

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'S HISTORY INFORMATION SYSTEM



SAMUEL HERMAN

NPM: 2010730013

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

ABSTRAK

Perbandingan jumlah dosen dan jumlah mahasiswa menjadi masalah bagi dosen dalam mengingat riwayat setiap mahasiswa. Maka dari itu pada tugas akhir ini akan membahas tentang pembangun Sistem Informasi Riwayat Mahasiswa dapat disebut 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 fungsi login yang akan membatasi ruang lingkup pengguna. Markdown akan digunakan sebagai format dalam penulisan riwayat mahasiswa. StrapdownJS akan digunakan untuk menampilkan teks yang ditulis menggunakan sintakas Markdown ke HTML. Zurb Foundation akan digunakan untuk membuat tampilan antarmuka SIRM. Dengan pembangunan SIRM, diharapkan dapat membantu dosen dalam memfasilitasi mahasiswa.

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

ABSTRACT

The ratio between the number of the lecturers and the students has become the main problem for the lecturers to remember every history of their students. Therefore, this thesis will discuss about the establishment of the Student's History Information System can be called SIRM. SIRM is build by using four technology, they are Google Authentication, Markdown, StrapdownJS, and Zurb Foundation.

Google Authentication will be used a login system which limits the scope for the users. Markdown functions as a form in recording the student's history. Meanwhile, we can use StrapdownJS to feature the text which is written using Markdown syntax to HTML. Finally, Zurb Foundation acts as the appearance of SIRM interface. It is hoped that SIRM can help the lecturers to facilitate their students well.

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	3
1.6 Sistematika Pembahasan	3
2 LANDASAN TEORI	5
2.1 Google OAuth [1]	5
2.1.1 Langkah Dasar	5
2.1.2 Skenario Google OAuth	6
2.1.3 Masa Habis Berlaku Token	10
2.1.4 Lingkup Otorisasi [2]	11
2.2 Markdown	13
2.2.1 Apa itu Markdown? [3]	13
2.2.2 Sintaks yang Berguna [3]	13
2.2.3 GitHub Flavored Markdown [4]	19
2.3 StrapdownJS [5]	23
2.4 Zurb Foundation [6]	24
2.4.1 Kompatibilitas	24
2.4.2 Apa Saja yang Dapat Dibuat dengan Foundation?	24
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 <i>Entity Relationship</i> Diagram (ERD)	50
3.5.4 <i>Data Flow</i> Diagram	50
4 PERANCANGAN	53

4.1	Perancangan Tampilan <i>Web</i> Yang Digunakan	53
4.1.1	Tampilan Halaman Awal	53
4.1.2	Tampilan <i>Web</i> Pilih Mahasiswa	53
4.1.3	Tampilan <i>Web</i> Info Mahasiswa	54
4.1.4	Tampilan <i>Web</i> Edit Mahasiswa	54
4.1.5	Tampilan <i>Web</i> Lihat Histori	55
4.1.6	Tampilan <i>Web</i> Lihat Versi Ini	55
4.1.7	Tampilan <i>Web</i> Entri Baru	56
4.2	Perancangan Modul	56
4.2.1	Modul Login	56
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	Implementasi Basis Data	62
5.3	Pengujian Perangkat Lunak	62
5.3.1	Lingkungan Pengujian Perangkat Keras	63
5.3.2	Lingkungan Pengujian Perangkat Lunak	63
5.3.3	Pengujian Fungsional	63
5.3.4	Pengujian Eksperimental	70
6	KESIMPULAN DAN SARAN	77
6.1	Kesimpulan	77
6.2	Saran	77
	DAFTAR REFERENSI	79
	A KODE PROGRAM	81
	B KODE PROGRAM UNTUK PENGUJIAN	89
	C KUESIONER	97

DAFTAR GAMBAR

2.1	Skenario Aplikasi Layanan <i>Web</i>	7
2.2	Skenario Aplikasi yang Terinstal	8
2.3	Skenario Aplikasi Sisi Klien (JavaScript)	8
2.4	Skenario Aplikasi Pada Perangkat Dengan Masukan Yang Terbatas	9
2.5	Skenario Layanan Akun	10
2.6	Markdown Cetak Tebal dan Cetak Miring	14
2.7	Markdown Judul Bab	14
2.8	Markdown Batas Baris Dengan Tiga Spasi	15
2.9	Markdown Batas Baris Tanpa Tiga Spasi	15
2.10	Markdown Paragraf	15
2.11	Markdown Menampilkan Gambar	16
2.12	Markdown Link	16
2.13	Markdown Kode	17
2.14	Markdown Kutipan	17
2.15	Markdown Garis Horisontal	18
2.16	Markdown Daftar Tidak Berurutan	19
2.17	Markdown Daftar Berurutan	19
2.18	GFM Garis Bawah	20
2.19	GFM Taut Otomatis URL	20
2.20	GFM Tanda Coret	20
2.21	GFM Blok Kode	21
2.22	GFM Penandaan Sintaks	21
2.23	GFM Tabel 1	21
2.24	GFM Tabel 2	22
2.25	GFM Tabel 3	22
2.26	GFM Tabel 4	22
2.27	GFM Tabel 5	23
2.28	Contoh <i>Grid</i> Zurb Foundation	25
2.29	<i>Grid</i> Utama	26
2.30	<i>Grid</i> Kecil	26
2.31	<i>Grid</i> Menengah	27
2.32	<i>Grid</i> Tingkat Lanjut	27
2.33	<i>Grid</i> Dengan Offset	27
2.34	<i>Grid</i> Dengan Baris yang Tidak Lengkap	28
2.35	<i>Grid</i> Dengan Baris yang Mengempis/Tidak Mengempis	28
2.36	<i>Grid</i> Dengan Kolom yang Berpusat	29
2.37	<i>Grid</i> Dengan Sumber yang Terurut	29
2.38	Contoh Tombol Zurb Foundation	30
2.39	Tombol Dasar	30
2.40	Ukuran dan Bentuk Tombol	31
2.41	Warna-warna Tombol	31
2.42	Aksesibilitas Tombol	32

2.43	Tabel Foundation	32
2.44	Contoh Navigasi Zurb Foundation	33
2.45	Contoh Plugins Zurb Foundation	33
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	<i>Use Case</i> Diagram Sistem Informasi Riwayat Mahasiswa	46
3.19	<i>Entity Relationship</i> Diagram	50
3.20	Data <i>Context</i> Diagram Sistem Informasi Riwayat Mahasiswa	51
3.21	Data <i>Flow</i> 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	65
5.2	Login Dengan Email yang Diakhiri "@student.unpar.ac.id"	65
5.3	Konfirmasi Email yang Dikelola oleh student.unpar.ac.id	66
5.4	CAS UNPAR	66
5.5	Izin Akses Dari Pihak Pengguna	66
5.6	Login Dengan Email yang Diakhiri "@gmail.com"	67
5.7	Alert Email yang Digunakan Tidak Dapat Mengakses SIRM	67
5.8	Memilih Mahasiswa	67
5.9	Melihat Info Mahasiswa	68
5.10	Mengedit Info Mahasiswa	68
5.11	Melihat Histori	68
5.12	Keterangan Versi Pertama	69
5.13	Keterangan Versi Kedua	69
5.14	Template Entri Baru	69
5.15	Membuat Entri Baru	70

5.16	Entri Baru Berhasil Dibuat	70
5.17	Diagram Perbandingan Rata-Rata Waktu Pengujian	74
5.18	Diagram Perbandingan Total Nilai Kuesioner Setiap Responden	75

DAFTAR TABEL

2.1	Tabel Daftar Pengujian Zurb Foundation	24
3.1	Tabel Skenario Login	47
3.2	Tabel Skenario Pilih Mahasiswa	48
3.3	Tabel Skenario Melihat Info Mahasiswa	48
3.4	Tabel Skenario Edit Mahasiswa	49
3.5	Tabel Skenario Lihat Histori	49
3.6	Tabel Skenario Membuat Entri Baru	50
4.1	Tabel Modul Login	57
4.2	Tabel Modul Pilih Mahasiswa	57
4.3	Tabel Modul Info Mahasiswa	57
4.4	Tabel Modul <i>Edit</i> Mahasiswa	58
4.5	Tabel Modul Lihat Histori	58
4.6	Tabel Modul Entri Baru	58
4.7	Tabel Rancangan Tabel Info Mahasiswa	59
4.8	Tabel Rancangan Tabel Histori	59
5.1	Tabel Catatan Waktu Pengujian Eksperimental Kelompok Pertama	71
5.2	Tabel Catatan Waktu Pengujian Eksperimental Kelompok Kedua	72
5.3	Tabel Perbandingan Rata-rata Waktu Pengujian Eksperimental	73
5.4	Tabel Sistem Penilaian Kuesioner	74
5.5	Tabel Nilai Kuesioner Kelompok Pertama	75
5.6	Tabel Nilai Kuesioner Kelompok Kedua	75
5.7	Tabel Rata-Rata Nilai Kuesioner Kedua Kelompok	76

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Saat ini jumlah dosen dan jumlah mahasiswa menjadi pemasalahan. Permasalahan tersebut disebabkan karena minimnya jumlah dosen dengan kata lain kurangnya tenaga 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 riwayat 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 data 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 sintaks Markdown?
- Bagaimana menampilkan teks dengan sintaks Markdown ke halaman website?
- Bagaimana merancang antarmuka Sistem Informasi Riwayat Mahasiswa menggunakan Zurb Foundation?

- 1 • Bagaimana mengimplementasikan Sistem Informasi Riwayat Mahasiswa yang telah
2 dirancang kedalam *script* PHP?

3 1.3 Tujuan

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

- 6 • Mengautentikasi pengguna menggunakan Google Authentication.
- 7 • Menggunakan teks dengan sintaks Markdown.
- 8 • Menampilkan teks dengan sintaks Markdown ke halaman website.
- 9 • Merancang antarmuka Sistem Informasi Riwayat Mahasiswa menggunakan Zurb Fo-
10 undation.
- 11 • Mengimplementasikan Sistem Informasi Riwayat Mahasiswa yang telah dirancang ke-
12 dalam *script* PHP.

13 1.4 Batasan Masalah

14 Dalam penelitian ini ditetapkan batasan-batasan yang akan menjadi pedoman dalam pelak-
15 sanaan penelitian:

- 16 • Perangkat lunak akan memiliki 6 fitur yaitu: Login, Pilih mahasiswa, Melihat info
17 mahasiswa, *Edit* mahasiswa, Lihat histori, dan Membuat entri baru.
- 18 • Untuk fitur login hanya untuk dosen yang diakhiri dengan @unpar.ac.id dan *username*
19 bukan angka semua.
- 20 • Untuk fitur pilih mahasiswa, pengguna dapat memilih mahasiswa yang ingin dilihat
21 atau dirubah dan pengguna juga bisa menekan tombol "*Add*" untuk menambah data
22 mahasiswa baru.
- 23 • Untuk fitur melihat info mahasiswa, pengguna dapat melihat info terkini dari maha-
24 siswa dan aksi ini dicatat dalam log untuk alasan penjagaan privasi.
- 25 • Untuk fitur edit mahasiswa, pengguna dapat mengubah info mahasiswa dan aksi ini
26 juga dicatat dalam log.
- 27 • Untuk fitur lihat histori, pengguna dapat melihat histori setiap aksi perubahan atau
28 aksi *view*. Pada fitur ini juga pengguna dapat melihat info mahasiswa versi pertama
29 dan versi selanjutnya hingga versi saat ini.
- 30 • Untuk fitur membuat entri baru, saat membuat entri baru akan dibuatkan *template*
31 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.
- Menganalisis cara kerja teknologi yang akan digunakan untuk membangun Sistem Informasi Riwayat Mahasiswa.
- Merancang Sistem Informasi Riwayat Mahasiswa yang akan dibuat.
- Melakukan implementasi untuk Sistem Informasi Riwayat Mahasiswa yang telah dirancang ke dalam *script* PHP.
- Melakukan pengujian perangkat lunak yang telah diimplementasikan.

1.6 Sistematika Pembahasan

Sistematika pembahasan dalam penelitian ini adalah sebagai berikut:

- Bab I Pendahuluan
Bab ini menjelaskan latar belakang permasalahan, rumusan masalah, tujuan, batasan masalah, metodologi penelitian, dan sistematika pembahasan.
- Bab II Dasar Teori
Bab ini menjelaskan teori-teori dasar mengenai Google Authentication, Markdown Syntax, StrapdownJS, dan Zurb Foundation yang menjadi referensi utama dalam pelaksanaan penelitian.
- Bab III Analisis
Bab ini berisi analisis mengenai Google Authentication, Markdown Syntax, StrapdownJS, dan Zurb Foundation yang akan digunakan pada penelitian ini.
- Bab IV Perancangan
Bab ini berisi perancangan Sistem Informasi Riwayat Mahasiswa yang akan dibuat.
- Bab V Implementasi dan Pengujian
Bab ini berisi pengimplementasian dan pengujian Sistem Informasi Riwayat Mahasiswa.
- Bab VI Kesimpulan dan Saran
Bab ini berisi kesimpulan dari hasil penelitian dan saran untuk pengembangan lebih lanjut.

BAB 2

LANDASAN TEORI

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

2.1 Google OAuth [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 eror.

(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 OAuth

Terdapat lima skenario yang dapat digunakan untuk Google OAuth 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.

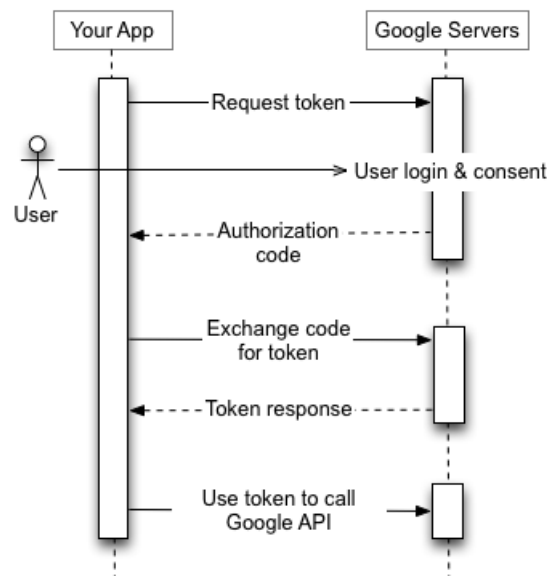
2.1.2.1 Skenario Aplikasi Layanan *Web*

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*

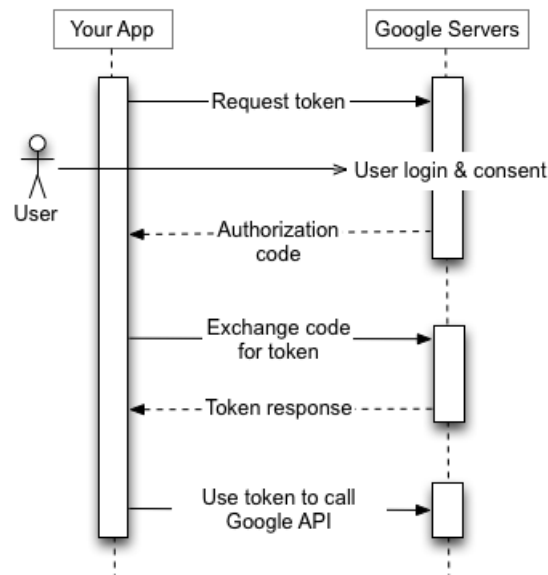
8 2.1.2.2 Skenario Aplikasi yang Terinstal

9 Google OAuth 2.0 mendukung aplikasi yang diinstal pada perangkat seperti komputer, per-
 10 angkat *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.

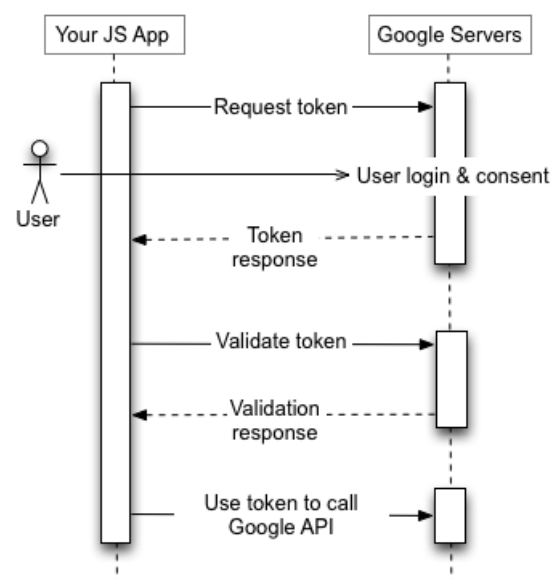
20 Aplikasi harus menyimpan token yang diperbaharui untuk penggunaan masa depan dan
 21 menggunakan token akses untuk mengakses API Google. Setelah masa token akses berakhir,
 22 maka aplikasi akan memperbaharui token untuk mendapatkan yang baru. Untuk gambar
 23 skenario dapat dilihat pada Gambar 2.2.



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.

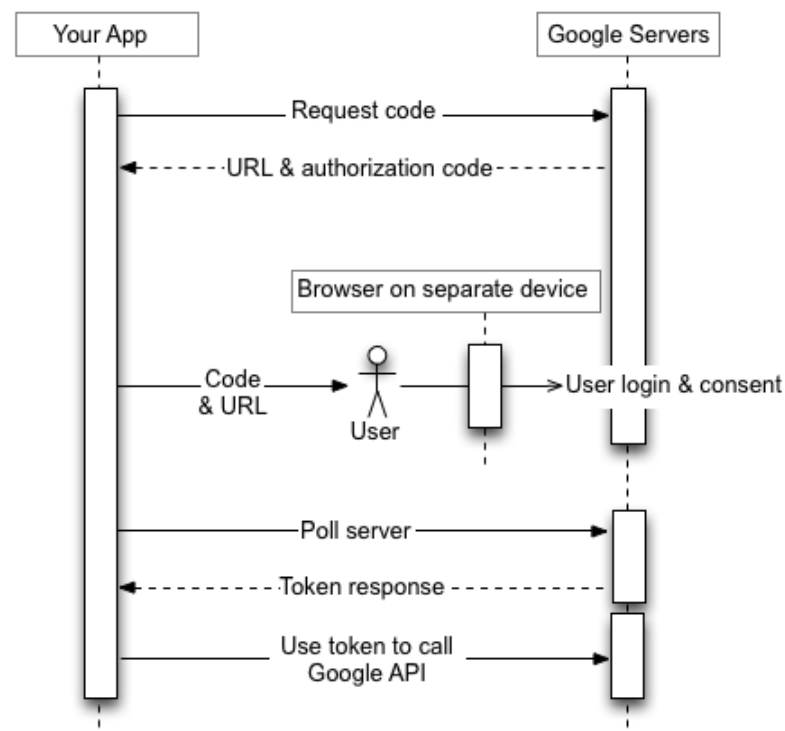


Gambar 2.3: Skenario Aplikasi Sisi Klien (JavaScript)

2.1.2.4 Skenario Aplikasi Pada Perangkat Dengan Masukan Yang Terbatas

Google OAuth 2.0 mendukung aplikasi yang berjalan pada perangkat dengan masukan yang terbatas seperti konsol permainan, kamera video, dan printer. Urutan otorisasi dimulai dengan aplikasi membuat permintaan layanan *web* ke URL Google untuk kode otorisasi. Tanggapan berisi beberapa parameter, termasuk URL dan kode bahwa aplikasi menunjukkan kepada pengguna. Pengguna memperoleh URL dan kode dari perangkat, kemudian beralih ke perangkat terpisah atau komputer dengan kemampuan masukan yang lebih. Pengguna membuka mesin pencari, menavigasi ke URL tertentu, melakukan *log in*, dan memasukkan kode.

Sementara itu, aplikasi jajak pendapat dari URL Google pada interval tertentu. Setelah pengguna menyetujui akses, respon dari server Google berisi token akses dan memperbaharui token. Aplikasi harus menyimpan token yang baru untuk penggunaan masa depan dan menggunakan token akses untuk mengakses Google API. Setelah masa token akses berakhir, maka aplikasi akan memperbaharui token untuk mendapatkan yang baru. Untuk gambar 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

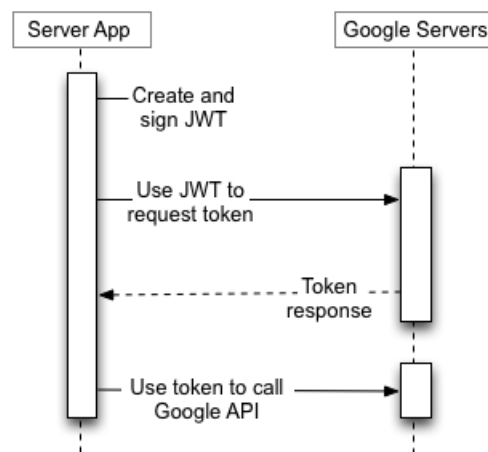
Google API seperti Prediction API dan Google Cloud Storage dapat bertindak atas nama aplikasi yang dibuat tanpa mengakses informasi pengguna. Dalam situasi ini aplikasi perlu membuktikan identitasnya sendiri ke API, tapi tidak diperlukan izin dari pihak pengguna. Demikian pula, dalam skenario perusahaan,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

Untuk jenis interaksi antara server memerlukan layanan akun, dimana akun tersebut

1 terdapat pada aplikasi yang dibuat, bukan individu ke pengguna akhir. Aplikasi memanggil
 2 Google API atas nama layanan akun, dan izin dari pihak pengguna tidak diperlukan. (Dalam
 3 skenario tanpa layanan akun, aplikasi memanggil Google API atas nama pengguna akhir,
 4 dan izin dari pihak pengguna kadang-kadang diperlukan.)

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

10 Kredensial layanan akun, yang diperoleh dari Google Developers Console, termasuk ala-
 11 mat email yang dihasilkan yang unik, klien id, dan setidaknya satu pasang kunci publik
 12 / privat. Menggunakan klien id dan satu kunci privat untuk membuat JWT (*JSON Web*
 13 *Token*) ditandatangani dan membangun permintaan token akses dalam format yang sesuai.
 14 Aplikasi kemudian mengirimkan permintaan token ke Google OAuth 2.0 Authorization Se-
 15 rver, yang mengembalikan token akses. Aplikasi menggunakan token untuk mengakses API
 16 Google. Ketika masa token berakhir, aplikasi mengulangi proses. Untuk gambar skenario
 17 dapat dilihat pada Gambar 2.5.



Gambar 2.5: Skenario Layanan Akun

18 2.1.3 Masa Habis Berlaku Token

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

- 22 • Pengguna telah mencabut akses.
- 23 • Token tidak digunakan selama enam bulan.
- 24 • Akun pengguna telah melampaui jumlah tertentu permintaan token.

25 Saat ini batas untuk setiap akun Google adalah 25 token. Jika pengguna akun telah
 26 memiliki 25 token, permintaan otentikasi untuk token ke-26 akan berhasil tapi token yang

1 paling tua atau token ke-1 akan dibuat tidak berlaku tanpa sepengetahuan pengguna. Jika
2 perlu untuk mengotorisasi beberapa program, mesin, atau perangkat, salah satu solusi ada-
3 lah untuk membatasi jumlah klien dimana harus mengotorisasi per pengguna akun antara
4 15 atau 20. Jika Anda adalah admin Google Apps, Anda dapat membuat admin tambahan
5 untuk mengizinkan beberapa klien.

6 2.1.4 Lingkup Otorisasi [2]

7 Lingkup disini merupakan sebuah *string* yang memungkinkan akses ke sumber daya ter-
8 tentu, misalnya akses ke data pengguna. Dengan memasukkan lingkup tertentu pada saat
9 permintaan otorisasi, kemudian mendapatkan izin sesuai dengan teks yang akan ditampilkan
10 ke pengguna. Setelah mendapat persetujuan dari pihak pengguna untuk izin atas lingkup
11 tersebut, maka Google mengirimkan token untuk aplikasi yang mengidentifikasi untuk mem-
12 berikan otorisasi khusus. Dengan kata lain, lingkup dan token menentukan apa saja data
13 pengguna yang diberi izin oleh pengguna untuk diakses.

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

21 2.1.4.1 Lingkup Profil

22 | `profile`

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

- 25 • Meminta agar aplikasi diberikan akses ke informasi profil dasar bagi pengguna yang
26 terotentikasi.
- 27 • Memungkinkan aplikasi untuk mengetahui siapa pengguna yang dikonfirmasi dengan
28 mengganti id pengguna dengan "*me*" yang mewakili pengguna yang telah teroten-
29 tikasi disetiap permintaan yang dilakukan.
- 30 • Memungkinkan aplikasi diakses melalui aplikasi android.

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

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

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

38 Lingkup ini juga memungkinkan lintas platform dengan pendaftaran tunggal.

1 2.1.4.2 Lingkup Email

2 | email

3 Lingkup ini meminta agar aplikasi diberikan akses ke:

- 4 • Alamat email Google dari pengguna. Mengakses alamat email dengan memanggil
5 people.get yang akan mengeluarkan *array* email atau dengan memanggil

6 | people.getOpenIdConnect

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

- 8 • Nama domain Google Apps jika ada yang dimiliki pengguna. Nama domain dikem-
9 balikan sebagai kepemilikan domain dari people.get atau properti hd dari getOpenI-
10 dConnect.

- 11 • Lingkup email ini setara dan menggantikan lingkup di bawah ini.

12 | https://www.googleapis.com/auth/userinfo.email

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

14 Lingkup ini meminta aplikasi agar diberikan akses ke:

- 15 • Alamat email Google pengguna yang telah diverifikasi di profil Google+. Mengakses
16 email dengan memanggil people.get dan mengembalikan hasil email dalam *array*.

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

18 2.1.4.3 Lingkup yang lain

19 | openid

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

23 Metode getOpenIdConnect mengembalikan profil pengguna dengan format OIDC meng-
24 ikuti jalur permintaan HTTP:

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

26 Untuk keperluan login menggunakan lingkup profil atau lingkup

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

28 karena lingkup

29 | https://www.googleapis.com/auth/plus.me

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

32 Lingkup ini melakukan hal berikut:

- 33 • Memungkinkan aplikasi untuk mengetahui siapa pengguna yang dikonfirmasi dengan
34 mengganti id pengguna dengan "me" yang mewakili pengguna yang telah teroten-
35 tikasi disetiap permintaan yang dilakukan.

2.1.4.4 Lingkup yang tidak dipakai lagi

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

Ganti dengan lingkup yang setara yaitu lingkup profil. Lingkup ini setara dengan lingkup profil dan meminta akses data yang sama. Catatan: lingkup ini tidak dipakai lagi namun tetap dipertahankan dan terus tersedia untuk kompatibilitas.

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

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

Lihat juga lingkup terkait:

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

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

2.2 Markdown

2.2.1 Apa itu Markdown? [3]

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

2.2.2 Sintaks yang Berguna [3]

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

2.2.2.1 Cetak Tebal dan Cetak Miring

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

```
*hello* untuk cetak miring
**hello** untuk cetak tebal
```

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

```

1 | <p><em>hello </em> untuk cetak miring</p>
2 |
3 | <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.

```

9 | # Judul Bab
10 |
11 | ## Judul Sub Bab
12 |
13 | ### Judul Sub Sub Bab
14 |
15 | #### Tingkat ke 4
16 |
17 | ##### Tingkat ke 5
18 |
19 | ##### Tingkat ke 6

```

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

```

22 | <h1>Judul Bab</h1>
23 |
24 | <h2>Judul Sub Bab</h2>
25 |
26 | <h3>Judul Sub Sub Bab</h3>
27 |
28 | <h4>Tingkat ke 4</h4>
29 |
30 | <h5>Tingkat ke 5</h5>
31 |
32 | <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.

```
1 | Baris ini dengan
2 | batas baris
```

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

```
6 | <p>Baris ini dengan <br />
7 | batas baris</p>
```

Baris ini dengan
batas baris

Gambar 2.8: Markdown Batas Baris Dengan Tiga Spasi

```
8 | Baris ini tanpa
9 | batas baris
```

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

```
13 | <p>Baris ini tanpa
14 | batas baris</p>
```

Baris ini tanpa batas baris

Gambar 2.9: Markdown Batas Baris Tanpa Tiga Spasi

15 2.2.2.4 Paragraf

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

```
18 | Ini kalimat pertama. Ini kalimat berikutnya. Ini kalimat terakhir.
19 |
20 | Ini paragraf baru.
```

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

```
24 | <p>Ini kalimat pertama. Ini kalimat berikutnya. Ini kalimat terakhir.</p>
25 |
26 | <p>Ini paragraf baru.</p>
```

Ini kalimat pertama. Ini kalimat berikutnya. Ini kalimat terakhir.
Ini paragraf baru.

Gambar 2.10: Markdown Paragraf

27 2.2.2.5 Gambar

28 Untuk menyisipkan gambar pada dokumen Markdown, gunakan sintaks berikut

```
29 | ![ teks ]( url_gambar "judul_gambar")
```

1 Contoh penggunaan sintaks gambar:

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

4 Url gambar dapat diganti dengan path yang mengarah pada *file* gambar yang akan digu-
 5 nakan. Sintaks tersebut akan menghasilkan kode HTML sebagai berikut. Untuk hasil pada
 6 HTML dapat dilihat pada Gambar 2.11.

```
7 | <p></p>
```



Gambar 2.11: Markdown Menampilkan Gambar

9 2.2.2.6 Link

10 Untuk menyisipkan *hyperlink* pada dokumen Markdown, gunakan sintaks berikut

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

12 Contoh penggunaan sintaks link:

```
13 | [[my_website](http://{it browser}native.com "Click Here")]
```

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

```
16 | <p><a href="http://{it browser}native.com" title="Click Here">my_website</a></p>
```



Gambar 2.12: Markdown Link

17 2.2.2.7 Kode

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

```
22 | Definisi dari 'initLabels()' dapat dilihat di bawah ini"
23 |
24 | '''
25 |     function initLabels(){
26 |         function setLabels(elementId, messageId){
27 |             var label = document.querySelector('label[for=' + elementId + ']');
28 |             label.textContent = chrome.i18n.getMessage(messageId);
```

```

1  |         }
2  |     }
3  | '''
4  |
5  | Paragraf lain .
6
7  'initLabels()' akan menjadi kode pada satu baris kalimat. Sedangkan fungsi yang ada di-
8  lam ''' akan menjadi blok kode. Sintaks tersebut akan menghasilkan kode HTML sebagai
9  berikut. Untuk hasil pada HTML dapat dilihat pada Gambar 2.13.
10
11 <p>Definisi dari 'initLabels()' dapat dilihat di bawah ini"</p>
12
13 <p>'''
14     function initLabels(){
15         function setLabels(elementId, messageId){
16             var label = document.querySelector('label[for=' + elementId + ']');
17             label.textContent = chrome.i18n.getMessage(messageId);
18         }
19     }
20 '''</p>
21
22 <p>Paragraf lain.</p>
23
24 Definisi dari 'initLabels()' dapat dilihat di bawah ini"
25
26 ''' function initLabels(){ function setLabels(elementId, messageId){ var label = document.querySelector('label[for=' + elementId + ']'); label.textContent =
27 chrome.i18n.getMessage(messageId); } } '''
28
29 Paragraflain.

```

Gambar 2.13: Markdown Kode

21 2.2.2.8 Kutipan

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

```

24 | Ini merupakan kutipan **favorit** saya:
25 |
26 | > The weak can never forgive.
27 | > Forgiveness is the attribute of the strong.
28
29 Sintaks tersebut akan menghasilkan kode HTML sebagai berikut. Untuk hasil pada HTML
30 dapat dilihat pada Gambar 2.14.
31
32 <p>Ini merupakan kutipan <strong>favorit</strong> saya:</p>
33
34 <blockquote>
35 | <p>The weak can never forgive.
36 | Forgiveness is the attribute of the strong.</p>
37 </blockquote>

```

Ini merupakan kutipan favorit saya:

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

Gambar 2.14: Markdown Kutipan

36 2.2.2.9 Garis Horisontal

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

```

39 | # Bagian Satu
40 |
41 | The quick brown fox jumps over the lazy dog. The quick brpwn fox jump over the lazy dog.
42 |
43 | ---
44 |
45 | Last Edited on *25th Dec 2014*

```

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

```

3 | <h1>Bagian Satu</h1>
4 |
5 | <p>The quick brown fox jumps over the lazy dog. The quick brpwn fox jump over the lazy dog.</p>
6 |
7 | <hr />
8 |
9 | <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

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

13 (1) Daftar tidak berurutan

- 14 Untuk membuat daftar tidak berurutan dapat menggunakan simbol bintang *, simbol
 15 tambah +, maupun tanda hubung - sebelum daftar item yang ingin dimasukkan. Untuk
 16 contoh penggunaan dapat dilihat di bawah ini.

```

17 | * Item
18 | * Item
19 | * Item
20 |
21 | + Item
22 | + Item
23 | + Item
24 |
25 | - Item
26 | - Item
27 | - Item
```

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

```

30 | <ul>
31 | <li>Item</li>
32 | <li>Item</li>
33 | <li><p>Item</p></li>
34 | <li><p>Item</p></li>
35 | <li>Item</li>
36 | <li><p>Item</p></li>
37 | <li><p>Item</p></li>
38 | <li>Item</li>
39 | <li>Item</li>
40 | </ul>
```


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

Gambar 2.16: Daftar Tidak Berurutan

1 (2) Daftar berurutan

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

```
4 | 1. Item 1
5 | 2. Item 2
6 | 3. Item 3
```

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

```
9 | <ol>
10 | <li>Item 1</li>
11 | <li>Item 2</li>
12 | <li>Item 3</li>
13 | </ol>
```

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

Gambar 2.17: Markdown Daftar Berurutan

14 2.2.3 GitHub Flavored Markdown [4]

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

19 2.2.3.1 Beberapa Garis Bawah Pada Kalimat

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

```
23 | wow_great_stuff
24 | do_this_and_do_that_and_another_thing.
```

- 1 Hal tersebut memungkinkan untuk merender kode dan nama dengan benar. Untuk me-
 2 nekankan sebagian kata dapat menggunakan tanda bintang (*). Untuk hasil pada HTML
 3 dapat dilihat pada Gambar 2.18.

`wow_great_stuff do_this_and_do_that_and_another_thing.`

Gambar 2.18: GFM Garis Bawah

4 2.2.3.2 Taut Otomatis URL

- 5 GFM membuat standar untuk taut otomatis URL. Tanpa mengatur teks untuk link se-
 6 buah URL, cukup menyisipkan URL dan URL tersebut akan menjadi taut otomatis yang
 7 mengarah ke URL tersebut.

8 `| http://example.com`

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

`http://www.google.com`

Gambar 2.19: GFM Taut Otomatis URL

11 2.2.3.3 Tanda Coret

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

14 `| ~Mistaken text.~`

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

`Mistaken text.`

Gambar 2.20: GFM Tanda Coret

16 2.2.3.4 Blok Kode

- 17 SM mengkonferensi blok kode dengan diawali empat spasi untuk setiap baris yang berada
 18 dalam blok kode. GFM juga mendukung blok kode namun cukup membungkus kode dengan
 19 tiga kutip belakang (``) tanpa harus memperhatikan empat spasi untuk awalan kode.

```
20 | Here's an example:
21 |
22 | ```
23 | function test() {
24 |   console.log("notice the blank line before this function?");
25 | }
26 | ```
```

- 27 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

2.2.3.5 Penandaan Sintaks

Blok kode dapat dilanjutkan dengan menambah sintaks. Dalam blok yang ditandai bahkan sebuah indentifikasi bahasa apa yang digunakan. Misalnya penandaan kode untuk sintaks Ruby.

```
5 | '''ruby
6 | require 'redcarpet'
7 | markdown = Redcarpet.new("Hello World!")
8 | puts markdown.to_html
9 | '''
```

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

2.2.3.6 Tabel

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

```
15 | First Header | Second Header
16 | -----
17 | Content Cell | Content Cell
18 | Content Cell | Content Cell
```

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

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

```
22 | | First Header | Second Header |
23 | | ----- |
24 | | Content Cell | Content Cell |
25 | | Content Cell | Content Cell |
```

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

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

```

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

```

- 7 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

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

```

10 | | Name | Description |
11 | |-----|-----|
12 | | Help | ~Display the~ help window. |
13 | | Close | _Closes_ a window |

```

- 14 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

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

```

17 | | Left-Aligned | Center Aligned | Right Aligned |
18 | | :-----: | :-----: | :-----: |
19 | | col 3 is | some wordy text | $1600 |
20 | | col 2 is | centered | $12 |
21 | | zebra stripes | are neat | $1 |

```

- 22 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>
```

Tabel 2.1: Daftar Pengujian Zurb Foundation¹

<i>Browser/OS</i>	<i>The Grid</i>	<i>Layout/UI</i>	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 Strapdonw.js juga memiliki beberapa fitur :

- 2 (1) Ramah dengan mesin pencari
- 3 (2) Kompatibel dengan berbagai *browser* (Sudah diuji dengan ponsel menggunakan Safari,
- 4 IE 8/9, Firefox, Chrome)
- 5 (3) Github menggunakan Markdown (Tabel, Sintaks, Judul Bab)
- 6 (4) Dapat menggunakan tema

7 2.4 Zurb Foundation [6]

8 Zurb Foundation merupakan alat bantu dalam membuat aplikasi baru maupun membuat
 9 *website* yang responsif. Jutaan desainer dan teknisi menggunakan Foundation sebagai bagian
 10 dari alur kerja mereka. Zurb Foundation adalah *framework* pertama yang memperkenalkan
 11 konsep responsif, semantik, mobile dan parsial. Zurb Foundation juga kompatibel dengan
 12 kebanyakan mesin pencari dan perangkat. Maka dari itu Zurb Foundation merupakan pilihan
 13 profesional bagi para desainer dan teknisi.

14 2.4.1 Kompatibilitas

15 Zurb Foundation dirancang dan diuji pada berbagai *browser* dan perangkat. Daftar pengujian
 16 pada berbagai *browser* dan perangkat dapat dilihat pada Tabel 2.1.

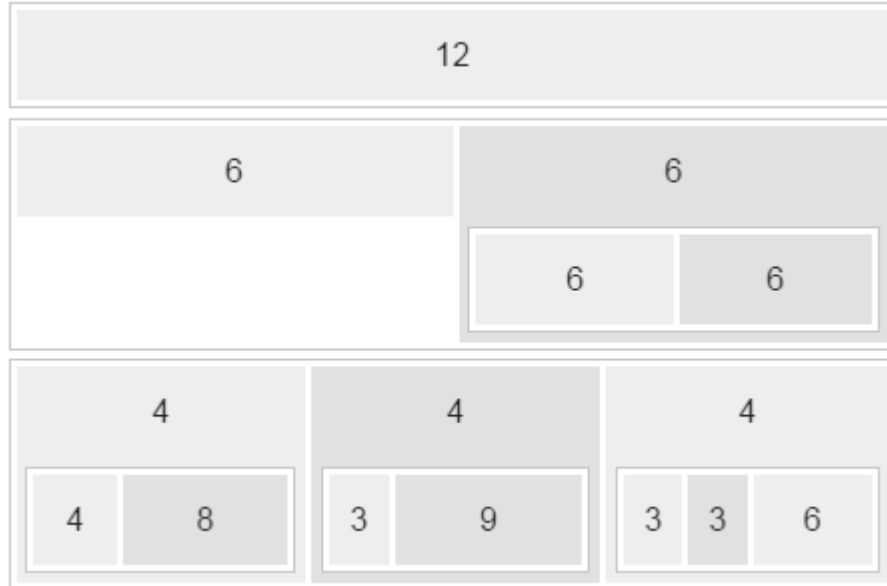
17 2.4.2 Apa Saja yang Dapat Dibuat dengan Foundation?

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

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

1 (1) *Grid*

2 *Grid* bekerja pada hampir semua perangkat dan memiliki dukungan untuk menjadi satu
 3 kesatuan, sumber pemesanan, *offset* dan perangkat presentasi. Hal tersebut sedikit
 4 mudah dengan waktu yang singkat dapat menciptakan tata letak yang kompleks seperti
 5 ini. Untuk contoh *grid* dapat dilihat pada Gambar 2.28.

Gambar 2.28: Contoh *Grid* Zurb Foundation6 • *Grid* Utama

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

```

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

```

```

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

```

18 <!-- no. 1 -->
19 <div class="row">
20     <div class="small-2 columns">2 columns</div>
21     <div class="small-10 columns">10 columns</div>
22 </div>
23 <!-- no. 2 -->
24 <div class="row">
25     <div class="small-3 columns">3 columns</div>
26     <div class="small-9 columns">9 columns</div>
27 </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.

```

33 <!-- no. 1 -->
34 <div class="row">
35     <div class="medium-2 columns">2 columns</div>
36     <div class="medium-10 columns">10 columns</div>
37 </div>
38 <!-- no. 2 -->
39 <div class="row">
40     <div class="medium-3 columns">3 columns</div>
41     <div class="medium-9 columns">9 columns</div>
42 </div>

```


1.	2 columns	10 columns
2.	3 columns	9 columns

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.

```

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

```

8	4
8 Nested	4
8 Nested Again	4

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>

```

1.	1	11
2.	1	10, offset 1
3.	1	9, offset 2
4.	1	8, offset 3

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

```

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

```

1.	3	3		3
2.	3	3	3 end	

Gambar 2.34: *Grid* Dengan Baris yang Tidak Lengkap

- *Grid* Dengan Baris yang Mengempis/Tidak Mengempis

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

```

26     <div class="row medium-uncollapse large-collapse">
27         <div class="small-6 columns">
28             Removes gutter at large media query
29         </div>
30         <div class="small-6 columns">
31             Removes gutter at large media query
32         </div>
33     </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

- *Grid* Dengan Kolom yang Berpusat

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

```

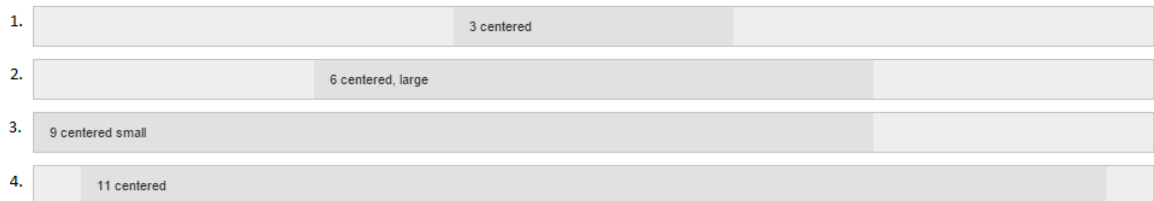
41     <!-- no. 1 -->
42     <div class="row">
43         <div class="small-3 small-centered columns">3 centered</div>

```

```

1      </div>
2      <!-- no. 2 -->
3      <div class="row">
4          <div class="small-6 large-centered columns">6 centered</div>
5      </div>
6      <!-- no. 3 -->
7      <div class="row">
8          <div class="small-9 small-centered large-uncentered columns">9 centered</div>
9      </div>
10     <!-- no. 4 -->
11     <div class="row">
12         <div class="small-11 small-centered columns">11 centered</div>
13     </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.

```

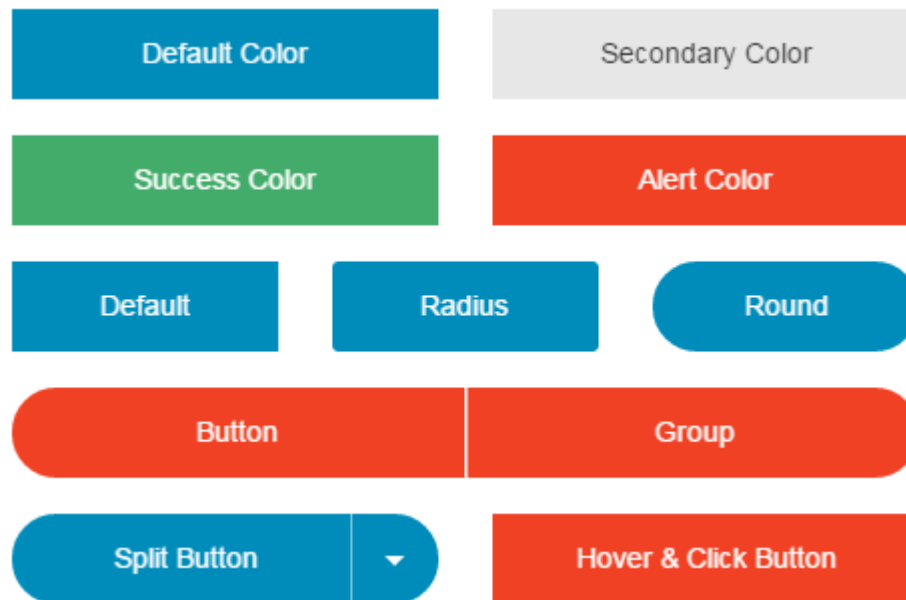
20     <!-- no. 1 -->
21     <div class="row">
22         <div class="small-10 small-push-2 columns">10</div>
23         <div class="small-2 small-pull-10 columns">2, last</div>
24     </div>
25     <!-- no. 2 -->
26     <div class="row">
27         <div class="large-9 large-push-3 columns">9</div>
28         <div class="large-3 large-pull-9 columns">3, last</div>
29     </div>
30     <!-- no. 3 -->
31     <div class="row">
32         <div class="large-8 large-push-4 columns">8</div>
33         <div class="large-4 large-pull-8 columns">4, last</div>
34     </div>
35     <!-- no. 4 -->
36     <div class="row">
37         <div class="small-5 small-push-7 medium-7 medium-push-5 columns">7</div>
38         <div class="small-7 small-pull-5 medium-5 medium-pull-7 columns">5, last</div>
39     </div>
40     <!-- no. 5 -->
41     <div class="row">
42         <div class="medium-6 medium-push-6 columns">6</div>
43         <div class="medium-6 medium-pull-6 columns">6, last</div>
44     </div>

```

Gambar 2.37: *Grid* Dengan Sumber yang Terurut

1 (2) Tombol

2 Mengklik tombol dengan material yang bagus merupakan hal yang mengagumkan. Meng-
 3 klik tombol juga menghubungkan pengguna dengan berbagai aksi. Ada beberapa gaya
 4 tombol yang ringan untuk ukuran, presentasi, dan warna untuk menyesuaikan tombol
 5 Anda sendiri semudah menambahkan kelas. Untuk contoh macam-macam tombol dapat
 6 dilihat pada Gambar 2.38.



Gambar 2.38: Contoh Tombol Zurb Foundation

7 • Tombol Utama

8 Dapat membuat tombol utama dengan kode HTML sebagai berikut dan dapat
 9 dilihat hasilnya pada Gambar 2.39.

10 | `Default Button`

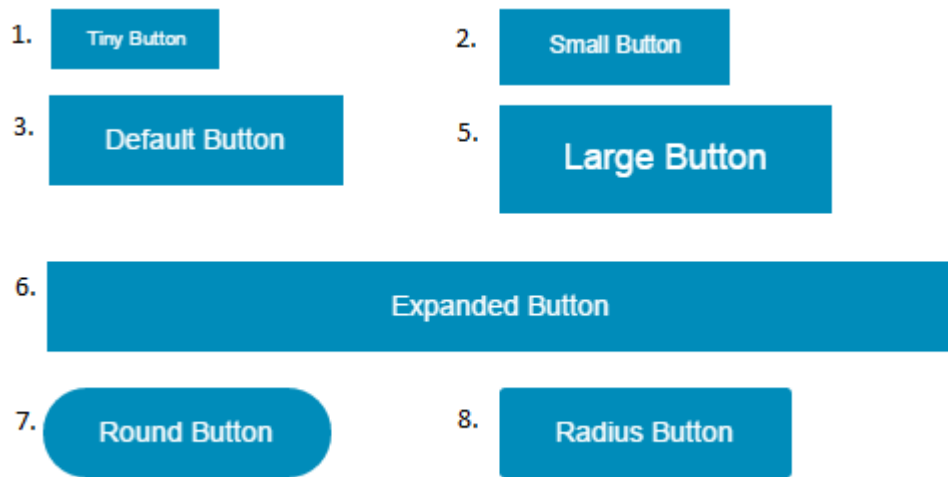


Gambar 2.39: Tombol Dasar

11 • Mengubah Ukuran Tombol

12 Terdapat kelas tambahan untuk merubah bentuk dan ukuran tombol, berikut kode
 13 HTML yang dapat digunakan dan dapat dilihat hasilnya pada Gambar 2.40.

14 | `<!-- Kelas untuk merubah ukuran -->`
 15 | `1. Tiny Button`
 16 | `2. Small Button`
 17 | `3. Default Button`
 18 | `4. Disabled Button`
 19 | `5. Large Button`
 20 | `6. Expanded Button`
 21 | `<!-- Kelas untuk merubah bentuk -->`
 22 | `7. Round Button`
 23 | `8. Radius Button`



Gambar 2.40: Ukuran dan Bentuk Tombol

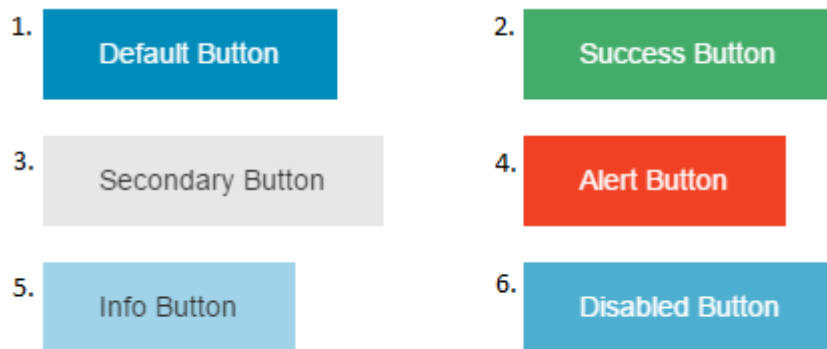
- Warna Tombol

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

```

4      <!-- Kelas untuk merubah warna -->
5      1. <a href="#" class="button">Default Button</a>
6      2. <a href="#" class="button success">Success Button</a>
7      3. <a href="#" class="button secondary">Secondary Button</a>
8      4. <a href="#" class="button alert">Alert Button</a>
9      5. <a href="#" class="button info">Info Button</a>
10     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.

```

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

```



Gambar 2.42: Aksesibilitas Tombol

1 (3) Tabel

2 Dapat membuat tabel dengan menggunakan markup yang minim. Berikut contoh peng-
 3 gunaan 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
23 metus.</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
30 metus.</td>
31 <td>Content Goes Here</td>
32 <td>Content Goes Here</td>
33 </tr>
34 </tbody>
35 </table>

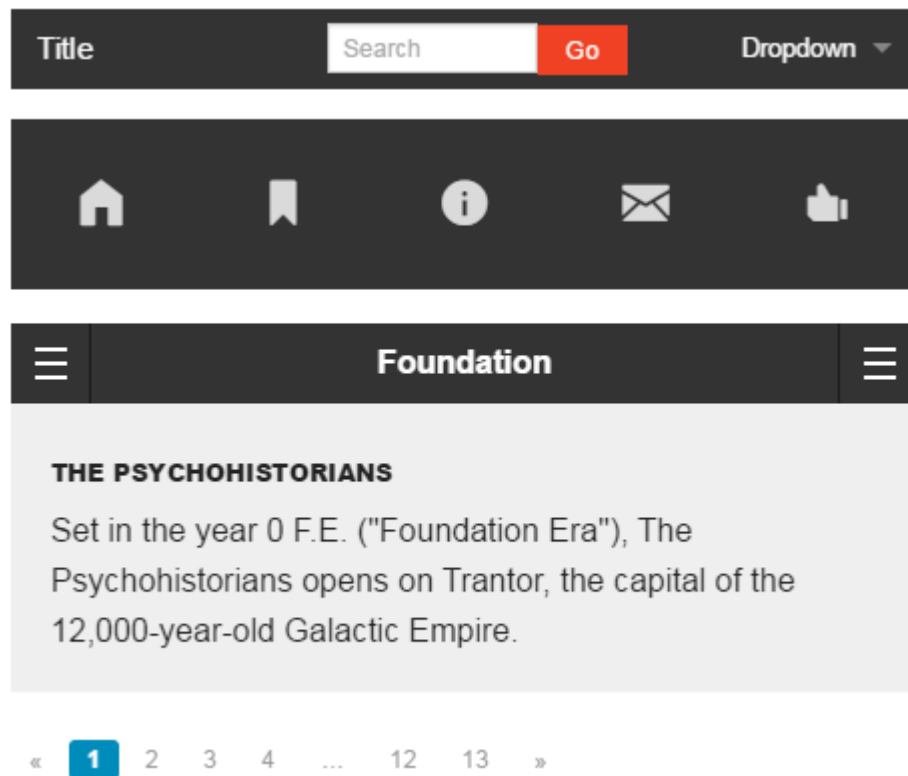
```

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

36 (4) Navigasi

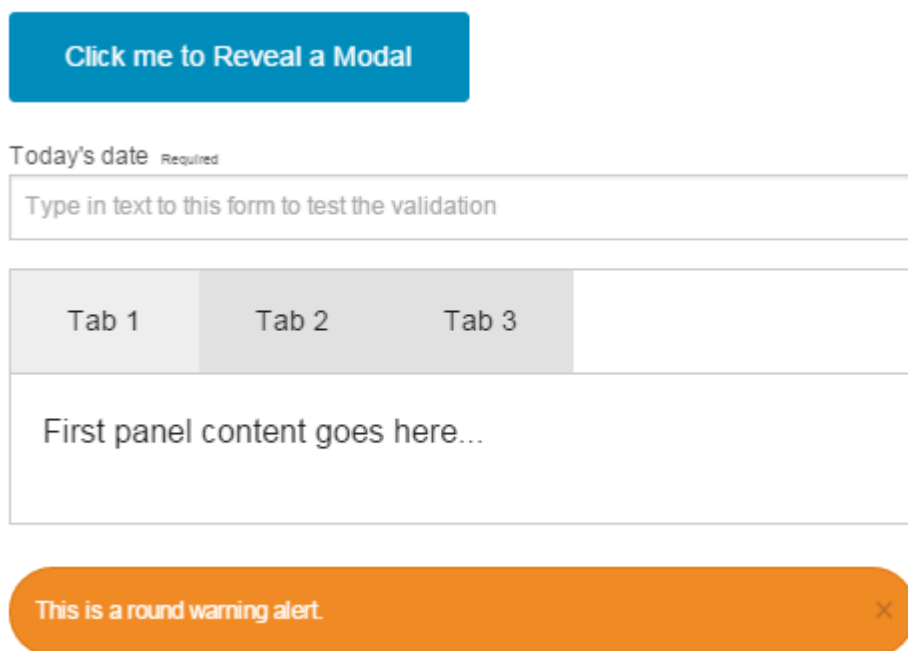
37 Orang yang mengakses harus bisa berkeliling melihat menu-menu yang ada. Gaya na-
 38avigasi pada Foundation meliputi : bar bagian atas yang kuat dengan menu dropdown;
 39 tombol; bar pencari; ikon bar yang keren; implementasi canvas yang lepas dari keluhan;
 40 dan sekelompok navigasi lainnya. Untuk contoh macam-macam navigasi dapat dilihat
 41 pada Gambar 2.44.



Gambar 2.44: Contoh Navigasi Zurb Foundation

1 (5) *Plugins*

2 Sudah meliputi banyak *plugin* JavaScript yang ditulis untuk modal dasar *pop-up*; me-
 3 nambat formulir validasi yang diperlukan; membuat *tab* konten; tanda peringatan; dan
 4 masih banyak lagi. Untuk contoh macam-macam *plugin* dapat dilihat pada Gambar
 5 [2.45](#).



Gambar 2.45: Contoh Plugins Zurb Foundation

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

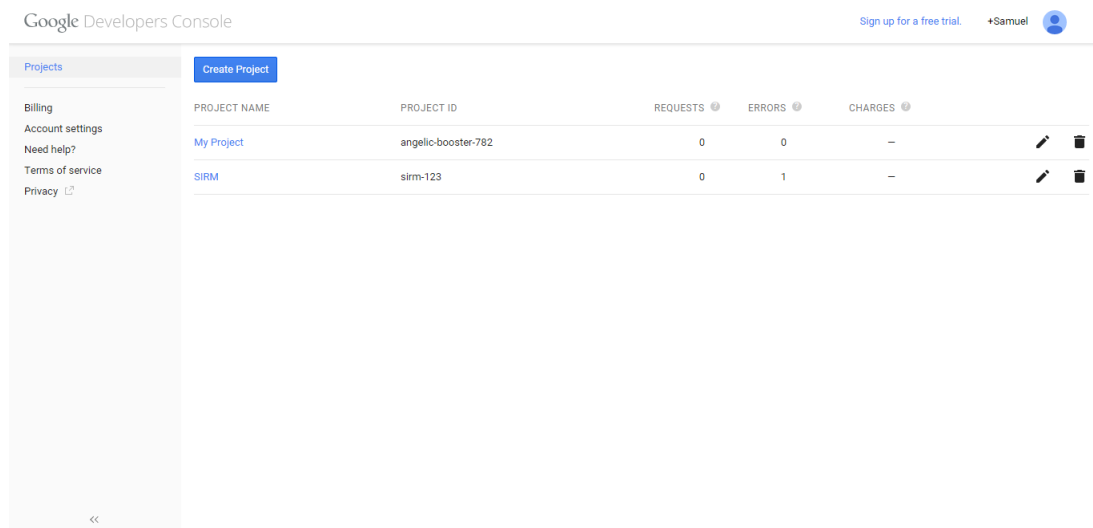
Setelah mendapatkan token akses untuk mengirimkannya ke API maka diperlukan ruang lingkup. Karena sesuai dengan landasan teori, jika token akses dikeluarkan untuk Google+ API maka token akses tersebut tidak berlaku untuk mengakses Google Contact API. Ruang lingkup yang akan digunakan pada penelitian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

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

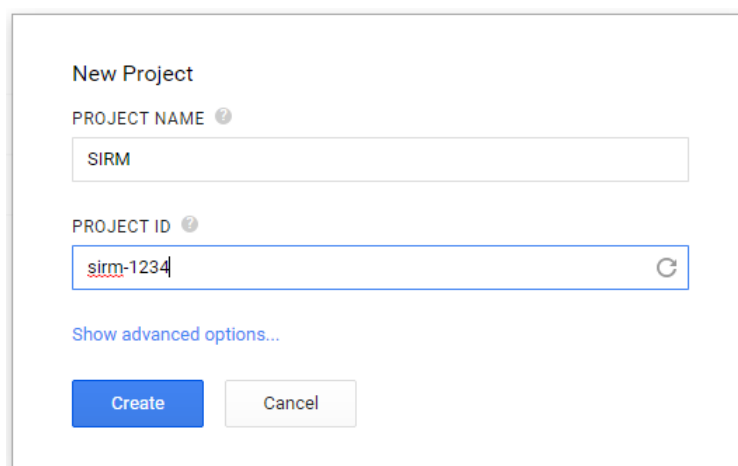
karena pada penelitian ini dibutuhkan email pengguna dan nama pengguna, keduanya itu dapat diperoleh dengan menggunakan kedua scope tersebut. Scope pertama untuk mendapatkan info dari pengguna dan scope kedua untuk mendapatkan email dari pengguna.

(4) Memperbaharui token akses jika diperlukan

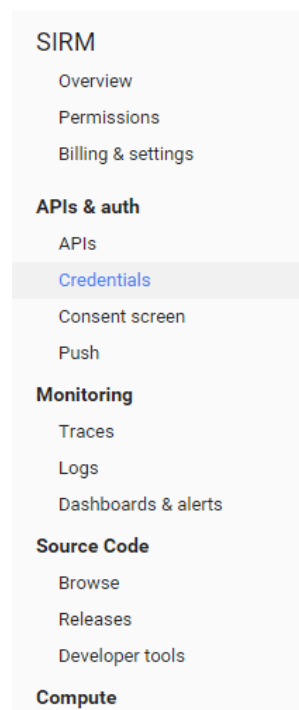
Pada penelitian ini tidak akan menggunakan tahap memperbaharui token akses karena token akses hanya digunakan selama penelitian ini berlangsung.



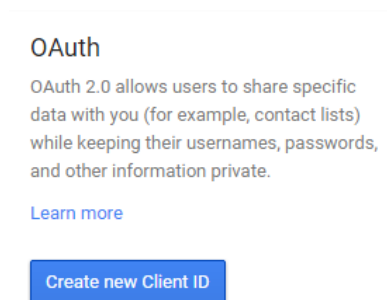
Gambar 3.1: Google Developers Console



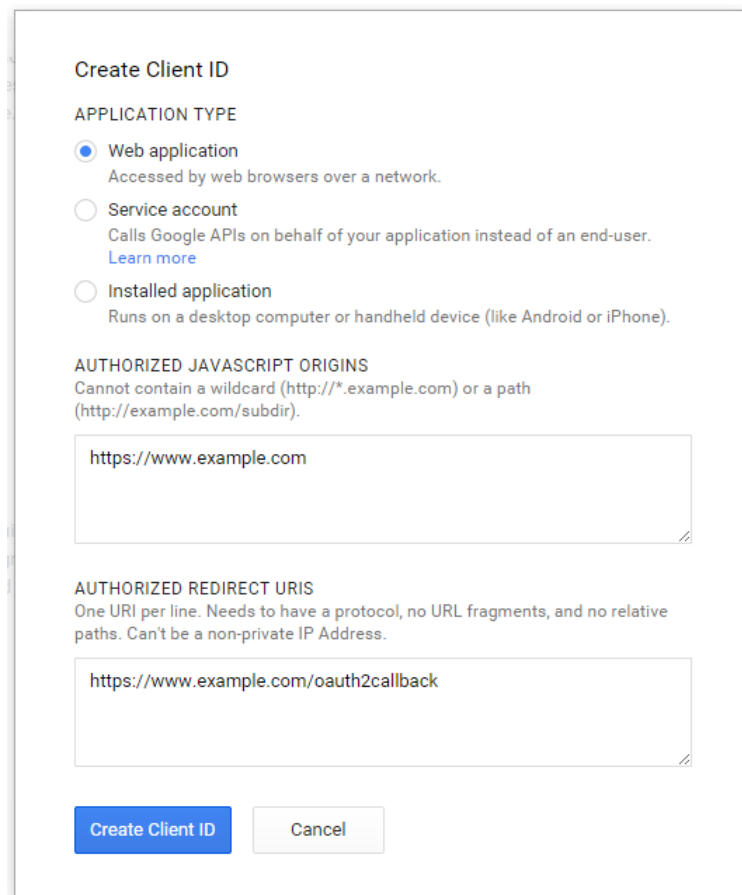
Gambar 3.2: Membuat Proyek Baru



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`).

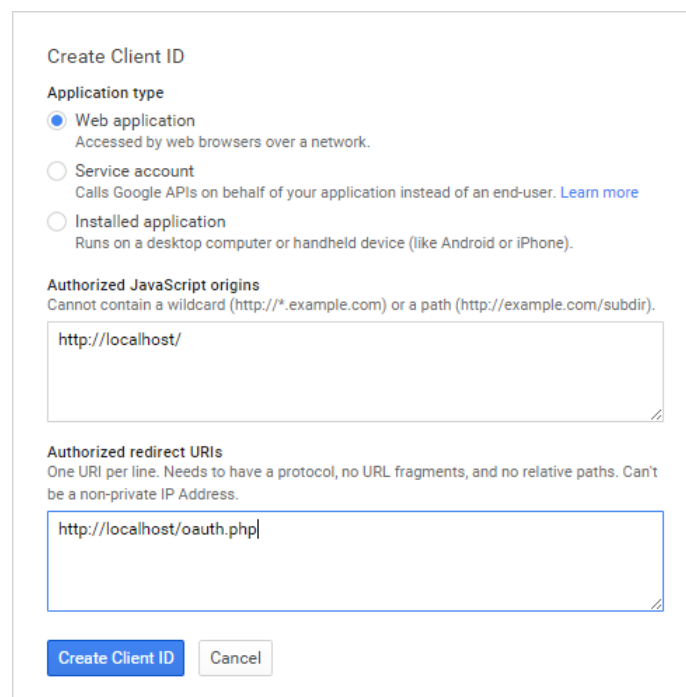
`https://www.example.com`

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**

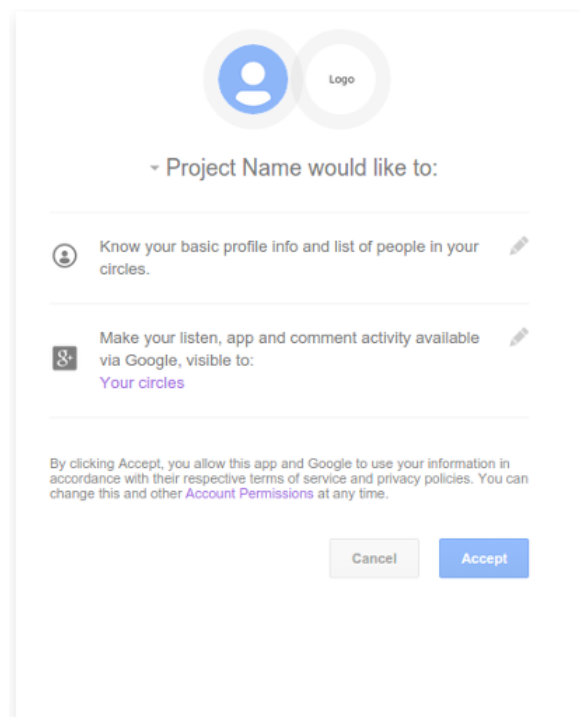
Gambar 3.6: Pengisian Tipe Aplikasi

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

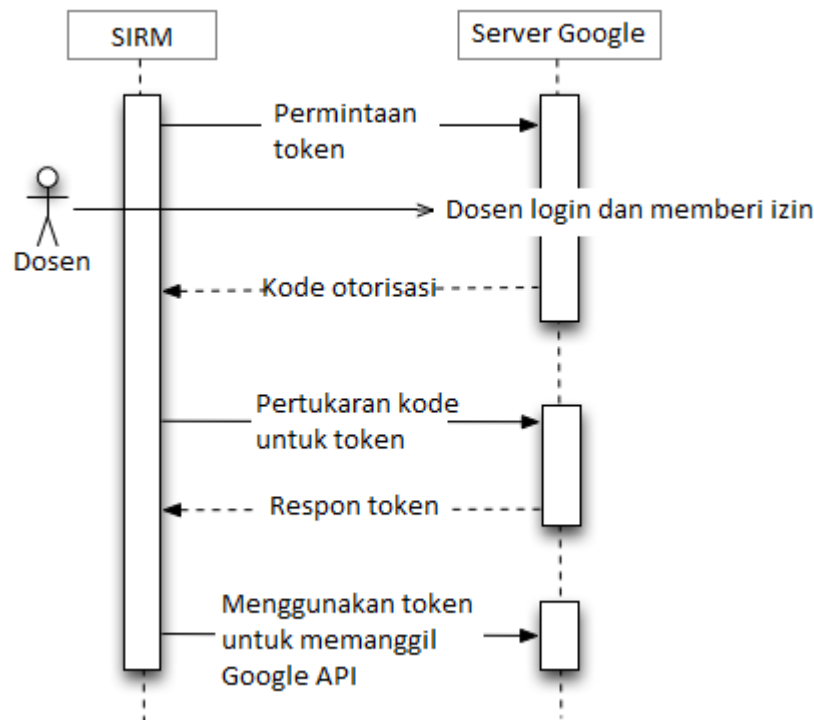
Gambar 3.7: Client ID



Gambar 3.8: Izin Pihak Pengguna

1 3.1.2 Skenario Aplikasi

2 Berdasarkan landasan teori skenario yang ada pada Sub Bab 2.1 dan berdasarkan perangkat
3 lunak 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

1 3.2 Analisis Markdown

2 Sintaks Markdown yang akan digunakan sesuai dengan landasan teori pada Sub Bab 2.2.
 3 Sintaks Markdown akan digunakan pada bagian keterangan mahasiswa agar seragam dalam
 4 penulisannya. Keterangan mahasiswa yang akan ditampilkan antara lain; NPM, nama,
 5 deskripsi umum, catatan. Maka dari itu sintaks Markdown yang akan digunakan adalah
 6 Cetak Tebal dan Cetak Miring, Judul Bab, Batas Baris, Paragraf, *Link*, dan Daftar.

7 • Sintaks Cetak Tebal dan Cetak Miring

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

10 | ****NPM**** – *2010730013*

11 • Sintaks Judul Bab

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

```

15 | # Judul 1
16 | ## Judul 2
17 | ### Judul 3
18 | #### Judul 4
19 | ##### Judul 5
20 | ##### Judul 6
  
```

21 • Sintaks Batas Baris

22 Sintaks ini digunakan pada penulisan paragraf jika diperlukan untuk mengakhiri se-
 23 buah baris atau ingin membuat baris baru. Berikut penggunaan sintaks dan hasilnya
 24 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

Berikut penggunaan sintaks Markdown secara keseluruhan untuk bagian keterangan mahasiswa. 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://bleack.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

3.3 Analisis StrapdownJS

StrapdownJS digunakan untuk menampilkan sintaks Markdown ke halaman HTML. Pada penelitian ini strapdown.js terlebih dahulu diunduh dan untuk menggunakannya menggunakan *path* seperti di bawah ini.

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

Skrip tersebut disisipkan pada skrip view.php dan past.php yang berfungsi untuk menampilkan info mahasiswa yang dimana info tersebut ditulis menggunakan sintaks Markdown. Berikut skrip view.php yang menggunakan strapdown.js.

```
<!doctype html>
<html class="no-js" lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>SIRM | Welcome</title>
    <link rel="stylesheet" href="css/foundation.css" />
    <script src="js/vendor/modernizr.js"></script>
  </head>
  <body>
    <div class="row">
      <h5>Anda melihat catatan mahasiswa ini sebagai test@unpar.ac.id.</h5>
    </div>
    <div class="row">
      <ul class="button-group">
        <li><a href="editmahasiswa.php" class="button">Edit</a></li>
        <li><a href="lihathistori.php" class="button">Lihat Histori</a></li>
      </ul>
    </div>
    <hr/>
    <xmp style="display:none;">
    ### 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://bleack.
    blogspot.com/](http://bleack.blogspot.com/).
    ### Catatan
    * 9 Oktober 2014, bimbingan skripsi
    * 3 Oktober 2014, bimbingan skripsi
    * 1 September 2014, perwalian
    * 1 September 2014, pertama kali dibuat
    </xmp>
    <script src="js/0.2/strapdown.js"></script>
```

```

1 | </body>
2 | </html>

```

3 Untuk baris 30 sampai baris 50 pada skrip view.php akan diambil dari database.

4 3.4 Analisis Zurb Foundation

5 Zurb Foundation digunakan untuk membuat tampilan antarmuka aplikasi yang akan diba-
6 ngun. Sesuai landasan teori pada Sub Bab 2.4, pada aplikasi ini menggunakan dua bagian
7 yaitu *Grid* dan Tombol. *Grid* digunakan untuk mengatur pembagian tata letak kompleks
8 sehingga terlihat rapih. Tombol digunakan untuk merubah tombol yang biasa menjadi lebih
9 enak untuk dilihat. Berikut sintaks penggunaan *Grid* dan Tombol pada list.php dan untuk
10 gambar dapat dilihat pada Gambar 3.17.

```

11 | <!doctype html>
12 | <html class="no-js" lang="en">
13 |     <head>
14 |         <meta charset="utf-8" />
15 |         <meta name="viewport" content="width=device-width, initial-scale=1.0" />
16 |         <title>SIRM | List</title>
17 |         <link rel="stylesheet" href="css/foundation.css" />
18 |         <script src="js/vendor/modernizr.js"></script>
19 |     </head>
20 |     <body>
21 |         <?php
22 |             session_start();
23 |         ?>
24 |         <div class="row">
25 |             <h3>Pilih NPM yang ingin dicari / tambah baru.</h3>
26 |             <ul class="button-group">
27 |                 <li><a href="new.php" class="button secondary">Add</a></li>
28 |                 <li><a href="index.php?logout" class="button secondary">Logout</a></li>
29 |             </ul>
30 |             <hr/>
31 |         </div>
32 |
33 |         <div class="row">
34 |             <?php
35 |                 include_once "configDatabase.php";
36 |
37 |                 if(! $id_mysql)
38 |                 {
39 |                     die("Database tidak bisa dibuka");
40 |                 }
41 |
42 |                 if(! mysql_select_db("sirm", $id_mysql))
43 |                 {
44 |                     die("Database tidak bisa dipilih");
45 |                 }
46 |
47 |                 $hasil = mysql_query("SELECT * FROM info_mahasiswa", $id_mysql);
48 |
49 |                 if(! $hasil)
50 |                 {
51 |                     die("Permintaan gagal");
52 |                 }
53 |
54 |                 echo "<table>
55 |                 <thead>
56 |                 <tr>
57 |                     <th width='250'>NPM</th>
58 |                     <th width='500'>Nama</th>
59 |                     <th width='250'>Last Update</th>
60 |                 </tr>
61 |                 </thead>";
62 |
63 |                 while($row = mysql_fetch_array($hasil))
64 |                 {
65 |                     echo "<tr>";
66 |                     echo "<td><a href='view.php?npm=". $row['npm'] .' '>" . $row['npm'] . "</a></td>";
67 |                     echo "<td>" . $row['nama'] . "</td>";
68 |                     echo "<td>" . $row['pembaruan_terakhir'] . "</td>";
69 |                     echo "</tr>";
70 |                 }
71 |                 echo "</table>";
72 |             ?>
73 |         </div>

```

```

1 | </body>
2 | </html>

```

Pilih NPM yang ingin dicari / tambah baru.

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

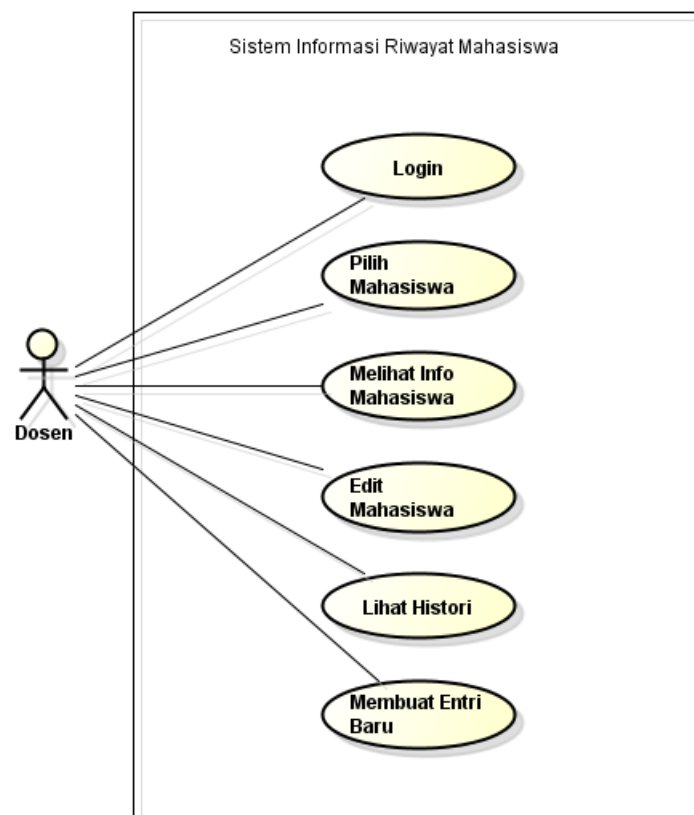
NPM	Nama	Last Update
2003730013	Pascal	2015-05-05 09:26:15
2010120031	Kenneth Natanael	2015-04-30 14:29:51
2010730001	Andri Agustian	2015-04-29 15:09:33
2010730005	Grady Ireneus	2015-04-30 14:31:19
2010730012	Kevin PL	2015-04-24 00:55:57
2010730013	Samuel Herman	2015-05-08 00:37:01

Gambar 3.17: Tampilan pilihmahasiswa.php dengan Zurb Foundation

3.5 Analisis Kebutuhan Perangkat Lunak

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

3.5.1 Use Case Diagram



Gambar 3.18: *Use Case* Diagram Sistem Informasi Riwayat Mahasiswa

Use case diagram merupakan pemodelan yang menunjukkan kegiatan apa saja yang dapat dilakukan pengguna dan kegiatan yang dilakukan sistem. Berikut adalah deskripsi dari use

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 case pada Gambar 3.18.

2 • Login

3 *Use case* ini memungkinkan pengguna untuk login via Google OAuth.

4 • Pilih Mahasiswa

5 *Use case* ini memungkinkan pengguna untuk memilih mahasiswa yang ingin dilihat
6 infonya. Selain itu pengguna juga bisa menemukan tombol "*Add*" untuk menambah
7 entri baru.

8 • Melihat Info Mahasiswa

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

12 • *Edit* Mahasiswa

13 *Use case* ini memungkinkan pengguna untuk mengubah info mahasiswa yang sudah
14 ada.

15 • Lihat Histori

16 *Use case* ini memungkinkan pengguna untuk melihat histori untuk setiap perubahan
17 dan aksi yang dilakukan pengguna.

18 • Membuat Entri Baru

19 *Use case* ini memungkinkan pengguna untuk membuat entri baru dengan memasukan
20 inputan pada *form* yang telah disediakan.

21 **3.5.2 Skenario**

22 **3.5.2.1 Login**

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

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 " <i>Edit</i> " 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 3.5.2.2 Pilih Mahasiswa

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

3 3.5.2.3 Melihat Info Mahasiswa

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

5 3.5.2.4 Edit Mahasiswa

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

7 3.5.2.5 Lihat Histori

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

9 3.5.2.6 Membuat Entri Baru

10 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

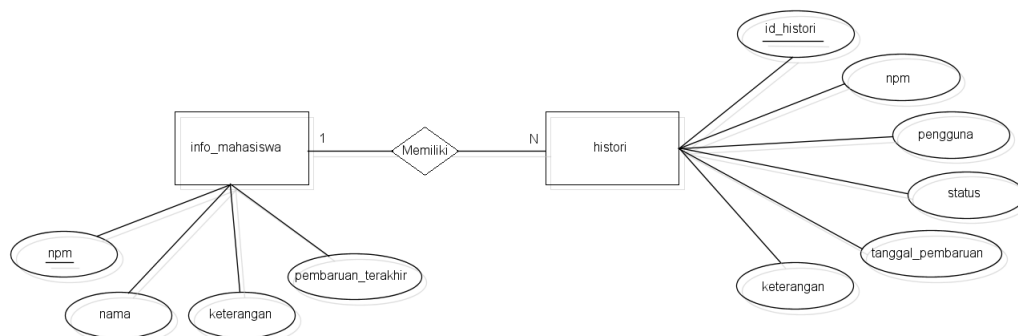
Nama	Lihat Histori		
Aktor	Pengguna		
Deskripsi	Melihat histori perubahan dan aksi melihat yang dilakukan pengguna		
Kondisi Awal	Menampilkan log histori perubahan dan aksi melihat		
Kondisi Akhir	Terus bertambah sesuai aksi yang dilakukan		
Skenario Utama	No	Aksi Aktor	Reaksi Sistem
	1	Pengguna melihat 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 data-base		
Kondisi Awal	Menampilkan form untuk menambah entri baru		
Kondisi Akhir	Input pada form akan dimasukkan kedalam data-base		
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

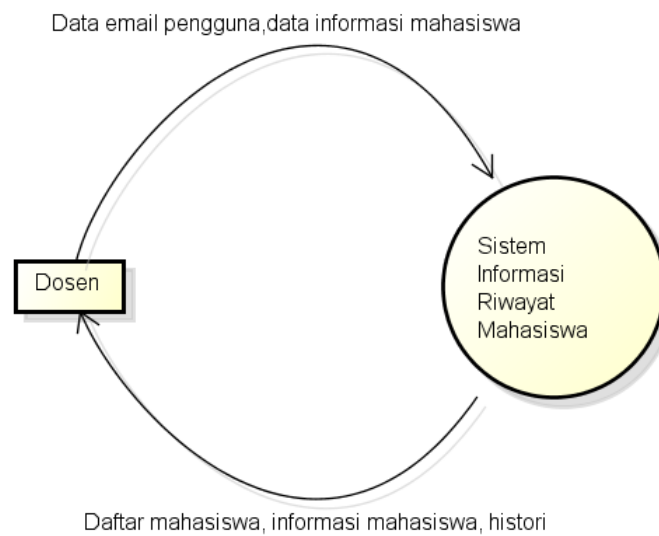
Pada ERD Sistem Informasi Riwayat Mahasiswa terdapat dua entitas yaitu info mahasiswa dan histori. Entitas info mahasiswa dengan relasi memiliki terbungung dengan entitas histori.

3.5.4 Data Flow Diagram

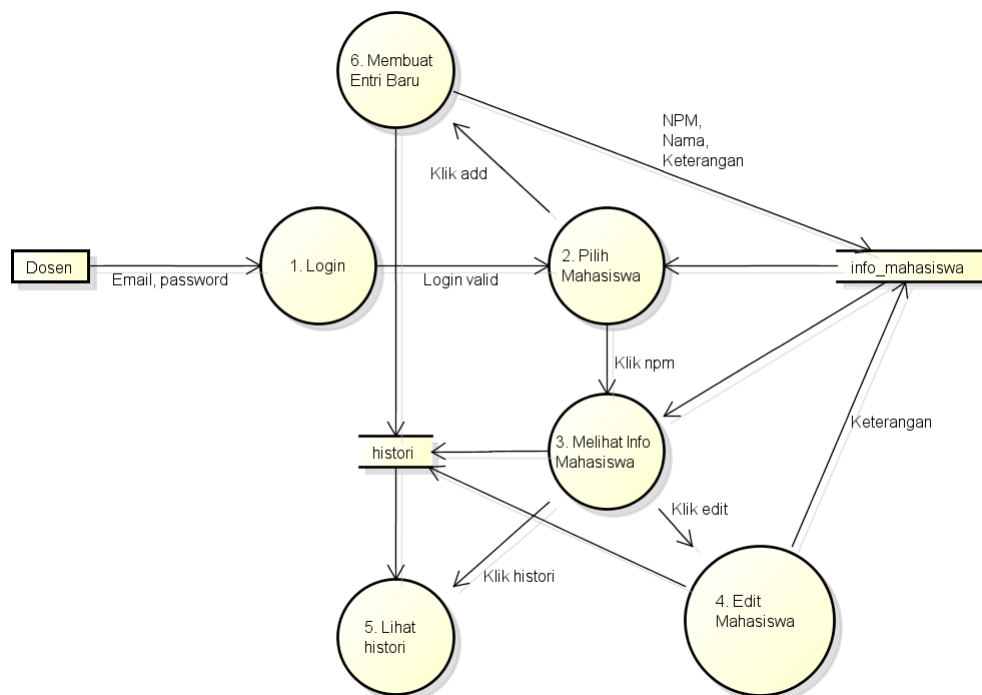
Data Flow Diagram Sistem Informasi Riwayat Mahasiswa digambarkan dalam dua level, yaitu Data Context Diagram / Data Flow Diagram Level 0, Data Flow Diagram Level 1.

(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

- 1 (2) Data *Flow* Diagram Level 1
- 2 Data *Flow* Diagram Level 1 Sistem Informasi Riwayat Mahasiswa dapat dilihat pada
- 3 Gambar 3.21.

Gambar 3.21: Data *Flow* Diagram Sistem Informasi Riwayat Mahasiswa

- 4 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 maha-
13 siswa.
- 14 • No dan Nama Proses : P3 Melihat Info Mahasiswa
15 Deskripsi : Proses melihat info mahasiswa
16 Data Input : -
17 Data Output : Keterangan mahasiswa
18 Proses : pengguna mendapatkan info mahasiswa dari mahasiswa yang telah peng-
19 guna pilih pada proses 2. Proses ini dicatat pada histori sebagai aksi melihat.
- 20 • No dan Nama Proses : P4 Edit Mahasiswa
21 Deskripsi : Proses mengedit info mahasiswa
22 Data Input : Keterangan mahasiswa terbaru
23 Data Output : -
24 Proses : pengguna memperbaharui keterangan yang dimiliki mahasiswa yang telah
25 pengguna pilih pada proses 2. Proses ini dicatat pada histori sebagai aksi mengedit.
- 26 • No dan Nama Proses : P5 Lihat Histori
27 Deskripsi : Proses melihat histori
28 Data Input : -
29 Data Output : Daftar histori
30 Proses : pengguna mendapatkan histori dari mahasiswa yang telah pengguna pilih
31 pada proses 2. Pengguna juga dapat melihat informasi mahasiswa versi sebelum-
32 nya.
- 33 • No dan Nama Proses : P6 Membuat Entri Baru
34 Deskripsi : Proses membuat entri baru
35 Data Input : NPM, Nama, Keterangan
36 Data Output : -
37 Proses : pengguna memasukan data NPM, Nama, Keterangan untuk entri baru
38 lalu menyimpannya. Proses ini dicatat pada histori sebagai aksi membuat entri
39 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.



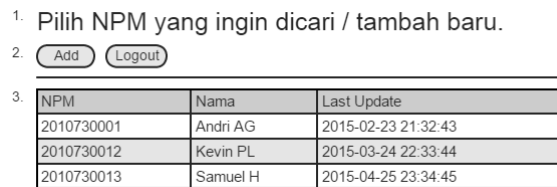
Gambar 4.1: Desain Antarmuka Halaman Awal

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 :

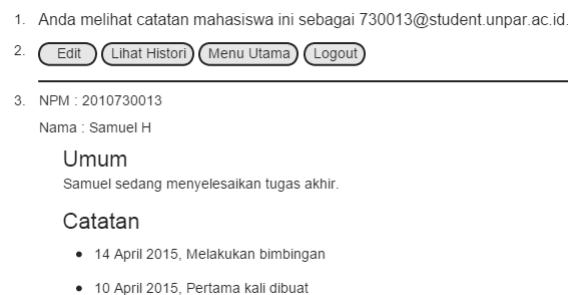
2 (1) Bagian ini merupakan judul dari halaman untuk memilih mahasiswa.

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 :

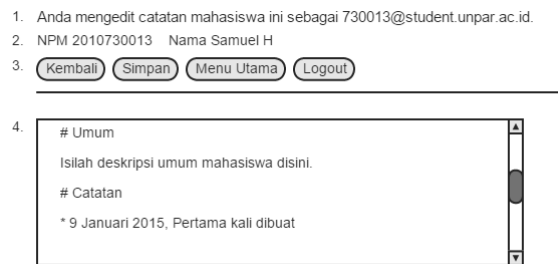
9 (1) Bagian ini merupakan teks yang menampilkan keterangan dan juga pengguna yang
10 sedang menggunakan Sistem Infomasi Riwayat Mahasiswa.

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.



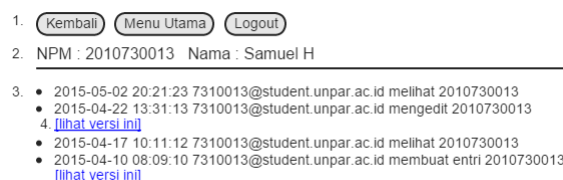
Gambar 4.4: Desain Antarmuka Edit Mahasiswa

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.



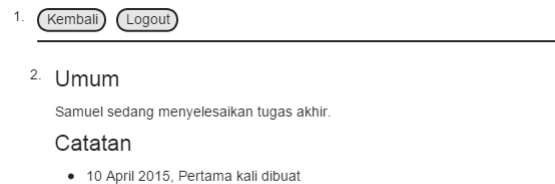
Gambar 4.5: Desain Antarmuka Lihat Histori

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

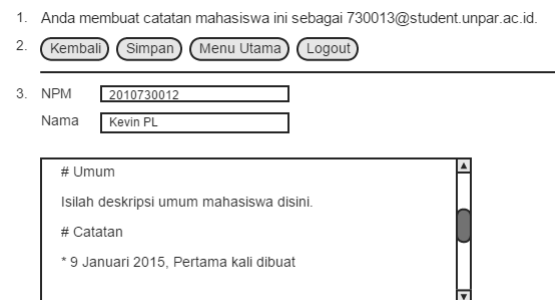
Keterangan :

(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 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.

4.2.1 Modul Login

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

Tabel 4.1: Modul Login

Nama Modul	login.php
Input	<i>username, password</i>
Output	-
Tabel yang diakses	-
Deskripsi	Pengguna memasukkan <i>username</i> dan <i>password</i> kemudian sistem akan melakukan autentikasi menggunakan Google Oauth.

1 4.2.2 Modul Pilih Mahasiswa

- 2 Modul pilih mahasiswa yang dilakukan oleh pengguna (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	InfoMahasiswa
Deskripsi	Pengguna memasukkan npm yang ingin dicari, pengguna juga dapat memilih langsung mahasiswa yang ingin dilihat infonya tanpa melakukan pencarian terlebih dahulu dan pengguna juga dapat membuat entri baru.

3 4.2.3 Modul Info Mahasiswa

- 4 Modul info mahasiswa yang dilakukan oleh pengguna (dosen) dapat dilihat pada Tabel 4.3.

Tabel 4.3: Modul Info Mahasiswa

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

5 4.2.4 Modul Edit Mahasiswa

- 6 Modul *edit* mahasiswa yang dilakukan oleh pengguna (dosen) dapat dilihat pada Tabel 4.4.

7 4.2.5 Modul Lihat Histori

- 8 Modul lihat histori yang dilakukan oleh pengguna (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	InfoMahasiswa dan Histori
Deskripsi	Pengguna memasukkan atau merubah keterangan mahasiswa pada teks area yang telah disediakan menggunakan teks dengan sintaks Markdown lalu pengguna menyimpan untuk menaruh perubahan yang dilakukan. Pengguna 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	Histori
Deskripsi	Pengguna mendapatkan laporan berupa daftar histori yang dimiliki setiap mahasiswa.

1 4.2.6 Modul Entri Baru

- 2 Modul entri baru yang dilakukan oleh pengguna (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	-
Deskripsi	Pengguna memasukkan npm, nama, dan keterangan mahasiswa pada teks area yang telah disediakan menggunakan teks dengan sintaks Markdown lalu pengguna menyimpan untuk membuat entri baru tersebut. Pengguna 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	-

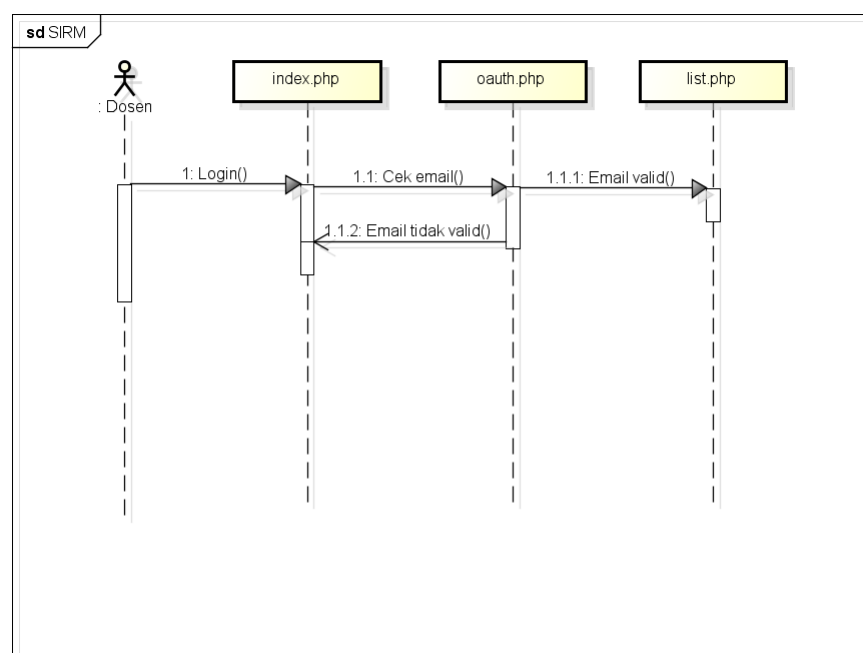
Tabel 4.8: Rancangan Tabel Histori

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	-

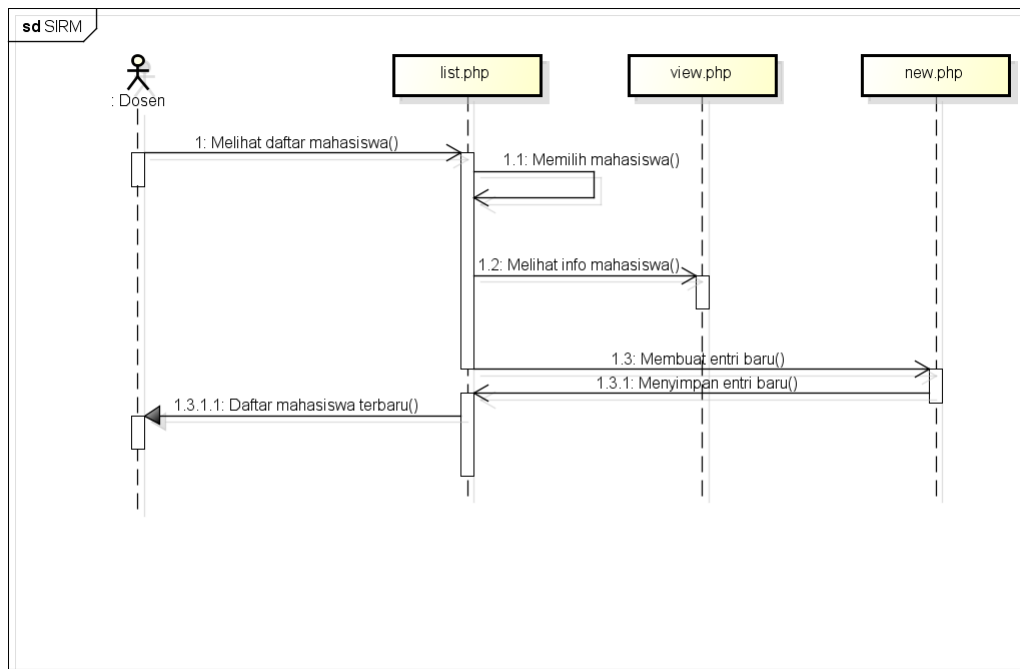
1 4.4 Diagram Sekuens

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

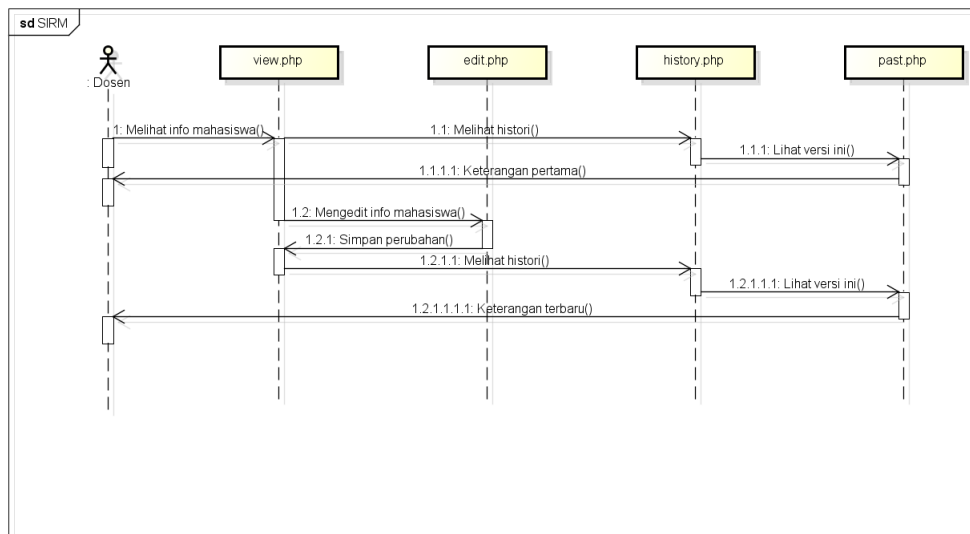
- 4 (1) Sekuens bagian satu mencakup proses login dapat dilihat pada Gambar 4.8.
- 5 (2) Sekuens bagian dua mencakup proses memilih mahasiswa, melihat info mahasiswa, dan
6 membuat entri baru. Dapat dilihat pada Gambar 4.9.
- 7 (3) Sekuens bagian tiga mencakup melihat info mahasiswa, mengedit info mahasiswa, me-
8 lihat 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

1

BAB 5

2

IMPLEMENTASI DAN PENGUJIAN PERANGKAT

3

LUNAK

4 Bab ini terdiri atas tiga bagian, yaitu Implementasi Perangkat Lunak, Implementasi Basis
5 Data dan Pengujian Perangkat Lunak. Bagian implementasi berisi penjelasan lingkungan
6 pengembangan perangkat lunak. Sedangkan bagian pengujian berisi hasil pengujian terha-
7 dap perangkat lunak yang telah dibangun.

8 5.1 Implementasi Perangkat Lunak

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

12 5.1.1 Lingkungan Implementasi Perangkat Keras

13 Dalam membangun perangkat lunak ini digunakan spesifikasi perangkat keras sebagai beri-
14 kut:

15 (a) Processor: AMD A10-5750M 2.5GHz

16 (b) RAM: 4 GB DDR3

17 (c) Harddisk: 1TB

18 (d) VGA: AMD Radeon HD 8650G 2GB

19 (e) Koneksi Internet: WAN

20 5.1.2 Lingkungan Implementasi Perangkat Lunak

21 Dalam membangun perangkat lunak ini digunakan spesifikasi perangkat lunak sebagai ber-
22 ikut:

23 (a) Sistem Operasi: Windows 8.1 Pro 64-bit

24 (b) Bahasa Pemrograman: PHP Version 5.6.3

25 (c) Aplikasi: XAMPP v5.6.3

26 (d) DBMS: MySQL

- 1 (e) Aplikasi web browser: Google Chrome
- 2 (f) Library: Google APIs Client Library untuk PHP
- 3 (g) Javascript: Strapdown.js
- 4 (h) Framework: Foundation 5

5 5.1.3 Hasil Implementasi Perangkat Lunak

6 Kode program perangkat lunak ditulis berdasarkan perancangan yang telah dibahas pada
 7 Bab 4. Hasil implementasi perangkat lunak menghasilkan kode program berbasis PHP. Kode
 8 program yang telah diimplementasi dapat dilihat pada Lahiran A.

9 5.2 Implementasi Basis Data

10 Implementasi basis data dalam sistem informasi riwayat mahasiswa dengan membuat basis
 11 data dengan nama sirm. Untuk kode dapat dilihat di bawah ini.

12 `| CREATE DATABASE 'sirm' ;`

13 Basis data sistem informasi riwayat mahasiswa menggunakan dua tabel basis data.
 14 Tabel-tabel tersebut terdiri dari :

- 15 • Tabel InfoMahasiswa, digunakan untuk menyimpan semua data mahasiswa. Untuk
 16 kode dapat dilihat di bawah ini.

```
17 CREATE TABLE 'info_mahasiswa' (
18     'npm' varchar(10) NOT NULL,
19     'nama' varchar(60) NOT NULL,
20     'keterangan' text NOT NULL,
21     'pembaruan_terakhir' datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
22     PRIMARY KEY ('npm'),
23 ) ENGINE=InnoDB DEFAULT CHARSET=latin1 ;
```

- 24 • Tabel Histori, digunakan untuk menyimpan semua data histori. Untuk kode dapat
 25 dilihat di bawah ini.

```
26 CREATE TABLE 'histori' (
27     'id_histori' int(5) NOT NULL AUTO_INCREMENT,
28     'npm' varchar(10) NOT NULL,
29     'pengguna' varchar(60) NOT NULL,
30     'status' text NOT NULL,
31     'tanggal_pembaruan' datetime NOT NULL,
32     'keterangan' text NOT NULL,
33     PRIMARY KEY ('id_histori'),
34 ) ENGINE=InnoDB DEFAULT CHARSET=latin1 ;
```

35 5.3 Pengujian Perangkat Lunak

36 Pada bagian ini akan dibahas mengenai pengujian yang akan dilakukan terhadap perangkat
 37 lunak. Pengujian tersebut terdiri dari dua bagian yaitu pengujian fungsional dan penguji-
 38 an eksperimental. Pengujian fungsional bertujuan untuk memastikan bahwa seluruh fungsi
 39 yang dibangun pada perangkat lunak berjalan sesuai dengan yang direncanakan. Sedangkan
 40 pengujian eksperimental bertujuan untuk mengujikan perangkat lunak langsung ke peng-
 41 guna. Pada bagian pengujian terdapat perubahan program pada bagian oauth.php, jadi
 42 dapat menjalankan pengujian dengan email yang diakhiri @student.unpar.ac.id dikarenakan
 43 penulis tidak memiliki email yang diakhiri @unpar.ac.id. Kode program untuk pengujian
 44 perangkat lunak dapat dilihat pada Lampiran B.

5.3.1 Lingkungan Pengujian Perangkat Keras

Dalam pengujian 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

5.3.2 Lingkungan Pengujian Perangkat Lunak

Dalam pengujian 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) DBMS: MySQL
- (e) Aplikasi web browser: Google Chrome
- (f) Library: Google APIs Client Library untuk PHP
- (g) Javascript: Strapdown.js
- (h) Framework: Foundation 5

5.3.3 Pengujian Fungsional

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

(1) Fungsi login

Pengujian fungsi ini dilakukan untuk memastikan perangkat lunak terhubung ke server Google untuk melakukan otentikasi dan otorisasi serta memeriksa apakah email yang digunakan untuk login diakhiri "@unpar.ac.id" atau "@student.unpar.ac.id" dikarenakan penulis tidak mempunyai akun dosen. Contoh kasus adalah melakukan login sebanyak dua kali, yang pertama menggunakan email yang diakhiri "@unpar.ac.id" atau "@student.unpar.ac.id", dan yang kedua menggunakan email yang diakhiri selain "@unpar.ac.id" dan "@student.unpar.ac.id". Pengujian pertama pengguna membuka halaman index.php dapat dilihat pada Gambar 5.1. Lalu pengguna melakukan login menggunakan email "7310013@student.unpar.ac.id" dapat dilihat pada Gambar 5.2. Lalu akan ada konfirmasi bahwa akun yang digunakan dikelola oleh student.unpar.ac.id dapat dilihat pada Gambar 5.3. Lalu pengguna akan diarahkan ke CAS (Central Authentication Service) UNPAR dan melakukan login kembali dapat dilihat pada Gambar

5.4. Lalu pengguna akan diminta untuk memberikan izin akses dari pihak pengguna dapat dilihat pada Gambar 5.5. Setelah pengguna memberikan izin akses maka akan dilakukan dengan fungsi memilih mahasiswa yang akan dibahas pada poin berikutnya. Sedangkan pengujian kedua pengguna melakukan login menggunakan email "bletack@gmail.com" dapat dilihat pada Gambar 5.6. Lalu pengguna akan mendapat *alert* karena email yang digunakan tidak sesuai dengan ketentuan dapat dilihat pada Gambar 5.7. Setelah pengguna menekan tombol ok pada *alert* maka pengguna akan dikembalikan ke halaman `index.php`. Hal ini menunjukkan fungsi login sudah berjalan dengan baik.

(2) Fungsi memilih mahasiswa

Pengujian fungsi ini dilakukan untuk memastikan pengguna dapat memilih mahasiswa. Pada halaman `list.php` terdapat tabel yang berisikan npm, nama, dan last update dan pengguna dapat memilih mahasiswa dengan menekan npm yang diinginkan. Contoh pengujian pengguna akan memilih mahasiswa dengan npm 2010730013 maka akan menghasilkan link yang mengarah ke `view.php?npm = 2010730013` dapat dilihat pada Gambar 5.8. Hal ini menunjukkan fungsi memilih mahasiswa sudah berjalan dengan baik.

(3) Fungsi melihat info mahasiswa

Pengujian fungsi ini dilakukan untuk memastikan pengguna dapat melihat informasi mahasiswa dari mahasiswa yang telah dipilih oleh pengguna. Contoh pengujian fungsi ini merupakan lanjutan dari fungsi memilih mahasiswa, dimana setelah pengguna memilih mahasiswa pada `list.php` maka sistem akan menampilkan informasi dari mahasiswa tersebut dapat dilihat pada Gambar 5.9. Hal ini menunjukkan fungsi melihat info mahasiswa sudah berjalan dengan baik.

(4) Fungsi mengedit info mahasiswa

Pengujian fungsi ini dilakukan untuk memastikan informasi mahasiswa dapat diedit. Contoh pengujian mengambil informasi dari mahasiswa yang telah dilihat informasinya pada fungsi melihat info mahasiswa. Dimana keterangan sebagai salah satu informasi mahasiswa yang ada akan ditampilkan dan pengguna dapat melakukan perubahan lalu menyimpan perubahan dengan menekan tombol "Simpan" dapat dilihat pada Gambar 5.10. Setelah menyimpan perubahan pengguna akan dibawa kembali ke halaman `list.php`. Hal ini menunjukkan fungsi mengedit info mahasiswa sudah berjalan dengan baik.

(5) Fungsi melihat histori

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

(6) Fungsi membuat entri baru

Pengujian fungsi ini dilakukan untuk memastikan pada saat membuat entri baru terda-

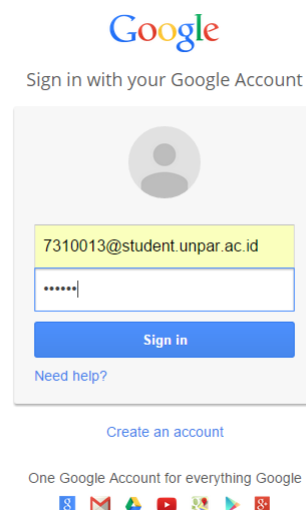
1 pat template markdown dan berhasil menyimpan entri baru tersebut. Contoh pengujian
2 menambahkan entri baru untuk mahasiswa yang memiliki npm 2010730014, nama Na-
3 dia, dan keterangan sesuai template. Terdapat template markdown pada saat membuka
4 halaman new.php dapat dilihat pada Gambar 5.14. Mengisi data npm 2010730014 dan
5 nama Nadia dapat dilihat pada Gambar 5.15. Setelah pengguna menekan tombol "Simp-
6 an" maka data yang telah dimasukan akan tersimpan dan pengguna akan dikembalikan
7 ke halaman list.php. Pengguna dapat melihat entri baru dengan npm 2010730014 dan
8 nama Nadia telah masuk kedalam tabel dapat dilihat pada Gambar 5.16. Hal ini me-
9 nunjukkan fungsi membuat entri baru sudah berjalan dengan baik.

WELCOME to SIRM

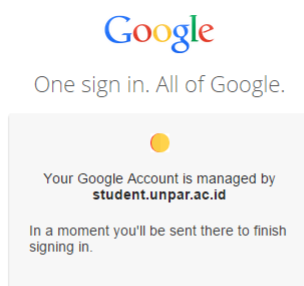
(Sistem Informasi Riwayat Mahasiswa)

[Login with Google](#)

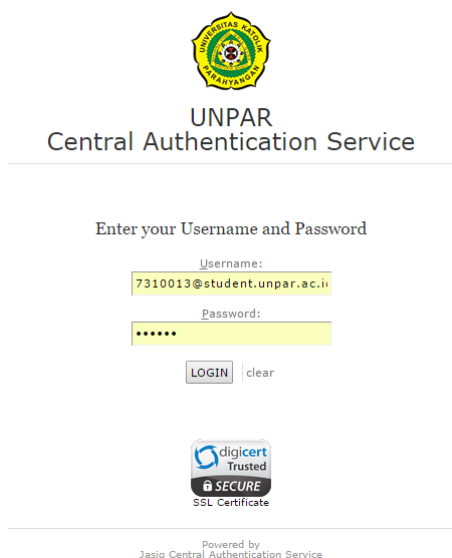
Gambar 5.1: Membuka Halaman index.php



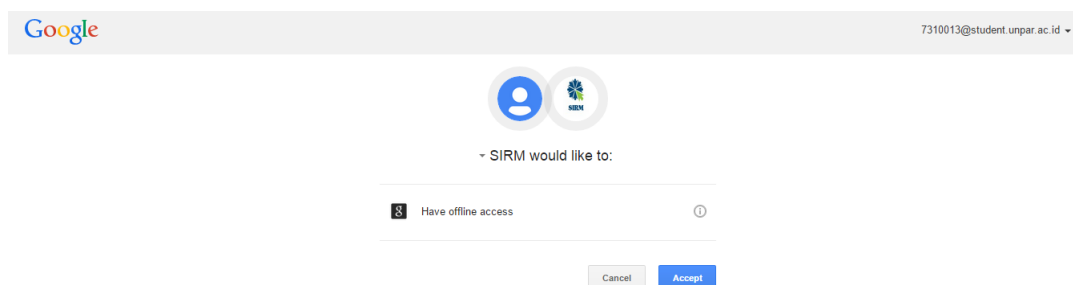
Gambar 5.2: Login Dengan Email yang Diakhiri "@student.unpar.ac.id"



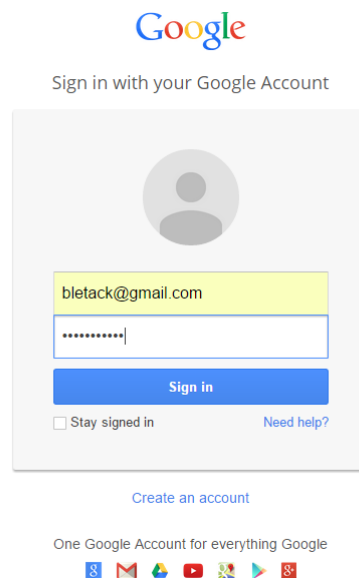
Gambar 5.3: Konfirmasi Email yang Dikelola oleh student.unpar.ac.id



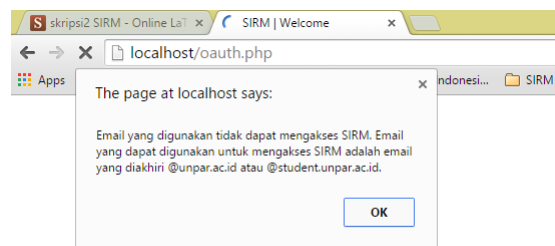
Gambar 5.4: CAS UNPAR



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

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

NPM 2010730013 Nama Samuel Herman

- 2015-05-08 00:08:54 7310013@student.unpar.ac.id melihat 2010730013
- 2015-05-08 00:06:57 7310013@student.unpar.ac.id melihat 2010730013
- 2015-05-08 00:06:51 7310013@student.unpar.ac.id melihat 2010730013
- 2015-05-08 00:06:48 7310013@student.unpar.ac.id melihat 2010730013
- 2015-05-08 00:06:34 7310013@student.unpar.ac.id melihat 2010730013
- 2015-05-08 00:06:28 7310013@student.unpar.ac.id melihat 2010730013
- 2015-05-08 00:03:56 7310013@student.unpar.ac.id melihat 2010730013
- 2015-05-05 09:20:01 pascal@unpar.ac.id melihat 2010730013
- 2015-05-04 15:40:24 7310013@student.unpar.ac.id melihat 2010730013
- 2015-04-30 14:31:24 1210031@student.unpar.ac.id melihat 2010730013
- 2015-04-29 15:24:57 7310071@student.unpar.ac.id melihat 2010730013
- 2015-04-24 00:31:06 7310013@student.unpar.ac.id melihat 2010730013
- 2015-04-24 00:23:34 7310013@student.unpar.ac.id mengedit 2010730013 [\[lihat versi ini\]](#)

Gambar 5.11: Melihat Histori

Kembali Logout

Umum

Samuel sedang mengambil tugas akhir

Catatan

- 21 April 2015, pertama kali dibuat

Gambar 5.12: Keterangan Versi Pertama

Kembali 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 5.13: 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
2010730014

Nama
Nadia

```
# 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

5.3.4 Pengujian Eksperimental

Pengujian eksperimental dilakukan kepada dua kelompok mahasiswa. Kelompok pertama terdiri dari empat mahasiswa jurusan teknik informatika dan kelompok kedua terdiri dari empat mahasiswa diluar jurusan 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 kuesioner. Penulis akan mengukur waktu selama pengujian dan mencatat berapa waktu yang dibutuhkan penguji untuk memahami dan menjalankan setiap fungsi dengan benar.

5.3.4.1 Pengujian Eksperimental Kelompok Pertama

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 5.1. 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

$$10 + 5 + 14 + 36/4 = 64/4 = 16.25\text{detik}$$

(6) Membuat Entri Baru

$$186 + 42 + 103 + 86/4 = 417/4 = 104.25\text{detik}$$

Tabel 5.1: 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 Mahasiswa	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 5.3.4.2 Pengujian Eksperimental Kelompok Kedua

2 Kelompok kedua terdiri dari empat orang mahasiswa diluar jurusan teknik informatika.
 3 Keempat penguji berhasil menjalankan dan memahami SIRM dengan baik. Untuk catatan
 4 waktu yang dimiliki oleh kelompok kedua dapat dilihat pada Tabel 5.2. Rata-rata waktu
 5 yang diperlukan kelompok kedua untuk menjalankan setiap fungsi dapat dilihat di bawah
 6 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

$$72 + 66 + 94 + 67/4 = 299/4 = 74.75detik$$

(6) Membuat Entri Baru

$$68 + 138 + 124 + 205/4 = 535/4 = 133.75detik$$

Tabel 5.2: 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 Mahasiswa	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

5.3.4.3 Perbandingan Rata-Rata Waktu Pengujian

- Perbandingan dari rata-rata waktu pengujian eksperimental antara kelompok pertama dan kelompok kedua dapat dilihat pada Tabel 5.3 dan Gambar 5.17.

Tabel 5.3: Perbandingan Rata-rata Waktu Pengujian Eksperimental

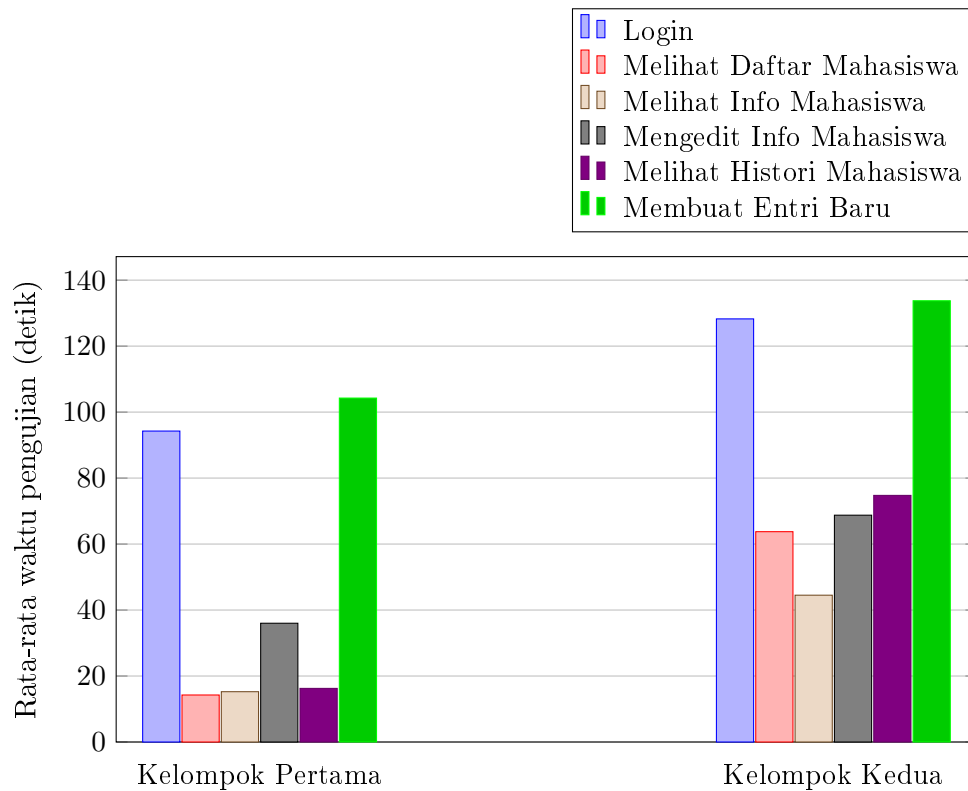
Aksi	Kelompok Pertama	Kelompok Kedua	Keterangan
Login	94.25 detik	128.25 detik	Kelompok pertama lebih cepat 34 detik dari kelompok kedua
Melihat Daftar Mahasiswa	14.25 detik	63.75 detik	Kelompok pertama lebih cepat 49.5 detik dari kelompok kedua
Melihat Info Mahasiswa	15.25 detik	44.5 detik	Kelompok 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

Waktu yang diperlukan oleh kelompok pertama lebih sebentar untuk mengerti dan menjalankan semua fungsi dengan benar dibandingkan dengan kelompok kedua. Dikarenakan kelompok pertama adalah kelompok mahasiswa yang mewakili jurusan teknik informatika yang lebih sering berinteraksi dengan komputer dibandingkan kelompok kedua yang merupakan kelompok mahasiswa diluar jurusan teknik informatika. Namun perbedaan waktu untuk mengerti dan menjalankan semua fungsi dengan benar antara kedua kelompok masih dalam hitungan detik dan tidak ada yang melebihi satu menit. Hal tersebut menunjukkan perangkat lunak SIRM mudah dimengerti bagi orang yang lebih sering berinteraksi dengan komputer maupun bagi orang yang jarang berinteraksi dengan komputer.

5.3.4.4 Hasil Kuesioner

Kuesioner yang digunakan pada pengujian eksperimental dapat dilihat pada Lampiran C. Sebelum masuk ke penilai dari data kuesioner, akan ditetapkan terlebih dahulu untuk sistem penilaian. Sistem penilaian yang akan digunakan untuk penilaian data kuesioner dapat dilihat pada Tabel 5.4.

Nilai kuesioner dari kelompok pertama dapat dilihat pada Tabel 5.5. Nilai kuesioner dari kelompok kedua dapat dilihat pada Tabel 5.6. Perbandingan nilai kuesioner antara



Gambar 5.17: Diagram Perbandingan Rata-Rata Waktu Pengujian

- 1 kelompok pertama dan kelompok kedua dapat dilihat pada Gambar 5.18. Perbandingan
- 2 rata-rata nilai kuesioner antara kelompok pertama dan kelompok kedua dapat dilihat pada
- 3 Tabel 5.7.

Tabel 5.4: Tabel Sistem Penilaian Kuesioner

Jawaban	Nilai	Keterangan
SS/SB	3	Sangat Setuju/Sangat Baik
S/B	2	Setuju/Baik
TS/TB	1	Tidak Setuju/Tidak Baik
STS/STB	0	Sangat Tidak Setuju/Sangat Tidak Baik

4 Total dari nilai kuesioner yang dimiliki kedua kelompok berdasarkan Gambar 5.18 tidak
 5 memiliki perbedaan yang signifikan. Hal tersebut membuktikan kedua kelompok sangat
 6 setuju dengan kinerja, akurasi dan kemudahan dalam menggunakan SIRM.

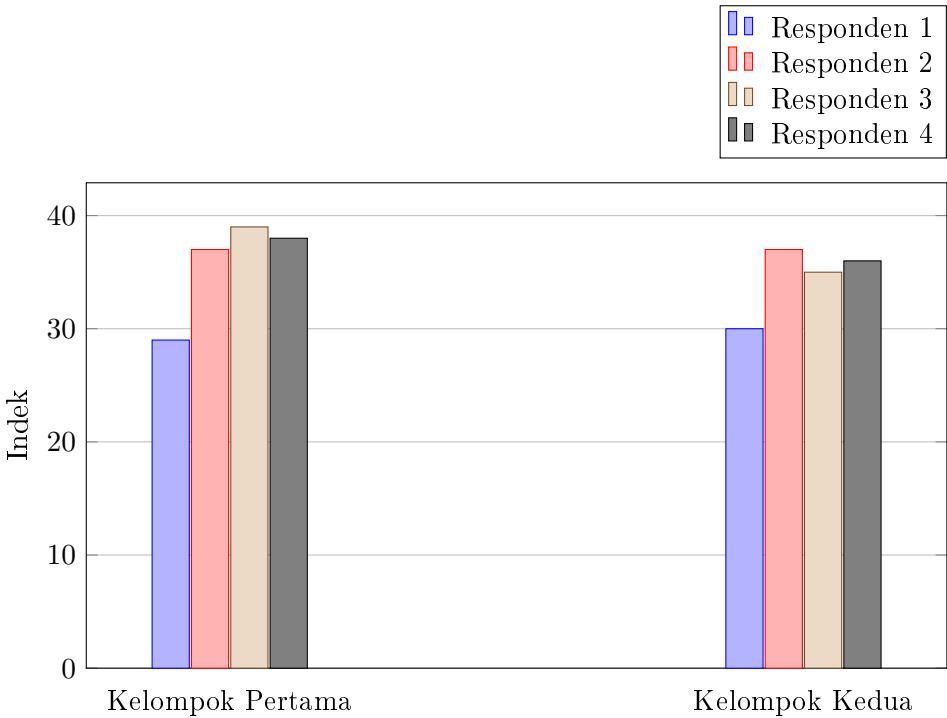
7 Berdasarkan Tabel 5.7 data rata-rata nilai kuesioner dari kedua kelompok, untuk rata-
 8 rata tertinggi terdapat pada pertanyaan kuesioner nomor 7 dan untuk rata-rata terendah
 9 terdapat pada pertanyaan kuesioner nomor 13. Rata-rata tertinggi pada pertanyaan kuesio-
 10 ner nomor 7 menyatakan kedua kelompok setuju untuk akurasi yang dimiliki SIRM terlebih
 11 untuk fungsi login. Rata-rata terendah pada pertanyaan kuesioner nomor 13 menyatak-
 12 an kedua kelompok kurang setuju untuk kemudahan yang dimiliki SIRM untuk membantu
 13 pemahaman pengguna dalam mengelola riwayat mahasiswa.

Tabel 5.5: Tabel Nilai Kuesioner Kelompok Pertama

Responden	No Pertanyaan															Total
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1	3	2	2	2	1	2	2	1	2	2	2	2	1	2	3	29
2	2	3	2	2	2	2	3	3	3	2	3	3	3	2	2	37
3	2	2	3	3	3	3	3	3	3	3	3	2	2	2	2	39
4	2	3	3	3	2	3	3	3	2	3	3	2	1	2	3	38
Total	9	10	10	10	8	10	11	10	10	10	11	9	7	8	10	143

Tabel 5.6: Tabel Nilai Kuesioner Kelompok Kedua

Responden	No Pertanyaan															Total
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1	2	2	2	2	3	2	2	2	2	2	2	1	2	2	2	30
2	2	3	2	3	3	2	3	2	2	3	3	2	2	3	2	37
3	3	3	2	2	2	3	3	3	2	2	2	3	1	2	2	35
4	3	2	2	2	2	2	3	3	3	3	3	2	2	2	2	36
Total	10	10	8	9	10	9	11	10	9	10	10	8	7	9	8	138



Gambar 5.18: Diagram Perbandingan Total Nilai Kuesioner Setiap Responden

Tabel 5.7: Tabel Rata-Rata Nilai Kuesioner Kedua Kelompok

No	Nilai Rata-Rata		Keterangan
	Kelompok Pertama	Kelompok Kedua	
1	2.25	2.5	-
2	2.5	2.5	-
3	2.5	2	-
4	2.5	2.25	-
5	2	2.5	-
6	2.5	2.25	-
7	2.75	2.75	Rata-rata tertinggi
8	2.5	2.5	-
9	2.5	2.25	-
10	2.5	2.5	-
11	2.75	2.5	-
12	2.25	2	-
13	1.75	1.75	Rata-rata terendah
14	2	2.25	-
15	2.5	2	-
Total	35.75	34.5	

BAB 6

KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang didapat dari proses perancangan dan pengujian perangkat lunak yang dibangun, juga saran-saran untuk penelitian ini jika ingin dikembangkan di kemudian hari.

6.1 Kesimpulan

Kesimpulan yang didapat dari pembangunan perangkat lunak Sistem Informasi Riwayat Mahasiswa (SIRM) antara lain :

- (1) Penggunaan Google Authentication berhasil mengautentikasi pengguna dan berfungsi dengan baik pada perangkat lunak Sistem Informasi Riwayat Mahasiswa (SIRM).
- (2) Penggunaan Markdown berhasil membuat format penulisan seragam dan berfungsi dengan baik pada perangkat lunak Sistem Informasi Riwayat Mahasiswa (SIRM).
- (3) Penggunaan StrapdownJS berhasil menampilkan teks dengan sintaks Markdown ke halaman website dan berfungsi dengan baik pada perangkat lunak Sistem Informasi Riwayat Mahasiswa (SIRM).
- (4) Penggunaan Zurb Foundation berhasil membuat tampilan antarmuka perangkat lunak Sistem Informasi Riwayat Mahasiswa (SIRM) menjadi responsif, tampilan antarmuka mengikuti lebar mesin pencari dan/atau layar komputer tanpa melakukan permintaan tambahan ke server.
- (5) Pada tahap pengujian dapat disimpulkan bahwa perangkat lunak Sistem Informasi Riwayat Mahasiswa (SIRM) sudah berjalan dengan baik dan memberikan *output* sesuai yang diharapkan pengguna.

6.2 Saran

Saran yang dapat diberikan untuk perbaikan dan pengembangan Sistem Informasi Riwayat Mahasiswa (SIRM) antara lain :

- (1) Agar setiap pengguna dapat saling berinteraksi untuk pengembangan dapat ditambahkan fitur-fitur seperti kirim pesan, *chat-room*, dan forum.
- (2) Pada fungsi histori untuk pengembangan dapat ditambahkan keterangan yang membandingkan versi baru dan versi lama. Jadi pengguna mengetahui bagian mana yang dihapus dan bagian mana yang ditambah atau dirubah.

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))
38                 {
39                     $result = 'invalid_email';
40                 }
41                 return $result;
42             }
43             $status = is_valid_email($email);
44
45             if($status == "valid_email")
46             {
47                 header("Location:./list.php");
48                 exit;
49             }
50             else
51             {
52                 echo "<script>alert('Email yang digunakan tidak dapat mengakses SIRM. Email yang
53                     dapat digunakan untuk mengakses SIRM adalah email yang diakhiri @unpar.ac.id
54                     atau @student.unpar.ac.id. ');window.location.href='index.php?logout';</script>";
55                 exit;
56             }
57         }
58     }
59     <?= is_valid_email($email); ?>
60 </body>
61 </html>

```

Listing A.3: list.php

```

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

```



```

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

```

Listing A.4: view.php

```

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

```

```

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

```

Listing A.5: edit.php

```

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

```

Listing A.6: history.php

```

95 | <!doctype html>

```

```

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

```

Listing A.7: past.php

```

74 <!doctype html>
75 <html class="no-js" lang="en">
76 <head>
77 <meta charset="utf-8" />
78 <meta name="viewport" content="width=device-width, _initial-scale=1.0" />
79 <title>SIRM | Past</title>
80 <link rel="stylesheet" href="css/foundation.css" />
81 <script src="js/vendor/modernizr.js"></script>
82 </head>
83 <body>
84 <?php
85 session_start();
86 $id = $_GET["id"];
87 ?>
88 <div class="row">
89 <div class="small-11-small-centered-columns">
90 <ul class="button-group">
91 <li><a href="javascript:history.back(1)" class="button_secondary">Kembali</a></li>
92 <li><a href="index.php?logout" class="button_secondary">Logout</a></li>
93 </ul>
94 <hr/>
95 <?php
96
97 include_once "configDatabase.php";
98

```

```

1         if (! $id_mysql)
2         {
3             die ("Database_tidak_bisa_dibuka");
4         }
5
6         if (! mysql_select_db("sirm", $id_mysql))
7         {
8             die ("Database_tidak_bisa_dipilih");
9         }
10
11         $cari = mysql_query ("SELECT_keterangan_FROM_histori_WHERE_id_histori='$id '",
12                               $id_mysql);
13         while($row = mysql_fetch_array($cari))
14         {
15             ?>
16         </div>
17     </div>
18 <xmp style="display:none;">
19 <?php
20 echo $row['keterangan'];
21 ?>
22 </xmp>
23
24     <?php
25     }
26     ?>
27     <script src="js/0.2/strapdown.js"></script>
28 </body>
</html>

```

Listing A.8: new.php

```

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

```

```

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

```

Listing A.9: client.php

```

48 <?php
49 $client = new Google_Client();
50 $client->setClientId('568951368854-ufmbistn0pcaq0khubafo1a133orfgve.apps.googleusercontent.com');
51
52 $client->setClientSecret('-cSZ-AUmeQ9PaWWry_IpiBBi');
53 $client->setRedirectUri('http://localhost/oauth.php');
54 $client->setDeveloperKey('AlzaSyDRoDJAzUR_TsNUNRUeTYsBb7dFBQKZy7M');
55 ?>

```

Listing A.10: configDatabase.php

```

56 <?php
57 $pemakai="admin";
58 $pass="admin";
59 $id_mysql=mysql_connect("localhost", $pemakai, $pass);
60 ?>

```


LAMPIRAN B

KODE PROGRAM UNTUK PENGUJIAN

Listing B.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 B.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>
74   </head>
```

```

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

```

Listing B.3: list.php

```

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

```



```

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

```

Listing B.4: view.php

```

29 <!doctype html>
30 <html class="no-js" lang="en">
31     <head>
32         <meta charset="utf-8" />
33         <meta name="viewport" content="width=device-width, initial-scale=1.0" />
34         <title>SIRM | View</title>
35         <link rel="stylesheet" href="css/foundation.css" />
36         <script src="js/vendor/modernizr.js"></script>
37     </head>
38     <body>
39         <?php
40             session_start();
41             $npm = $_GET['npm'];
42
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']?>.</h3>
47                 <ul class="button-group">
48                     <li><a href="edit.php?npm=<?php echo $npm?>" class="button_secondary">Edit</a>
49                     </li>
50                     <li><a href="history.php?npm=<?php echo $npm?>" class="button_secondary">Lihat
51                     Histori</a></li>
52                     <li><a href="list.php" class="button_secondary">Menu Utama</a></li>
53                     <li><a href="index.php?logout" class="button_secondary">Logout</a></li>
54                 </ul>
55             </div>
56         <?php
57             include_once "configDatabase.php";
58
59             if (! $id_mysql)
60             {
61                 die ("Database_tidak_bisa_dibuka");
62             }
63
64             if (! mysql_select_db("sirm", $id_mysql))
65             {
66                 die ("Database_tidak_bisa_dipilih");
67             }
68
69             $lihat = "INSERT INTO histori (npm, pengguna, status, tanggal_pembaruan, keterangan)
70             VALUES ('" . mysql_real_escape_string($npm) . "', '" . $_SESSION['email'] . "', '" .
71             melihat, now(), '')";
72
73             if (mysql_query($lihat) == TRUE)
74             {
75
76             }
77             else
78             {
79                 echo "Error: " . $lihat . "<br>" . $id_mysql->error;
80             }
81
82             $cari = mysql_query("SELECT * FROM info_mahasiswa WHERE npm='$npm'", $id_mysql);
83
84             while($row = mysql_fetch_array($cari))
85             {
86                 echo "NPM: " . $row['npm']; echo "<br>";
87                 echo "Nama: " . $row['nama']; echo "<br>";
88             }
89         </div>
90     </div>
91     <xmp style="display:none;">
92     <?php
93     echo $row['keterangan'];
94     ?>
95 </xmp>
96
97     <?php
98     }
99     ?>

```

```

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

```

Listing B.5: edit.php

```

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

```

Listing B.6: history.php

```

95 | <!doctype html>

```

```

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

```

Listing B.7: past.php

```

74 <!doctype html>
75 <html class="no-js" lang="en">
76 <head>
77 <meta charset="utf-8" />
78 <meta name="viewport" content="width=device-width, _initial-scale=1.0" />
79 <title>SIRM | Past</title>
80 <link rel="stylesheet" href="css/foundation.css" />
81 <script src="js/vendor/modernizr.js"></script>
82 </head>
83 <body>
84 <?php
85 session_start();
86 $id = $_GET["id"];
87 ?>
88 <div class="row">
89 <div class="small-11-small-centered-columns">
90 <ul class="button-group">
91 <li><a href="javascript:history.back(1)" class="button_secondary">Kembali</a></li>
92 <li><a href="index.php?logout" class="button_secondary">Logout</a></li>
93 </ul>
94 <hr/>
95 <?php
96
97 include_once "configDatabase.php";
98

```

```

1         if (! $id_mysql)
2         {
3             die ("Database_tidak_bisa_dibuka");
4         }
5
6         if (! mysql_select_db ("sirm", $id_mysql))
7         {
8             die ("Database_tidak_bisa_dipilih");
9         }
10
11         $cari = mysql_query ("SELECT_keterangan_FROM_histori_WHERE_id_histori=' $id '",
12                             $id_mysql);
13         while ($row = mysql_fetch_array ($cari))
14         {
15             ?>
16             </div>
17         </div>
18 <xmp style="display:none;">
19 <?php
20 echo $row['keterangan'];
21 ?>
22 </xmp>
23
24         <?php
25         }
26         ?>
27         <script src="js/0.2/strapdown.js"></script>
28 </body>
</html>

```

Listing B.8: new.php

```

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

```

```

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

```

Listing B.9: client.php

```

48 <?php
49 $client = new Google_Client();
50 $client->setClientId('568951368854-ufmbistn0pcaq0khubafo1a133orfgve.apps.googleusercontent.com');
51
52 $client->setClientSecret('-cSZ-AUmeQ9PaWWry_IpiBBi');
53 $client->setRedirectUri('http://localhost/oauth.php');
54 $client->setDeveloperKey('AlzaSyDRoDJAzUR_TsNUNRUeTYsBb7dFBQKZy7M');
55 ?>

```

Listing B.10: configDatabase.php

```

56 <?php
57 $pemakai="admin";
58 $pass="admin";
59 $id_mysql=mysql_connect("localhost", $pemakai, $pass);
60 ?>

```


LAMPIRAN C

KUESIONER

Identitas Responden

1 No. Responden : (diisi oleh peneliti)

2 Jenis Kelamin : Pria/Wanita

3 Usia : tahun

4 Jurusan :

5 Angkatan :

Berdasarkan atas pengalaman Sodara menjalankan program SIRM, berilah tanda centang pada bobot nilai alternatif jawaban yang paling merefleksikan persepsi Sodara pada setiap pertanyaan. (hanya diperbolehkan satu jawaban yang dianggap paling sesuai)

Keterangan

- SS/SB = Sangat Setuju/Sangat Baik
- S/B = Setuju/Baik
- TS/TB = Tidak Setuju/Tidak Baik
- STS/STB = Sangat Tidak Setuju/Sangat Tidak Baik

No	Pertanyaan	Alternatif			
		SS/SB	S/B	TS/TB	STS/STB
1	Bagaimana program SIRM bekerja di komputer yang Anda gunakan?				
2	Bagaimana kinerja fungsi login yang ada dalam program SIRM?				
3	Bagaimana kinerja fungsi melihat daftar mahasiswa yang ada dalam program SIRM?				
4	Bagaimana kinerja fungsi melihat info mahasiswa yang ada dalam program SIRM?				
5	Bagaimana kinerja fungsi melihat histori yang ada dalam program SIRM?				
6	Bagaimana kinerja fungsi membuat entri baru yang ada dalam program SIRM?				
7	Bagaimana akurasi fungsi login yang ada dalam program SIRM?				
8	Bagaimana akurasi fungsi melihat daftar mahasiswa yang ada dalam program SIRM?				
9	Bagaimana akurasi fungsi melihat info mahasiswa yang ada dalam program SIRM?				
10	Bagaimana akurasi fungsi melihat histori yang ada dalam program SIRM?				
11	Bagaimana akurasi fungsi membuat entri baru yang ada dalam program SIRM?				
12	Apakah program SIRM sudah cukup mudah dalam penggunaannya?				
13	Apakah program SIRM membantu pemahaman Anda dalam mengelola riwayat mahasiswa?				
14	Apakah Anda merasa bekerja dengan program SIRM bisa lebih cepat dalam mengingat riwayat mahasiswa?				
15	Apakah Anda merasa bekerja dengan program SIRM bisa lebih akurat dalam mengingat riwayat mahasiswa?				