -
23 Proses : pengguna memperbaharui keterangan yang dimiliki mahasiswa yang telah
24 pengguna pilih pada proses 2. Proses ini dicatat pada histori sebagai aksi mengedit.
- 25 • No dan Nama Proses : P5 Lihat Histori
26 Deskripsi : Proses melihat histori
27 Data Input : -
28 Data Output : Daftar histori
29 Proses : pengguna mendapatkan histori dari mahasiswa yang telah pengguna pilih
30 pada proses 2. Pengguna juga dapat melihat informasi mahasiswa versi sebelumnya.
- 31 • No dan Nama Proses : P6 Membuat Entri Baru
32 Deskripsi : Proses membuat entri baru
33 Data Input : NPM, Nama, Keterangan
34 Data Output : -
35 Proses : pengguna memasukan data NPM, Nama, Keterangan untuk entri baru lalu
36 menyimpannya. Proses ini dicatat pada histori sebagai aksi membuat entri baru.

BAB 4

PERANCANGAN

Pada bab ini akan dijelaskan mengenai perancangan Sistem Informasi Riwayat Mahasiswa yang akan dibuat. Mulai dari perancangan tampilan web yang digunakan, perancangan modul, dan perancangan diagram sekuens.

4.1 Perancangan Tampilan Web Yang Digunakan

Perancangan tampilan web yang akan dibuat untuk mengimplementasikan Sistem Informasi Riwayat Mahasiswa terdapat tujuh buah perancangan yaitu halaman awal, pilih mahasiswa, info mahasiswa, edit mahasiswa, lihat histori, lihat versi ini dan entri baru.

4.1.1 Tampilan Halaman Awal

Perancangan tampilan web untuk halaman utama dapat dilihat pada Gambar 4.2.



Gambar 4.1: Desain Antarmuka Halaman Awal

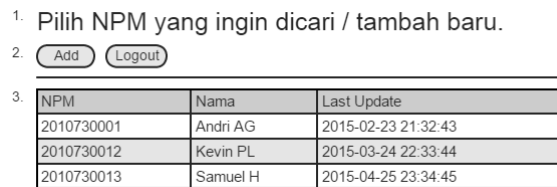
Keterangan :

(1) Bagian ini merupakan judul yang merupakan keterangan dari perangkat lunak.

(2) Bagian ini merupakan teks yang dapat diklik untuk melakukan login.

4.1.2 Tampilan Web Pilih Mahasiswa

Perancangan tampilan web untuk pilih mahasiswa dapat dilihat pada Gambar 4.2.



Gambar 4.2: Desain Antarmuka Pilih Mahasiswa

1 Keterangan :

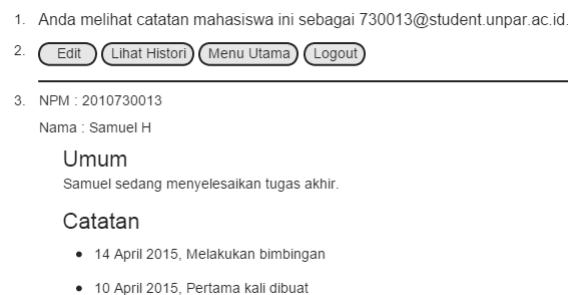
2 (1) Bagian ini merupakan judul dari halaman untuk memilih mahasiswa.

3 (2) Bagian ini merupakan tombol untuk melakukan aksi add atau logout.

4 (3) Bagian ini merupakan tempat menampilkan data mahasiswa dalam bentuk tabel. NPM
5 dapat diklik untuk memilih mahasiswa.

6 4.1.3 Tampilan Web Info Mahasiswa

7 Perancangan tampilan web untuk info mahasiswa dapat dilihat pada Gambar 4.3.



Gambar 4.3: Desain Antarmuka Info Mahasiswa

8 Keterangan :

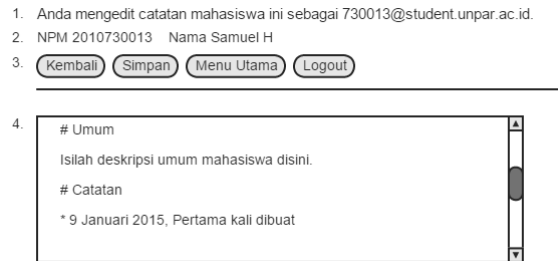
9 (1) Bagian ini merupakan teks yang menampilkan keterangan dan juga pengguna yang
10 sedang menggunakan Sistem Infomasi Riwayat Mahasiswa.

11 (2) Bagian ini merupakan tombol untuk melakukan aksi edit, lihat histori, pindah ke menu
12 utama, dan logout.

13 (3) Bagian ini merupakan tempat menampilkan info mahasiswa yang berasal dari database.

14 4.1.4 Tampilan Web Edit Mahasiswa

15 Perancangan tampilan web untuk edit mahasiswa dapat dilihat pada Gambar 4.4.



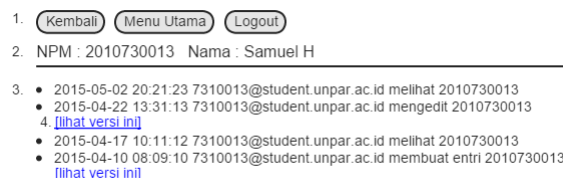
Gambar 4.4: Desain Antarmuka Edit Mahasiswa

1 Keterangan :

- 2 (1) Bagian ini merupakan teks yang menampilkan keterangan dan juga pengguna yang
3 sedang menggunakan Sistem Infomasi Riwayat Mahasiswa.
- 4 (2) Bagian ini merupakan teks yang menampilkan NPM dan nama mahasiswa yang telah
5 dipilih untuk diedit.
- 6 (3) Bagian ini merupakan tombol untuk melakukan aksi kembali, simpan untuk perubahan
7 yang telah dilakukan, pindah ke menu utama, dan logout.
- 8 (4) Bagian ini merupakan tempat menampilkan catatan mahasiswa yang berasal dari data-
9 base dan dapat diedit (ditulis dengan format markdown).

10 4.1.5 Tampilan Web Lihat Histori

11 Perancangan tampilan web untuk lihat histori dapat dilihat pada Gambar 4.5.



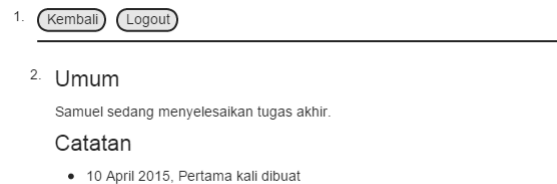
Gambar 4.5: Desain Antarmuka Lihat Histori

12 Keterangan :

- 13 (1) Bagian ini merupakan tombol untuk melakukan aksi kembali, pindah ke menu utama,
14 dan logout.
- 15 (2) Bagian ini merupakan teks yang menampilkan keterangan NPM dan nama mahasiswa
16 yang telah dipilih untuk dilihat historinya.
- 17 (3) Bagian ini merupakan daftar histori dari mahasiswa yang telah dipilih.

18 4.1.6 Tampilan Web Lihat Versi Ini

19 Perancangan tampilan web untuk lihat versi ini dapat dilihat pada Gambar 4.6.



Gambar 4.6: Desain Antarmuka Lihat Versi Ini

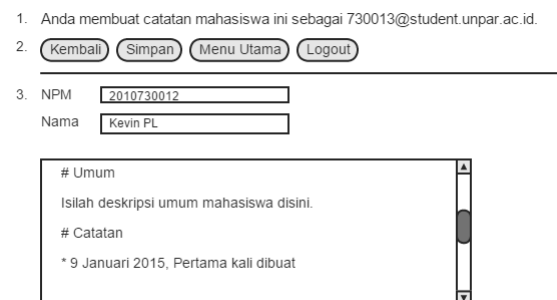
Keterangan :

(1) Bagian ini merupakan tombol untuk melakukan aksi kembali dan logout.

(2) Bagian ini merupakan daftar histori dari mahasiswa yang telah dipilih.

4.1.7 Tampilan Web Entri Baru

Perancangan tampilan web untuk entri baru dapat dilihat pada Gambar 4.7.



Gambar 4.7: Desain Antarmuka Pilih Entri Baru

Keterangan :

(1) Bagian ini merupakan teks yang menampilkan keterangan dan juga pengguna yang sedang menggunakan Sistem Infomasi Riwayat Mahasiswa.

(2) Bagian ini merupakan tombol untuk melakukan aksi kembali ke pilih mahasiswa, simpan, menu utama, dan logout.

(3) Bagian ini merupakan tombol untuk melakukan aksi simpan untuk penambahan entri baru.

(4) Bagian ini merupakan form yang terdiri dari area untuk memasukkan NPM mahasiswa, nama mahasiswa, dan keterangan mahasiswa yang akan ditambah dengan format yang telah disediakan (ditulis dengan format markdown).

4.2 Perancangan Modul

Perancangan modul untuk Sistem Informasi Riwayat Mahasiswa yang akan dibuat dapat dilihat pada sub bab berikut.

1 4.2.1 Modul Login

2 Modul login yang dilakukan oleh user (dosen) dapat dilihat pada Tabel 4.1.

Tabel 4.1: Modul Login

Nama Modul	login.php
Input	username, password
Output	-
Tabel yang diakses	User
Deskripsi	User memasukkan username dan password kemudian sistem akan melakukan autentikasi menggunakan Google OAuth.

3 4.2.2 Modul Pilih Mahasiswa

4 Modul pilih mahasiswa yang dilakukan oleh user (dosen) dapat dilihat pada Tabel 4.2.

Tabel 4.2: Modul Pilih Mahasiswa

Nama Modul	pilihmahasiswa.php
Input	npm
Output	Tabel mahasiswa
Tabel yang diakses	Mahasiswa
Deskripsi	User memasukkan npm yang ingin dicari, user juga dapat memilih langsung mahasiswa yang ingin dilihat infonya tanpa melakukan pencarian terlebih dahulu dan user juga dapat membaut entri baru.

5 4.2.3 Modul Info Mahasiswa

6 Modul info mahasiswa yang dilakukan oleh user (dosen) dapat dilihat pada Tabel 4.3.

Tabel 4.3: Modul Info Mahasiswa

Nama Modul	infomahasiswa.php
Input	-
Output	Info mahasiswa
Tabel yang diakses	Mahasiswa
Deskripsi	User mendapatkan laporan berupa info mahasiswa yang telah dipilih sebelumnya pada modul pilih mahasiswa. User dapat merubah info mahasiswa yang ada dan dapat melihat histori setiap mahasiswa.

7 4.2.4 Modul Edit Mahasiswa

8 Modul edit mahasiswa yang dilakukan oleh user (dosen) dapat dilihat pada Tabel 4.4.

9 4.2.5 Modul Lihat Histori

10 Modul lihat histori yang dilakukan oleh user (dosen) dapat dilihat pada Tabel 4.5.

Tabel 4.4: Modul Edit Mahasiswa

Nama Modul	editmahasiswa.php
Input	teks dalam format markdown
Output	-
Tabel yang diakses	Mahasiswa
Deskripsi	User memasukkan atau merubah keterangan mahasiswa pada teks area yang telah disediakan menggunakan teks dengan format penulisan markdown lalu user menyimpan untuk menaruh perubahan yang dilakukan. User dapat kembali ke modul info mahasiswa tanpa melakukan perubahan.

Tabel 4.5: Modul Lihat Histori

Nama Modul	lihathistori.php
Input	-
Output	Daftar histori mahasiswa
Tabel yang diakses	Mahasiswa
Deskripsi	User mendapatkan laporan berupa daftar histori yang dimiliki setiap mahasiswa.

1 4.2.6 Modul Entri Baru

- 2 Modul entri baru yang dilakukan oleh user (dosen) dapat dilihat pada Tabel 4.6.

Tabel 4.6: Modul Entri Baru

Nama Modul	entribaru.php
Input	npm, nama, dan teks dalam format markdown
Output	-
Tabel yang diakses	Mahasiswa
Deskripsi	User memasukkan npm, nama, dan keterangan mahasiswa pada teks area yang telah disediakan menggunakan teks dengan format penulisan markdown lalu user menyimpan untuk membuat entri baru tersebut. User dapat kembali ke modul pilih mahasiswa tanpa melakukan perubahan.

3 4.3 Perancangan Tabel Sistem Informasi Riwayat Mahasiswa

4 4.3.1 Perancangan Tabel Info Mahasiswa

- 5 Untuk rancangan tabel info mahasiswa dapat dilihat pada Tabel 4.7.

6 4.3.2 Perancangan Tabel Histori

- 7 Untuk rancangan tabel histori dapat dilihat pada Tabel 4.8.

Tabel 4.7: Rancangan Tabel Info Mahasiswa

Atribut	Tipe Data	Ukuran	Primary Key	Foreign Key	Null	Keterangan
npm	varchar	10	yes	no	no	-
nama	varchar	60	no	no	no	-
keterangan	text	-	no	no	no	-
pembaruan_terakhir	datetime	-	no	no	no	-

Tabel 4.8: Rancangan Tabel Histori

Atribut	Tipe Data	Ukuran	Primary Key	Foreign Key	Null	Keterangan
id_histori	int	5	yes	no	no	AUTO_INCREMENT
npm	varchar	10	no	no	no	-
pengguna	varchar	60	no	no	no	-
status	text	-	no	no	no	-
tanggal_pembaruan	datetime	-	no	no	no	-
keterangan	text	-	no	no	no	-

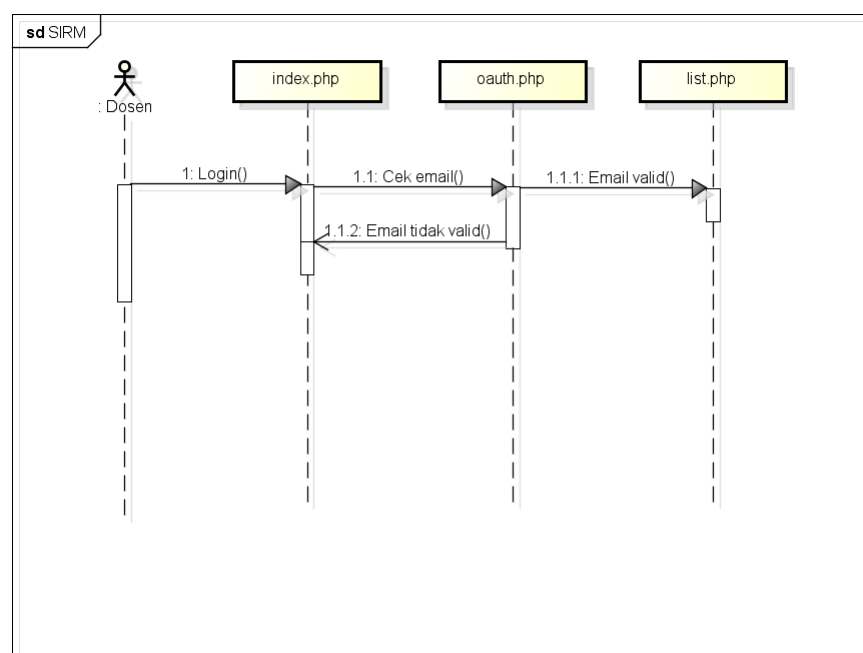
1 4.4 Diagram Sekuens

2 Pembuatan diagram sekuens mengacu pada Gambar 3.18. Terdapat tiga diagram sekuens
3 yaitu

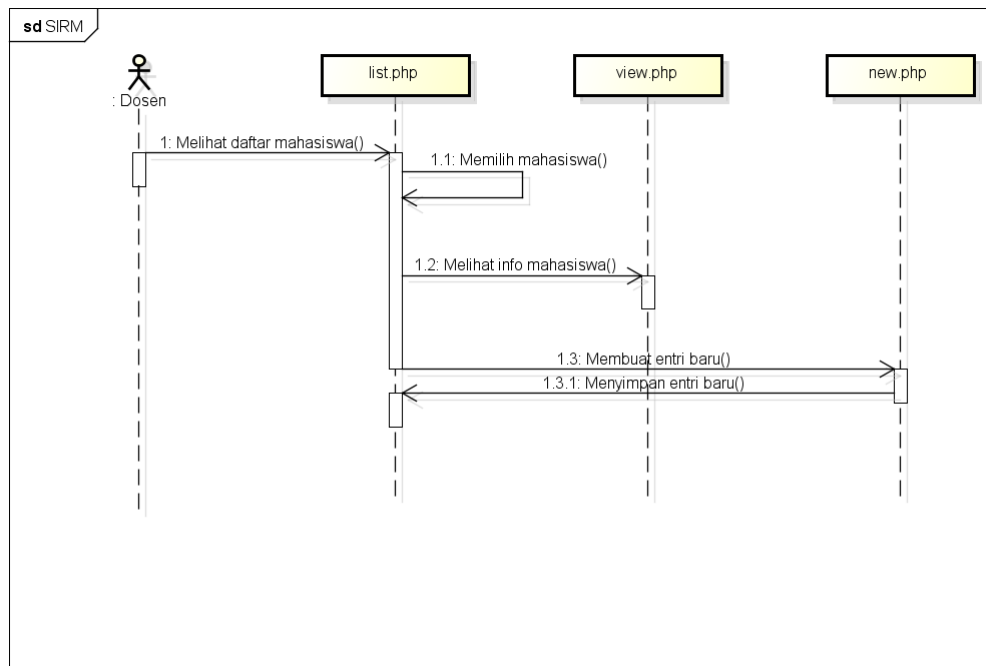
4 1 Sekuens bagian satu mencakup proses login dapat dilihat pada Gambar 4.8.

5 2 Sekuens bagian dua mencakup proses memilih mahasiswa, melihat info mahasiswa, dan
6 membuat entri baru. Dapat dilihat pada Gambar 4.9.

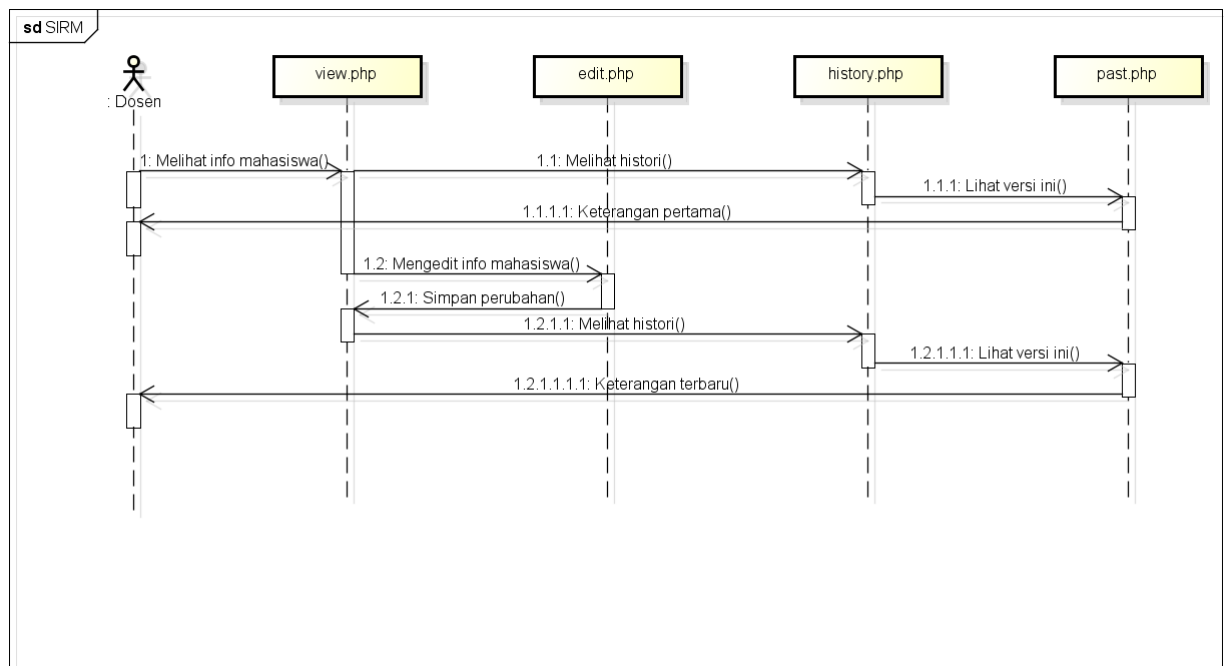
7 3 Sekuens bagian tiga mencakup melihat info mahasiswa, mengedit info mahasiswa, melihat
8 histori, dan melihat keterangan versi ini. Dapat dilihat pada Gambar 4.10.



Gambar 4.8: Diagram Sekuens Bagian Satu



Gambar 4.9: Diagram Sekuens Bagian Dua



Gambar 4.10: Diagram Sekuens Bagian Tiga

BAB 5

IMPLEMENTASI DAN PENGUJIAN PERANGKAT LUNAK

Bab ini terdiri atas dua bagian, yaitu Implementasi Perangkat Lunak dan Pengujian Perangkat Lunak. Bagian implementasi berisi penjelasan lingkungan pengembangan perangkat lunak. Sedangkan bagian pengujian berisi hasil pengujian terhadap perangkat lunak yang telah dibangun.

5.1 Implementasi Perangkat Lunak

Pada bagian ini akan dibahas hasil implementasi perangkat lunak yang telah dibangun. Subbab ini terdiri atas tiga bagian, yaitu lingkungan perangkat keras, lingkungan perangkat lunak, dan hasil implementasi perangkat lunak.

5.1.1 Lingkungan Implementasi Perangkat Keras

Dalam membangun perangkat lunak ini digunakan spesifikasi perangkat keras sebagai berikut:

- (a) Processor: AMD A10-5750M 2.5GHz
- (b) RAM: 4 GB DDR3
- (c) Harddisk: 1TB
- (d) VGA: AMD Radeon HD 8650G 2GB
- (e) Koneksi Internet: WAN

5.1.2 Lingkungan Implementasi Perangkat Lunak

Dalam membangun perangkat lunak ini digunakan spesifikasi perangkat lunak sebagai berikut:

- (a) Sistem Operasi: Windows 8.1 Pro 64-bit
- (b) Bahasa Pemrograman: PHP Version 5.6.3
- (c) Aplikasi: XAMPP v5.6.3
- (d) Aplikasi web browser: Google Chrome

- 1 (e) Library: Google APIs Client Library untuk PHP
- 2 (f) Javascript: Strapdown.js
- 3 (g) Framework: Foundation

4 **5.1.3 Hasil Implementasi Perangkat Lunak**

5 Kode program perangkat lunak ditulis berdasarkan perancangan yang telah dibahas pada
6 bab 4. Kode program untuk membangun menggunakan PHP dengan library Google APIs
7 Client.

8 **5.2 Pengujian Perangkat Lunak**

9 Pada bagian ini akan dibahas mengenai pengujian yang akan dilakukan terhadap perangkat
10 lunak. Pengujian tersebut terdiri dari dua bagian yaitu pengujian fungsional dan penguji-
11 an eksperimental. Pengujian fungsional bertujuan untuk memastikan bahwa seluruh fungsi
12 yang dibangun pada perangkat lunak berjalan sesuai dengan yang direncanakan. Sedangkan
13 pengujian eksperimental bertujuan untuk menguji eksepsi-eksepsi yang terdapat pada per-
14angkat lunak. Pada bagian ini juga terdapat perubahan program pada bagian oauth.php,
15 jadi dapat menjalankan pengujian dengan email yang diakhiri @student.unpar.ac.id dikare-
16 nakan penulis tidak memiliki email yang diakhiri @unpar.ac.id.

17 **5.2.1 Lingkungan Pengujian Perangkat Keras**

18 Dalam pengujian perangkat lunak ini digunakan spesifikasi perangkat keras sebagai berikut:

- 19 (a) Processor: AMD A10-5750M 2.5GHz
- 20 (b) RAM: 4 GB DDR3
- 21 (c) Harddisk: 1TB
- 22 (d) VGA: AMD Radeon HD 8650G 2GB
- 23 (e) Koneksi Internet: WAN

24 **5.2.2 Lingkungan Pengujian Perangkat Lunak**

25 Dalam pengujian perangkat lunak ini digunakan spesifikasi perangkat lunak sebagai berikut:

- 26 (a) Sistem Operasi: Windows 8.1 Pro 64-bit
- 27 (b) Bahasa Pemrograman: PHP Version 5.6.3
- 28 (c) Aplikasi: XAMPP v5.6.3
- 29 (d) Aplikasi web browser: Google Chrome
- 30 (e) Library: Google APIs Client Library untuk PHP
- 31 (f) Javascript: Strapdown.js
- 32 (g) Framework: Foundation

1 5.2.3 Pengujian Fungsional

2 Pengujian fungsional menguji tampilan antar muka perangkat lunak beserta fungsi dasar.
3 Berikut ini adalah daftar pengujian yang dilakukan:

4 (1) Fungsi login

5 Pengujian fungsi ini dilakukan untuk memastikan perangkat lunak terhubung ke se-
6 rver google untuk melakukan otentikasi dan otorisasi serta memeriksa apakah email
7 yang digunakan untuk login diakhiri "@unpar.ac.id" atau "@student.unpar.ac.id" di-
8 karenakan penulis tidak mempunyai akun dosen. Contoh kasus adalah melakukan lo-
9 gin sebanyak dua kali, yang pertama menggunakan email yang diakhiri "@unpar.ac.id"
10 atau "@student.unpar.ac.id", dan yang kedua menggunakan email yang diakhiri selain
11 "@unpar.ac.id" dan "@student.unpar.ac.id". Pengujian pertama pengguna membuka
12 halaman index.php dapat dilihat pada Gambar 5.1. Lalu pengguna melakukan login
13 menggunakan email "7310013@sudent.unpar.ac.id" dapat dilihat pada Gambar 5.2. La-
14 lu akan ada konfirmasi bahwa akun yang digunakan dikelola oleh student.unpar.ac.id
15 dapat dilihat pada Gambar 5.3. Lalu pengguna akan diarahkan ke CAS (Central Au-
16 thentication Service) UNPAR dan melakukan login kembali dapat dilihat pada Gambar
17 5.4. Lalu pengguna akan diminta untuk memberikan izin akses dari pihak penggu-
18 na dapat dilihat pada Gambar 5.5. Setelah pengguna memberikan izin akses maka
19 akan dilakukan dengan fungsi memilih mahasiswa yang akan dibahas pada poin ber-
20 ikutnya. Sedangkan pengujian kedua pengguna melakukan login menggunakan email
21 "bletask@gmail.com" dapat dilihat pada Gambar 5.6. Lalu pengguna akan mendapat
22 alert karena email yang digunakan tidak sesuai dengan ketentuan dapat dilihat pada
23 Gambar 5.7. Setelah pengguna menekan tombol ok pada alert maka pengguna akan
24 dikembalikan ke halaman index.php. Hal ini menunjukkan fungsi login sudah berjalan
25 dengan baik.

26 (2) Fungsi memilih mahasiswa

27 Pengujian fungsi ini dilakukan untuk memastikan pengguna dapat memilih mahasis-
28 wa. Pada halaman list.php terdapat tabel yang berisikan npm, nama, dan last update
29 dan pengguna dapat memilih mahasiswa dengan menekan npm yang diinginkan. Con-
30 toh pengujian pengguna akan memilih mahasiswa dengan npm 2010730013 maka akan
31 menghasilkan link yang mengarah ke *view.php?npm = 2010730013* dapat dilihat pada
32 Gambar 5.8. Hal ini menunjukkan fungsi memilih mahasiswa sudah berjalan dengan
33 baik.

34 (3) Fungsi melihat info mahasiswa

35 Pengujian fungsi ini dilakukan untuk memastikan mahasiswa yang telah dipilih oleh
36 pengguna, dapat dilihat info dari mahasiswa tersebut. Contoh pengujian fungsi me-
37 rupakan lanjutan dari fungsi memilih mahasiswa, dimana setelah pengguna memilih
38 mahasiswa pada list.php maka akan menampilkan info dari mahasiswa tersebut dapat
39 dilihat pada Gambar 5.9. Hal ini menunjukkan fungsi melihat info mahasiswa sudah
40 berjalan dengan baik.

41 (4) Fungsi mengedit info mahasiswa

42 Pengujian fungsi ini dilakukan untuk memastikan info mahasiswa dapat diedit. Con-

1 toh pengujian mengambil info dari mahasiswa yang telah dilihat infonya pada fungsi
2 melihat info mahasiswa. Dimana keterangan info mahasiswa yang ada akan ditampilk-
3 an dan pengguna dapat mengedit lalu menyimpan perubahan dengan menekan tombol
4 "Simpan" dapat dilihat pada Gambar 5.10. Setelah menyimpan perubahan pengguna
5 akan dibawa kembali ke halaman list.php. Hal ini menunjukkan fungsi mengedit info
6 mahasiswa sudah berjalan dengan baik.

7 (5) Fungsi melihat histori

8 Pengujian fungsi ini dilakukan untuk memastikan adanya histori dari mahasiswa yang
9 dipilih lalu dapat melihat versi keterangan yang pertama kali dibuat dan versi-versi ber-
10 ikutnya yang sudah dirubah. Contoh pengujian melihat histori dari mahasiswa dengan
11 npm 2010730013 dapat dilihat pada Gambar 5.11 dan juga melihat keterangan versi
12 pertama berserta versi berikutnya dapat dilihat pada Gambar 5.12 dan Gambar 5.13.
13 Hal ini menunjukkan fungsi melihat histori sudah berjalan dengan baik.

14 (6) Fungsi membuat entri baru

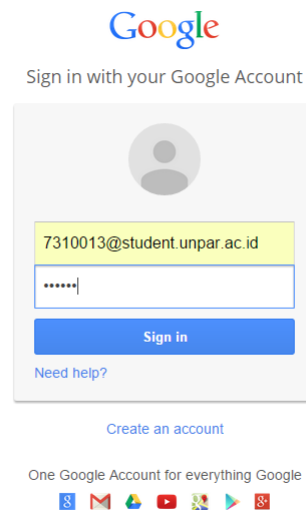
15 Pengujian fungsi ini dilakukan untuk memastikan pada saat membuat entri baru terda-
16 pat template markdown dan berhasil menyimpan entri baru tersebut. Contoh pengujian
17 menambahkan entri baru dengan npm 2010730012, nama Kevin PL, dan keterangan se-
18 suai template. Terdapat template markdown pada saat membuka halaman new.php
19 dapat dilihat pada Gambar 5.14. Mengisi data npm 2010730012 dan nama Kevin PL
20 dapat dilihat pada Gambar 5.15. Setelah pengguna menekan tombol "Simpan" maka
21 data yang telah dimasukan akan tersimpan dan pengguna akan dikembalikan ke halaman
22 list.php. Pengguna dapat melihat entri baru dengan npm 2010730012 dan nama Kevin
23 PL telah masuk kedalam tabel dapat dilihat pada Gambar 5.16. Hal ini menunjukkan
24 fungsi membuat entri baru sudah berjalan dengan baik.

WELCOME to SIRM

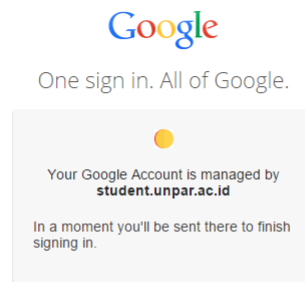
(Sistem Informasi Riwayat Mahasiswa)

[Login with Google](#)

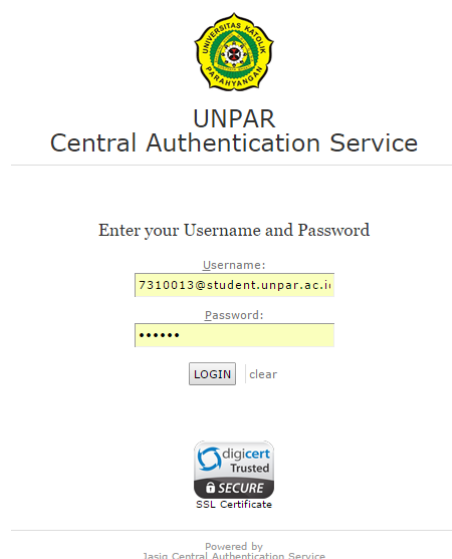
Gambar 5.1: Membuka Halaman index.php



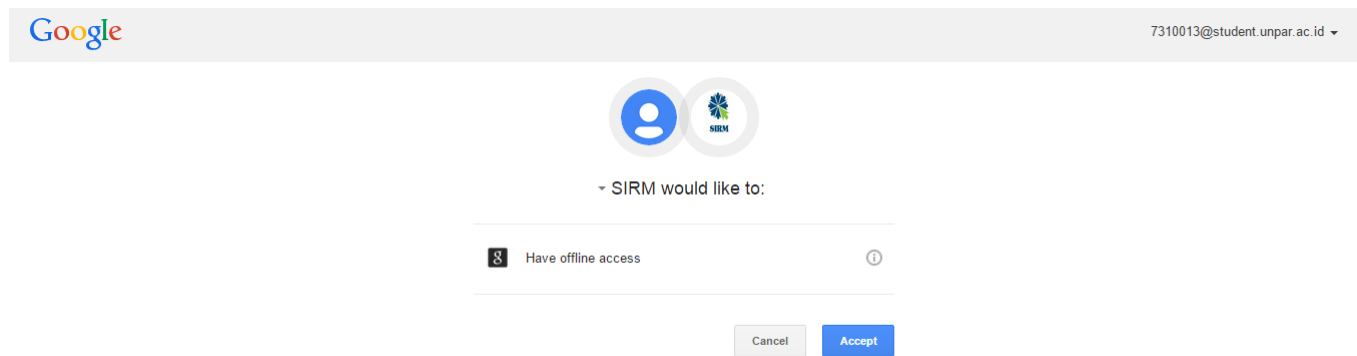
Gambar 5.2: Login Dengan Email yang Diakhiri "@student.unpar.ac.id"



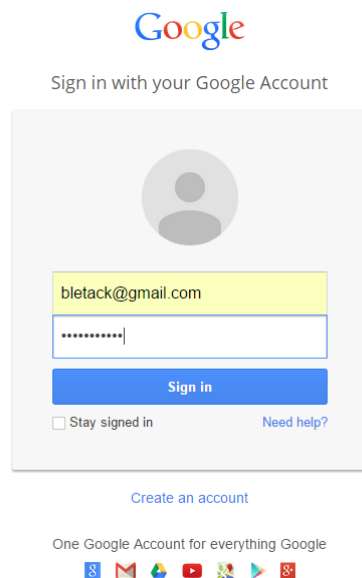
Gambar 5.3: Konfirmasi Email yang Dikelola oleh student.unpar.ac.id



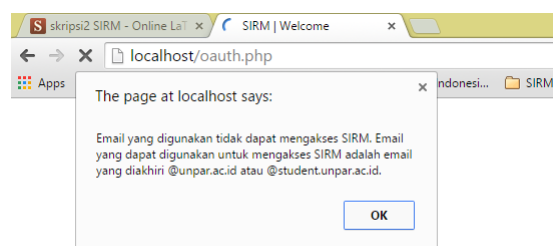
Gambar 5.4: CAS UNPAR



Gambar 5.5: Izin Akses Dari Pihak Pengguna



Gambar 5.6: Login Dengan Email yang Diakhiri "@gmail.com"



Gambar 5.7: Alert Email yang Digunakan Tidak Dapat Mengakses SIRM

Pilih NPM yang ingin dicari / tambah baru.

Add Logout

NPM	Nama	Last Update
2003730013	Pascal	2015-05-05 09:26:15
2010120031	Kenneth Natanael	2015-04-30 14:29:51
2010730001	Andri Agustian	2015-04-29 15:09:33
2010730005	Grady Ireneus	2015-04-30 14:31:19
2010730012	Kevin PL	2015-04-24 00:55:57
2010730013	Samuel Herman	2015-04-24 00:23:34
2010730039	henry	2015-04-29 18:11:20
2010730040	david	2015-04-29 18:10:12
2010730041	dewi	2015-04-29 14:16:08
2010730048	James tjandra	2015-04-29 15:12:03
2010730071	Andreas	2015-04-29 15:25:55
localhost/view.php?npm=2010730013	ryner	2015-04-28 18:08:41

Gambar 5.8: Memilih Mahasiswa

Anda melihat catatan mahasiswa ini sebagai 7310013@student.unpar.ac.id.

Edit	Lihat Histori	Menu Utama	Logout
------	---------------	------------	--------

NPM : 2010730013
 Nama : Samuel Herman

Umum

Samuel sedang mengambil tugas akhir dan sekarang sedang melakukan tahap *finishing*

Catatan

- 22 April 2015, melakukan perwalian ke-2
- 21 April 2015, pertama kali dibuat

Gambar 5.9: Melihat Info Mahasiswa

Anda mengedit catatan mahasiswa ini sebagai 7310013@student.unpar.ac.id.

NPM 2010730013 Nama Samuel Herman

Kembali	Simpan	Menu Utama	Logout
---------	--------	------------	--------

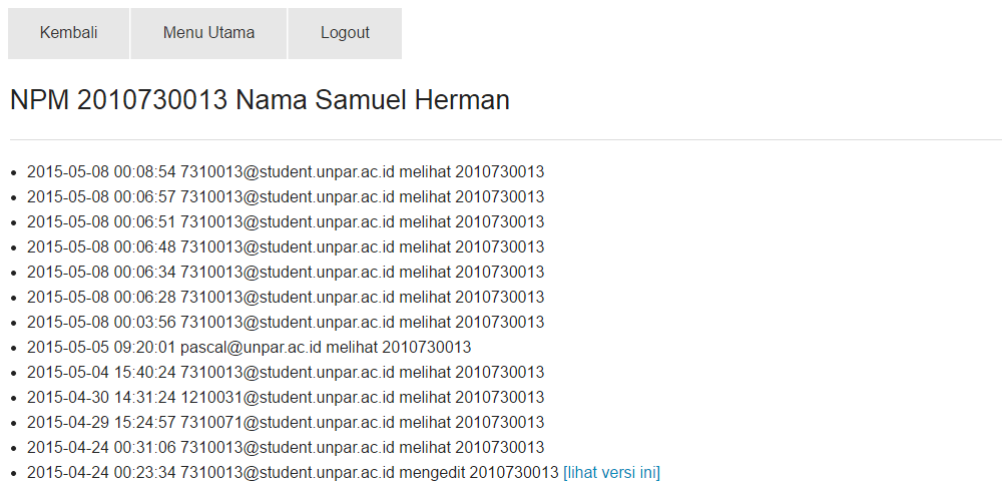
Umum

Samuel sedang mengambil tugas akhir dan sekarang sedang melakukan tahap *_finishing_*

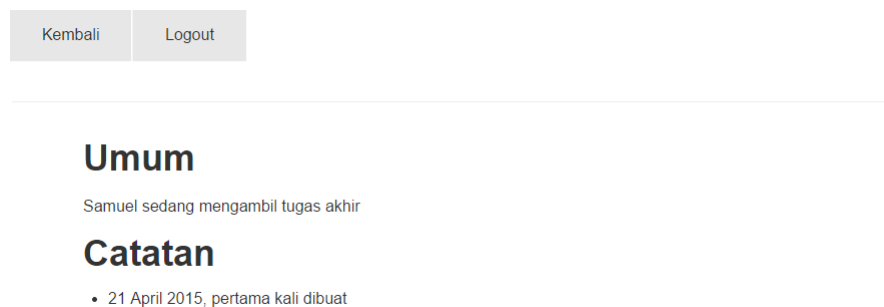
Catatan

- * 22 April 2015, melakukan perwalian ke-2
- * 21 April 2015, pertama kali dibuat

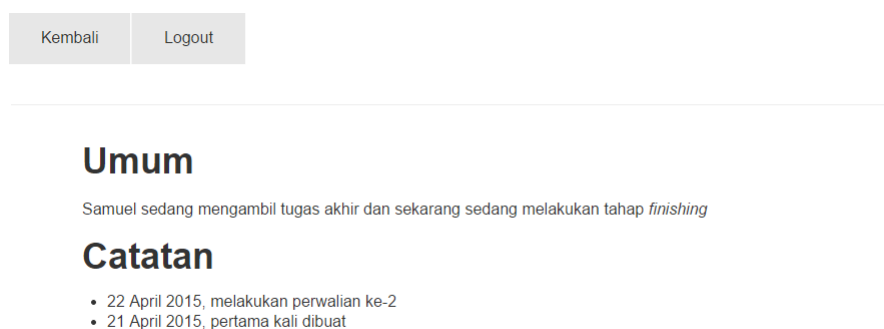
Gambar 5.10: Mengedit Info Mahasiswa



Gambar 5.11: Melihat Histori



Gambar 5.12: Keterangan Versi Pertama



Gambar 5.13: Keterangan Versi Kedua

Anda membuat catatan mahasiswa ini sebagai 7310013@student.unpar.ac.id.

Kembali
Simpan
Menu Utama
Logout

NPM

Nama

Umum
Isilah deskripsi umum mahasiswa disini.
Catatan
* 9 Oktober 2014, pertama kali dibuat

Gambar 5.14: Template Entri Baru

Anda membuat catatan mahasiswa ini sebagai 7310013@student.unpar.ac.id.

Kembali
Simpan
Menu Utama
Logout

NPM

Nama

Umum
Nadia mahasiswa tingkat akhir yang sedang menunggu sidang
Catatan
* 8 April 2015, pertama kali dibuat

Gambar 5.15: Membuat Entri Baru

Pilih NPM yang ingin dicari / tambah baru.

Add
Logout

NPM	Nama	Last Update
2003730013	Pascal	2015-05-05 09:26:15
2010120031	Kenneth Natanael	2015-04-30 14:29:51
2010730001	Andri Agustian	2015-04-29 15:09:33
2010730005	Grady Ireneus	2015-04-30 14:31:19
2010730012	Kevin PL	2015-04-24 00:55:57
2010730013	Samuel Herman	2015-04-24 00:23:34
2010730014	Nadia	2015-05-08 00:18:08

Gambar 5.16: Entri Baru Berhasil Dibuat

5.2.4 Hasil Pengujian Fungsional

Hasil pengujian fungsional dapat dilihat pada Tabel 5.1.

Tabel 5.1: Hasil Pengujian Fungsional

No	Aksi pengguna	Hasil yang diharapkan	Hasil yang didapatkan	Keterangan
1	✓	✓	✓	Berhasil
2	✓	✓	✓	Berhasil
3	✓	✓	✓	Berhasil
4	✓	✓	✓	Berhasil
5	✓	✓	✓	Berhasil
6	✓	✓	✓	Berhasil

5.2.5 Pengujian Eksperimental

Pengujian eksperimental dilakukan kepada dua kelompok mahasiswa. Kelompok pertama adalah mahasiswa jurusan teknik informatika dan kelompok kedua adalah mahasiswa jurusan non-teknik informatika. Kepada semua penguji baik kelompok pertama maupun kelompok kedua telah dijelaskan terlebih dahulu apa itu SIRM (Sistem Informasi Riwayat Mahasiswa), fungsi-fungsi yang terdapat pada SIRM, dan mekanisme yang dapat berlaku pada SIRM. Kemudian semua penguji diberikan tugas seolah-olah mereka adalah dosen yang menjalankan seluruh fungsi yang dimiliki SIRM (login, melihat daftar mahasiswa, melihat info mahasiswa, mengedit info mahasiswa, melihat histori mahasiswa, dan membuat entri baru) dan memahami SIRM secara keseluruhan. Setelah selesai melakukan pengujian, penguji akan diminta untuk mengisi angket. Penulis akan mengukur waktu selama pengujian dan mencatat berapa waktu yang dibutuhkan penguji untuk memahami dan menjalankan setiap fungsi dengan benar.

Kelompok pertama terdiri dari empat orang mahasiswa jurusan teknik informatika. Keempat penguji berhasil menjalankan dan memahami SIRM dengan baik. Untuk catatan waktu yang dimiliki oleh kelompok pertama dapat dilihat pada Tabel 5.2. Rata-rata waktu yang diperlukan kelompok pertama untuk menjalankan setiap fungsi dapat dilihat di bawah ini.

1 Login

$$105 + 52 + 93 + 127/4 = 377/4 = 94.25 \text{detik}$$

2 Melihat Daftar Mahasiswa

$$31 + 8 + 11 + 7/4 = 57/4 = 14.25 \text{detik}$$

3 Melihat Info Mahasiswa

$$24 + 12 + 6 + 19/4 = 61/4 = 15.25 \text{detik}$$

4 Mengedit Info Mahasiswa

$$37 + 43 + 39 + 25/4 = 144/4 = 36 \text{detik}$$

5 Melihat Histori Mahasiswa

$$10 + 5 + 14 + 36/4 = 64/4 = 16.25detik$$

6 Membuat Entri Baru

$$186 + 42 + 103 + 86/4 = 417/4 = 104.25detik$$

Tabel 5.2: Catatan Waktu Pengujian Eksperimental Kelompok Pertama

Aksi	Penguji 1	Penguji 2	Penguji 3	Penguji 4
Login	1 menit 45 detik	52 detik	1 menit 33 detik	2 menit 7 detik
Melihat Daftar Mahasiswa	31 detik	8 detik	11 detik	7 detik
Melihat Info Ma- hasiswa	24 detik	12 detik	6 detik	19 detik
Mengedit Info Mahasiswa	37 detik	43 detik	39 detik	25 detik
Melihat Histori Mahasiswa	10 detik	5 detik	14 detik	36 detik
Membuat Entri Baru	3 menit 6 detik	42 detik	1 menit 43 detik	1 menit 26 detik

1 Kelompok kedua terdiri dari empat orang mahasiswa jurusan non-teknik informatika.
 2 Keempat penguji berhasil menjalankan dan memahami SIRM dengan baik. Untuk catatan
 3 waktu yang dimiliki oleh kelompok kedua dapat dilihat pada Tabel 5.3. Rata-rata waktu
 4 yang diperlukan kelompok kedua untuk menjalankan setiap fungsi dapat dilihat di bawah
 5 ini.

1 Login

$$168 + 117 + 99 + 129/4 = 513/4 = 128.25detik$$

2 Melihat Daftar Mahasiswa

$$66 + 63 + 68 + 58/4 = 255/4 = 63.75detik$$

3 Melihat Info Mahasiswa

$$67 + 63 + 46 + 2/4 = 178/4 = 44.5detik$$

4 Mengedit Info Mahasiswa

$$64 + 90 + 73 + 48/4 = 275/4 = 68.75detik$$

5 Melihat Histori Mahasiswa

$$72 + 66 + 94 + 67/4 = 299/4 = 74.75\text{detik}$$

6 Membuat Entri Baru

$$68 + 138 + 124 + 205/4 = 535/4 = 133.75\text{detik}$$

Tabel 5.3: Catatan Waktu Pengujian Eksperimental Kelompok kedua

Aksi	Penguji 1	Penguji 2	Penguji 3	Penguji 4
Login	2 menit 48 detik	1 menit 57 detik	1 menit 39 detik	2 menit 9 detik
Melihat Daftar Mahasiswa	1 menit 6 detik	1 menit 3 detik	1 menit 8 detik	58 detik
Melihat Info Ma- hasiswa	1 menit 7 detik	1 menit 3 detik	46 detik	2 detik
Mengedit Info Mahasiswa	1 menit 4 detik	1 menit 30 detik	1 menit 13 detik	48 detik
Melihat Histori Mahasiswa	1 menit 12 detik	1 menit 6 detik	1 menit 34 detik	1 menit 7 detik
Membuat Entri Baru	1 menit 8 detik	2 menit 18 detik	2 menit 4 detik	3 menit 25 detik

- 1 Perbandingan dari rata-rata waktu pengujian eksperimental antara kelompok pertama
- 2 dan kelompok kedua dapat dilihat pada Tabel 5.4.

Tabel 5.4: Perbandingan Rata-rata Waktu Pengujian Eksperimental

Aksi	Kelompok Pertama	Kelompok Kedua	Keterangan
Login	94.25 detik	128.25 detik	Kelompok pertama lebih cepat 34 detik dari kelompok kedua
Melihat Daftar Mahasiswa	14.25 detik	63.75 detik	Kelompok pertama lebih cepat 49.5 detik dari kelompok kedua
Melihat Info Mahasiswa	15.25 detik	44.5 detik	Kelompok pertama lebih cepat 29.25 detik dari kelompok kedua
Mengedit Info Mahasiswa	36 detik	68.75 detik	Kelompok pertama lebih cepat 32.75 detik dari kelompok kedua
Melihat Histori Mahasiswa	16.25 detik	74.75 detik	Kelompok pertama lebih cepat 58.5 detik dari kelompok kedua
Membuat Entri Baru	104.25 detik	133.75 detik	Kelompok pertama lebih cepat 29.5 detik dari kelompok kedua

1

BAB 6

2

KESIMPULAN DAN SARAN

3 Pada bab ini akan diberikan kesimpulan yang didapat dari proses pemodelan dan pengujian
4 perangkat lunak yang dibangun, juga saran-saran untuk penelitian ini jika ingin dikem-
5 bangkan di kemudian hari.

6 **6.1 Kesimpulan**

7 SIRM

8 **6.2 Saran**

9 SIRM

DAFTAR REFERENSI

- [1] Google, Inc, *Google Identity Platform*. <https://developers.google.com/accounts/docs/OAuth2>, 2013.
- [2] Google, Inc, *Google+ Platform*. <https://developers.google.com/+api/oauth>, 2013.
- [3] A. Kumar, *Markdown Guide*. Self-published, 2015.
- [4] GitHub, Inc, *GitHub Flavored Markdown*. <https://help.github.com/articles/github-flavored-markdown/>, 2015.
- [5] A. Adib, *Strapdown.js - Instant and elegant Markdown documents*. <http://strapdownjs.com/>, 2014.
- [6] ZURB, Inc, *Foundation Documentation*. <http://foundation.zurb.com/docs/>, 2015.

LAMPIRAN A

KODE PROGRAM

Listing A.1: index.php

```
3 <!doctype html>
4 <html class="no-js" lang="en">
5   <head>
6     <meta charset="utf-8" />
7     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
8     <title>SIRM | Welcome</title>
9     <link rel="stylesheet" href="css/foundation.css" />
10    <script src="js/vendor/modernizr.js"></script>
11
12    <script src="https://apis.google.com/js/client:platform.js" async defer></script>
13  </head>
14  <body>
15    <div class="row">
16      <h1>WELCOME to SIRM <h4>(Sistem Informasi Riwayat Mahasiswa)</h4></h1>
17      <hr/>
18
19      <?php
20        include_once "google-api-php-client-master/src/Google/Client.php";
21        include_once "google-api-php-client-master/src/Google/Service/Oauth2.php";
22
23        session_start();
24
25        include_once "client.php";
26        $client->setScopes(array('https://www.googleapis.com/auth/plus.login','email'));
27        $plus = new Google_Service_Oauth2($client);
28
29        if (isset($_REQUEST['logout']))
30        {
31          unset($_SESSION['access_token']);
32          header('Location:_https://www.google.com/accounts/Logout?continue=https://
33            appengine.google.com/_ah/logout?continue=http://' . $_SERVER['HTTP_HOST']
34              . $_SERVER['PHP_SELF']);
35        }
36
37        if (isset($_GET['code']))
38        {
39          $client->authenticate($_GET['code']);
40          $_SESSION['access_token'] = $client->getAccessToken();
41          header('Location:_http://' . $_SERVER['HTTP_HOST'] . $_SERVER['PHP_SELF']);
42        }
43
44        if (isset($_SESSION['access_token']))
45        {
46          $client->setAccessToken($_SESSION['access_token']);
47        }
48
49        if ($client->getAccessToken())
50        {
51
52        }
53        else
54        {
55          $authUrl = $client->createAuthUrl();
56        }
57
58        if (isset($authUrl))
59        {
60          echo "<a class='login' href='" . $authUrl . "'>Login with Google</a>";
61        }
62      <?>
63    </div>
64  </body>
65 </html>
```

Listing A.2: oauth.php

```
66 <!doctype html>
67 <html class="no-js" lang="en">
68   <head>
69     <meta charset="utf-8" />
70     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
71     <title>SIRM | Oauth</title>
72     <link rel="stylesheet" href="css/foundation.css" />
73     <script src="js/vendor/modernizr.js"></script>
```

```

1      </head>
2      <body>
3          <?php
4              include_once "google-api-php-client-master/src/Google/Client.php";
5              include_once "google-api-php-client-master/src/Google/Service/Oauth2.php";
6
7              session_start();
8
9              include_once "client.php";
10             $client->setScopes(array('https://www.googleapis.com/auth/plus.login','email'));
11             $plus = new Google_Service_Oauth2($client);
12
13             if (isset($_GET['code']))
14             {
15                 $client->authenticate($_GET['code']);
16                 $_SESSION['access_token'] = $client->getAccessToken();
17                 header('Location:'.http://'. $_SERVER['HTTP_HOST'] . $_SERVER['PHP_SELF']);
18             }
19
20             if (isset($_SESSION['access_token']))
21             {
22                 $client->setAccessToken($_SESSION['access_token']);
23             }
24
25             if ($client->getAccessToken())
26             {
27                 $info = $plus->userinfo;
28                 $userinfo = $info->get();
29                 $email = ($userinfo['email']);
30                 $_SESSION['email'] = $email;
31             }
32
33             $status="";
34             function is_valid_email($email)
35             {
36                 $result = 'valid_email';
37                 if(!preg_match("[a-zA-Z0-9_+]+@unpar.ac.id+$", $email) && !preg_match("[a-zA-Z0-9_+]+@student.unpar.ac.id+$", $email))
38                 {
39                     $result = 'invalid_email';
40                 }
41                 return $result;
42             }
43             $status = is_valid_email($email);
44
45             if($status == "valid_email")
46             {
47                 header("Location: _list.php");
48                 exit;
49             }
50             else
51             {
52                 echo "<script>alert('Email yang digunakan tidak dapat mengakses SIRM. Email yang dapat digunakan untuk mengakses SIRM adalah email yang diakhiri @unpar.ac.id atau @student.unpar.ac.id. ');window.location.href='index.php?logout';</script>";
53                 exit;
54             }
55             ?>
56             <?= is_valid_email($email); ?>
57         </body>
58     </html>

```

Listing A.3: list.php

```

63     <!doctype html>
64     <html class="no-js" lang="en">
65         <head>
66             <meta charset="utf-8" />
67             <meta name="viewport" content="width=device-width, initial-scale=1.0" />
68             <title>SIRM | List</title>
69             <link rel="stylesheet" href="css/foundation.css" />
70             <script src="js/vendor/modernizr.js"></script>
71         </head>
72         <body>
73             <?php
74                 session_start();
75
76                 ?>
77                 <div class="row">
78                     <h3>Pilih NPM yang ingin dicari / tambah baru.</h3>
79                     <ul class="button-group">
80                         <li><a href="new.php" class="button_secondary">Add</a></li>
81                         <li><a href="index.php?logout" class="button_secondary">Logout</a></li>
82                     </ul>
83                     <hr/>
84                 </div>
85
86                 <div class="row">
87                     <?php
88                         include_once "configDatabase.php";
89
90                         if(! $id_mysql)
91                         {
92                             die("Database tidak bisa dibuka");
93                         }
94
95                         if(! mysql_select_db("sirm", $id_mysql))
96                         {
97                             die("Database tidak bisa dipilih");
98                         }

```

```

1      $hasil = mysql_query("SELECT_*_FROM_info_mahasiswa", $id_mysql);
2
3      if(! $hasil)
4      {
5          die("Permintaan_gagal");
6      }
7
8      echo "<table>
9      .....<thead>
10     .....<tr>
11     .....<th_width='250'>NPM</th>
12     .....<th_width='500'>Nama</th>
13     .....<th_width='250'>Last_Update</th>
14     .....</tr>
15     .....</thead>";
16
17     while($row = mysql_fetch_array($hasil))
18     {
19         echo "<tr>";
20         echo "<td><a_href='view.php?npm='". $row['npm'] . "'>" . $row['npm'] . "</a></td>";
21         echo "<td>" . $row['nama'] . "</td>";
22         echo "<td>" . $row['pembaruan_terakhir'] . "</td>";
23         echo "</tr>";
24     }
25     echo "</table>";
26
27     ?>
28     </div>
29     </body>
30 </html>

```

Listing A.4: view.php

```

30 <!doctype html>
31 <html class="no-js" lang="en">
32     <head>
33         <meta charset="utf-8" />
34         <meta name="viewport" content="width=device-width, initial-scale=1.0" />
35         <title>SIRM | View</title>
36         <link rel="stylesheet" href="css/foundation.css" />
37         <script src="js/vendor/modernizr.js"></script>
38     </head>
39     <body>
40         <?php
41             session_start();
42             $npm = $_GET["npm"];
43         ?>
44         <div class="row">
45             <div class="small-11 small-centered columns">
46                 <h3>Anda melihat catatan mahasiswa ini sebagai <?php echo $_SESSION['email']?>. </
47                 h3>
48                 <ul class="button-group">
49                     <li><a href="edit.php?npm=<?php echo $npm?>" class="button_secondary">Edit </a
50                     ></li>
51                     <li><a href="history.php?npm=<?php echo $npm?>" class="button_secondary">Lihat
52                     Histori </a></li>
53                     <li><a href="list.php" class="button_secondary">Menu Utama </a></li>
54                     <li><a href="index.php?logout" class="button_secondary">Logout </a></li>
55                 </ul>
56                 <hr/>
57             <?php
58                 include_once "configDatabase.php";
59
60                 if(! $id_mysql)
61                 {
62                     die("Database_tidak_bisa_dibuka");
63                 }
64
65                 if(! mysql_select_db("sirm", $id_mysql))
66                 {
67                     die("Database_tidak_bisa_dipilih");
68                 }
69
70                 $lihat = "INSERT INTO _histori_ (npm, _pengguna, _status, _tanggal_pembaruan, _keterangan) _
71                 VALUES_ ('" . mysql_real_escape_string($npm) . "', '" . $_SESSION['email'] . "', '" .
72                 melihat', _now(), _)" ;
73
74                 if (mysql_query($lihat) == TRUE)
75                 {
76
77                 }
78                 else
79                 {
80                     echo "Error:_" . $lihat . "<br>" . $id_mysql->error;
81                 }
82
83                 $cari = mysql_query("SELECT_*_FROM_info_mahasiswa WHERE _npm='$_npm'", $id_mysql);
84
85                 while($row = mysql_fetch_array($cari))
86                 {
87                     echo "NPM:_"; echo $row['npm']; echo "<br>";
88                     echo "Nama:_"; echo $row['nama']; echo "<br>";
89                 }
90             ?>
91         </div>
92     <xmp style="display:none;">
93     <?php
94         echo $row['keterangan'];
95     ?>
96 </xmp>
97     <?php
98     }

```

```

1      ?>
2      <script src="js/0.2/strapdown.js"></script>
3  </body>
4  </html>

```

Listing A.5: edit.php

```

5  <!doctype html>
6  <html class="no-js" lang="en">
7      <head>
8          <meta charset="utf-8" />
9          <meta name="viewport" content="width=device-width, initial-scale=1.0" />
10         <title>SIRM | Edit</title>
11         <link rel="stylesheet" href="css/foundation.css" />
12         <script src="js/vendor/modernizr.js"></script>
13     </head>
14     <body>
15         <?php
16             session_start();
17             $npm = $_GET['npm'];
18
19             include_once "configDatabase.php";
20
21             if(! $id_mysql)
22             {
23                 die("Database_tidak_bisa_dibuka");
24             }
25
26             if(! mysql_select_db("sirm", $id_mysql))
27             {
28                 die("Database_tidak_bisa_dipilih");
29             }
30
31             $hasil = mysql_query("SELECT_*FROM_info_mahasiswa_WHERE_npm='$npm'", $id_mysql);
32
33             if(! $hasil)
34             {
35                 die("Permintaan_gagal");
36             }
37
38             while($row = mysql_fetch_array($hasil))
39             {
40                 $carinama = $row['nama'];
41                 $cariketerangan = $row['keterangan'];
42             }
43
44             if(isset($_POST['submit']))
45             {
46                 $keteranganbaru = "";
47                 $keteranganbaru = $_POST['keteranganbaru'];
48
49                 $sql1 = "UPDATE_info_mahasiswa_SET_keterangan='$keteranganbaru',_pembaruan_terakhir=
50                     now()_WHERE_npm='$npm'";
51                 $sql2 = "INSERT INTO_histori_(npm,_pengguna,_status,_tanggal_pembaruan,_keterangan)_
52                     VALUES('". mysql_real_escape_string($npm) ."',_,'". $SESSION['email'] ."',_,'
53                     mengedit',_now(),_,'". mysql_real_escape_string($keteranganbaru) ."')";
54
55                 if (mysql_query($sql1) & mysql_query($sql2) == TRUE)
56                 {
57                     echo 'META-HTTP-EQUIV="Refresh" _CONTENT="1;URL=list.php">';
58                 }
59                 else
60                 {
61                     echo "Error:_ " . $sql1 . " <br>" . $id_mysql->error;
62                     echo "Error:_ " . $sql2 . " <br>" . $id_mysql->error;
63                 }
64             }
65             else
66             {
67                 ?>
68                 <div class="row">
69                     <h3>Anda mengedit catatan mahasiswa ini sebagai <?php echo $SESSION['email']?>.<br>
70                     NPM <?php echo $npm; ?> Nama <?php echo $carinama; ?>
71                     </h3>
72                 </div>
73                 <form method="post" action="edit.php?npm=<?php echo $npm?>">
74                     <div class="row">
75                         <ul class="button-group">
76                             <li><a href="view.php?npm=<?php echo $npm?>" class="button_secondary">Kembali
77                                 </a></li>
78                             <li><input class="button" type="submit" name="submit" value="Simpan"></li>
79                             <li><a href="list.php" class="button_secondary">Menu Utama</a></li>
80                             <li><a href="index.php?logout" class="button_secondary">Logout</a></li>
81                         </ul>
82                         <hr/>
83                     </div>
84                     <div class="row">
85                         <div class="small-12_columns">
86                             <textarea style="height:_300px;" placeholder="<?php echo $cariketerangan; ?>"
87                                 name="keteranganbaru"><?php echo $cariketerangan; ?></textarea>
88                         </div>
89                     </div>
90                 </form>
91             <?php
92             }
93             ?>
94     </body>
95 </html>

```


Listing A.6: history.php

```

1 <!doctype html>
2 <html class="no-js" lang="en">
3   <head>
4     <meta charset="utf-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <title>SIRM | History</title>
7     <link rel="stylesheet" href="css/foundation.css" />
8     <script src="js/vendor/modernizr.js"></script>
9   </head>
10  <body>
11    <?php
12      $npm = $_GET["npm"];
13
14      include_once "configDatabase.php";
15
16      if(! $id_mysql)
17      {
18        die("Database_tidak_bisa_dibuka");
19      }
20
21      if(! mysql_select_db("sirm", $id_mysql))
22      {
23        die("Database_tidak_bisa_dipilih");
24      }
25
26      $hasil = mysql_query("SELECT_*_FROM_info_mahasiswa_WHERE_npm='$npm'", $id_mysql);
27
28      if(! $hasil)
29      {
30        die("Permintaan_gagal");
31      }
32
33      while($row = mysql_fetch_array($hasil))
34      {
35        $carinpm = $row['npm'];
36        $carinama = $row['nama'];
37      }
38
39    ?>
40    <div class="row">
41      <ul class="button-group">
42        <li><a href="view.php?npm=<?php echo $npm?>" class="button_secondary">Kembali</a>
43        </li>
44        <li><a href="list.php" class="button_secondary">Menu Utama</a></li>
45        <li><a href="index.php?logout" class="button_secondary">Logout</a></li>
46      </ul>
47
48    </div>
49    <div class="row">
50      <h3>NPM <?php echo $carinpm; ?> Nama <?php echo $carinama; ?>
51      <hr/>
52      <ul class="disc">
53        <?php
54          $hasil = mysql_query("SELECT_*_FROM_histori_WHERE_npm='$npm'_ORDER_BY_
55            id_histori_DESC", $id_mysql);
56
57          if(! $hasil)
58          {
59            die("Permintaan_gagal");
60          }
61
62          while($row = mysql_fetch_array($hasil))
63          {
64            echo "<li>" . $row['tanggal_pembaruan'] . " " . $row['pengguna'] . " " .
65              $row['status'] . " " . $row['npm'] . " " . ($row['keterangan'] != "" ?
66                '<a href="past.php?id=_'. $row['id_histori'] . "'>[lihat_versi_ini]</a>'
67                >' : ""') . "</li>";
68          }
69
70        ?>
71      </ul>
72    </div>
73  </body>
74 </html>

```

Listing A.7: past.php

```

75 <!doctype html>
76 <html class="no-js" lang="en">
77   <head>
78     <meta charset="utf-8" />
79     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
80     <title>SIRM | Past</title>
81     <link rel="stylesheet" href="css/foundation.css" />
82     <script src="js/vendor/modernizr.js"></script>
83   </head>
84   <body>
85     <?php
86       session_start();
87       $id = $_GET["id"];
88       ?>
89     <div class="row">
90       <div class="small-11_small-centered_columns">
91         <ul class="button-group">
92           <li><a href="javascript:history.back(1)" class="button_secondary">Kembali</a></li>
93           <li><a href="index.php?logout" class="button_secondary">Logout</a></li>
94         </ul>
95       </div>
96     <?php
97

```

```

1      include_once "configDatabase.php";
2
3      if (! $id_mysql)
4      {
5          die("Database_tidak_bisa_dibuka");
6      }
7
8      if (! mysql_select_db("sirm", $id_mysql))
9      {
10         die("Database_tidak_bisa_dipilih");
11     }
12
13     $cari = mysql_query("SELECT_keterangan_FROM_histori_WHERE_id_histori=' $id '",
14         $id_mysql);
15     while($row = mysql_fetch_array($cari))
16     {
17         ?>
18     }
19 }
20 </div>
21 <?php
22 echo $row['keterangan'];
23 ?>
24 </xmp>
25 <?php
26 }
27 ?>
28 <script src="js/0.2/strapdown.js"></script>
29 </body>
30 </html>

```

Listing A.8: new.php

```

31 <!doctype html>
32 <html class="no-js" lang="en">
33 <head>
34 <meta charset="utf-8" />
35 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
36 <title>SIRM | New</title>
37 <link rel="stylesheet" href="css/foundation.css" />
38 <script src="js/vendor/modernizr.js"></script>
39 </head>
40 <body>
41 <?php
42     session_start();
43
44     if (isset($_POST['submit']))
45     {
46         include_once "configDatabase.php";
47
48         if (! $id_mysql)
49         {
50             die("Database_tidak_bisa_dibuka");
51         }
52
53         if (! mysql_select_db("sirm", $id_mysql))
54         {
55             die("Database_tidak_bisa_dipilih");
56         }
57
58         $npm = $nama = $keterangan = "";
59
60         $npm = $_POST['npm'];
61         $nama = $_POST['nama'];
62         $keterangan = $_POST['keterangan'];
63
64         $cek = "SELECT_npm_from_info_mahasiswa_where_npm='". mysql_real_escape_string($npm) .
65             "'";
66         $found = mysql_query($cek) or die(mysql_error());
67         if (mysql_num_rows($found) > 0)
68         {
69             echo "<script>alert('Data_telah_terdaftar._Silahkan_diulangi_dengan_data_yang_lain
70                 .');window.location.href='new.php';</script>";
71         }
72     }
73     else
74     {
75         $sql1 = "INSERT INTO info_mahasiswa (npm, nama, keterangan) VALUES ('" .
76             mysql_real_escape_string($npm) . "', '" . mysql_real_escape_string($nama) . "',
77             '" . mysql_real_escape_string($keterangan) . "')";
78         $sql2 = "INSERT INTO histori (npm, pengguna, status, tanggal_pembaruan, keterangan)
79             VALUES ('" . mysql_real_escape_string($npm) . "', '" . $_SESSION['email'] . "', '" .
80             membuat_entri($_now()) . "', '" . mysql_real_escape_string($keterangan) . "')";
81
82         if (mysql_query($sql1) & mysql_query($sql2) === TRUE)
83         {
84             echo '<META_HTTP-EQUIV="Refresh" _CONTENT="1;_URL=list.php">';
85         }
86         else
87         {
88             echo "Error:_" . $sql1 . "<br>" . $id_mysql->error;
89             echo "Error:_" . $sql2 . "<br>" . $id_mysql->error;
90         }
91     }
92 }
93 <?php
94 <div class="row">
95 <h3>Anda membuat catatan mahasiswa ini sebagai <?php echo $_SESSION['email']?>.</
96     h3>
97 </div>
98

```

```

1      <form method="post" action="<?php_echo_htmlspecialchars($_SERVER["PHP_SELF"]);?>">
2          <div class="row">
3              <ul class="button-group">
4                  <li><a href="list.php" class="button-secondary">Kembali</a></li>
5                  <li><input class="button" type="submit" name="submit" value="Simpan"></li>
6                  <li><a href="list.php" class="button-secondary">Menu Utama</a></li>
7                  <li><a href="index.php?logout" class="button-secondary">Logout</a></li>
8              </ul>
9              <hr/>
10          </div>
11          <div class="small-8-columns">
12              <div class="row">
13                  <div class="small-3-columns">
14                      <label for="right-label" class="right-inline">NPM</label>
15                  </div>
16                  <div class="small-9-columns">
17                      <input type="text" name="npm" id="right-label" placeholder="Masukan_
18                          NPM">
19                  </div>
20              </div>
21              <div class="row">
22                  <div class="small-3-columns">
23                      <label for="right-label" class="right-inline">Nama</label>
24                  </div>
25                  <div class="small-9-columns">
26                      <input type="text" name="nama" id="right-label" placeholder="Masukan_
27                          nama">
28                  </div>
29              </div>
30          </div>
31          <div class="row">
32              <div class="small-12-columns">
33                  <textarea style="height: 300px;" name="keterangan">
34          # Umum
35
36          Isilah deskripsi umum mahasiswa disini.
37
38          # Catatan
39
40          * 9 Oktober 2014, pertama kali dibuat
41              </textarea>
42          </div>
43      </div>
44  </form>
45  <?php
46      }
47  ?>
48 </body>
49 </html>

```

Listing A.9: client.php

```

50 <?php
51 $client = new Google_Client();
52 $client->setClientId('568951368854-ufmbistn0pcaq0khubafola133orfgve.apps.googleusercontent.com');
53 $client->setClientSecret('-cSZ-AUmeQ9PaWWry_IpiBBi');
54 $client->setRedirectUri('http://localhost/oauth.php');
55 $client->setDeveloperKey('AlzaSyDRoDJAzUR_TsNUNRUeTYsBb7dFBQKZy7M');
56 ?>

```

Listing A.10: configDatabase.php

```

58 <?php
59 $pemakai="admin";
60 $pass="admin";
61 $id_mysql=mysql_connect("localhost", $pemakai, $pass);
62 ?>

```


LAMPIRAN B

THE SOURCE CODE

Listing B.1: MyFurSet.java

```
3
4 import java.util.ArrayList;
5 import java.util.Collections;
6 import java.util.HashSet;
7
8 /**
9  *
10  * @author Lionov
11  */
12
13 //class for set of vertices close to furthest edge
14 public class MyFurSet {
15     protected int id; //id of the set
16     protected MyEdge FurthestEdge; //the furthest edge
17     protected HashSet<MyVertex> set; //set of vertices close to furthest edge
18     protected ArrayList<ArrayList<Integer>> ordered; //list of all vertices in the set for each
19         trajectory
20     protected ArrayList<Integer> closeID; //store the ID of all vertices
21     protected ArrayList<Double> closeDist; //store the distance of all vertices
22     protected int totaltrj; //total trajectories in the set
23
24     /**
25      * Constructor
26      * @param id : id of the set
27      * @param totaltrj : total number of trajectories in the set
28      * @param FurthestEdge : the furthest edge
29      */
30     public MyFurSet(int id,int totaltrj,MyEdge FurthestEdge) {
31         this.id = id;
32         this.totaltrj = totaltrj;
33         this.FurthestEdge = FurthestEdge;
34         set = new HashSet<MyVertex>();
35         ordered = new ArrayList<ArrayList<Integer>>();
36         for (int i=0;i<totaltrj;i++) ordered.add(new ArrayList<Integer>());
37         closeID = new ArrayList<Integer>(totaltrj);
38         closeDist = new ArrayList<Double>(totaltrj);
39         for (int i = 0;i <totaltrj;i++) {
40             closeID.add(-1);
41             closeDist.add(Double.MAX_VALUE);
42         }
43     }
44
45     /**
46      * set a vertex into the set
47      * @param v : vertex to be added to the set
48      */
49     public void add(MyVertex v) {
50         set.add(v);
51     }
52
53     /**
54      * check whether vertex v is a member of the set
55      * @param v : vertex to be checked
56      * @return true if v is a member of the set, false otherwise
57      */
58     public boolean contains(MyVertex v) {
59         return this.set.contains(v);
60     }
61
62     /**
63      * create a column for table Gamma, sorted for each row
64      */
65     public void createColumn() {
66         for (MyVertex v : set) {
67             for (Integer key : v.vertexnum.keySet()) {
68                 for (Integer values : v.vertexnum.get(key)) {
69                     ordered.get(key).add(values);
70                 }
71             }
72         }
73         for (ArrayList<Integer> al : ordered) Collections.sort(al);
74     }
75
76
77 }
```