

TUGAS AKHIR

SISTEM INFORMASI RIWAYAT MAHASISWA



SAMUEL HERMAN

NPM: 2010730013

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

FINAL PROJECT

STUDENT INFORMATION HISTORY RECORDING SYSTEM



SAMUEL HERMAN

NPM: 2010730013

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

DAFTAR ISI

DAFTAR ISI	v
DAFTAR GAMBAR	vii
DAFTAR TABEL	x
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 Authentication [1]	5
2.1.1 Langkah Dasar	5
2.1.2 Skenario Google Authentication	7
2.1.3 Masa Habis Berlaku Token	12
2.1.4 Lingkup Otorisasi [2]	13
2.2 Markdown	16
2.2.1 Apa itu Markdown? [3]	16
2.2.2 Sintaks yang Berguna [3]	16
2.2.3 GitHub Flavored Markdown [4]	23
2.3 StrapdownJS [5]	27
2.4 Zurb Foundation [6]	29
2.4.1 Kompatibilitas	29
2.4.2 Apa Saja yang Dapat Dibuak Dengan Foundation?	29
3 ANALISIS	41
3.1 Analisis Google Authentication	41
3.1.1 Langkah Dasar Penggunaan OAuth 2.0	41
3.1.2 Skenario Aplikasi	46
3.2 Analisis Markdown	47
3.3 Analisis StrapdownJS	50
3.4 Analisis Zurb Foundation	51
3.5 Analisis Kebutuhan Perangkat Lunak	53
3.5.1 Use Case Diagram	54
3.5.2 Skenario	55
3.5.3 Entity Relationship Diagram (ERD)	58
3.5.4 Data Flow Diagram	58
4 PERANCANGAN	63

4.1	Perancangan Tampilan Web Yang Digunakan	63
4.1.1	Tampilan Halaman Awal	63
4.1.2	Tampilan Web Pilih Mahasiswa	63
4.1.3	Tampilan Web Info Mahasiswa	64
4.1.4	Tampilan Web Edit Mahasiswa	64
4.1.5	Tampilan Web Lihat Histori	65
4.1.6	Tampilan Web Lihat Versi Ini	66
4.1.7	Tampilan Web Entri Baru	66
4.2	Perancangan Modul	67
4.2.1	Modul Login	67
4.2.2	Modul Pilih Mahasiswa	67
4.2.3	Modul Info Mahasiswa	67
4.2.4	Modul Edit Mahasiswa	67
4.2.5	Modul Lihat Histori	68
4.2.6	Modul Entri Baru	68
4.3	Perancangan Tabel Sistem Informasi Riwayat Mahasiswa	68
4.3.1	Perancangan Tabel Info Mahasiswa	68
4.3.2	Perancangan Tabel Histori	69
4.4	Diagram Sekuens	69
5	IMPLEMENTASI DAN PENGUJIAN PERANGKAT LUNAK	73
5.1	Implementasi Perangkat Lunak	73
5.1.1	Lingkungan Implementasi Perangkat Keras	73
5.1.2	Lingkungan Implementasi Perangkat Lunak	74
5.1.3	Hasil Implementasi Perangkat Lunak	74
5.2	Pengujian Perangkat Lunak	74
5.2.1	Lingkungan Pengujian Perangkat Keras	74
5.2.2	Lingkungan Pengujian Perangkat Lunak	75
5.2.3	Pengujian Fungsional	75
5.2.4	Hasil Pengujian Fungsional	85
5.2.5	Pengujian Eksperimental	86
	DAFTAR REFERENSI	91

DAFTAR GAMBAR

2.1	Gambar Skenario Aplikasi Web Server	8
2.2	Gambar Skenario Aplikasi yang Terinstal	9
2.3	Gambar Skenario Aplikasi Sisi Klien (JavaScript)	10
2.4	Gambar Skenario Aplikasi Pada Perangkat Dengan Masukan Yang Terbatas	11
2.5	Gambar Skenario Layanan Akun	12
2.6	Gambar Markdown Cetak Tebal dan Cetak Miring	17
2.7	Gambar Markdown Judul Bab	17
2.8	Gambar Markdown Batas Baris Dengan Tiga Spasi	18
2.9	Gambar Markdown Batas Baris Tanpa Tiga Spasi	18
2.10	Gambar Markdown Paragraf	19
2.11	Gambar Markdown Menampilkan Gambar	19
2.12	Gambar Markdown Link	20
2.13	Gambar Markdown Kode	20
2.14	Gambar Markdown Kutipan	21
2.15	Gambar Markdown Garis Horisontal	22
2.16	Gambar Markdown Daftar Tidak Berurutan	23
2.17	Gambar Markdown Daftar Berurutan	23
2.18	Gambar GFM Garis Bawah	24
2.19	Gambar GFM Taut Otomatis URL	24
2.20	Gambar GFM Tanda Coret	24
2.21	Gambar GFM Blok Kode	25
2.22	Gambar GFM Penandaan Sintaks	25
2.23	Gambar GFM Tabel 1	26
2.24	Gambar GFM Tabel 2	26
2.25	Gambar GFM Tabel 3	26
2.26	Gambar GFM Tabel 4	27
2.27	Gambar GFM Tabel 5	27
2.28	Gambar Contoh Grid	30
2.29	Gambar Grid Utama	31
2.30	Gambar Grid Kecil	31
2.31	Gambar Grid Menengah	32
2.32	Gambar Grid Tingkat Lanjut	32
2.33	Gambar Grid Dengan Offset	33
2.34	Gambar Grid Dengan Baris yang Tidak Lengkap	33
2.35	Gambar Grid Dengan Baris yang Mengempis/Tidak Mengempis	34
2.36	Gambar Grid Dengan Kolom yang Berpusat	35
2.37	Gambar Grid Dengan Sumber yang Terurut	35
2.38	Gambar Contoh Tombol	36
2.39	Gambar Tombol Dasar	36
2.40	Gambar Ukuran dan Bentuk Tombol	37
2.41	Gambar Warna-warna Tombol	37
2.42	Gambar Aksesibilitas Tombol	38

2.43	Gambar Tabel Foundation	39
2.44	Gambar Contoh Navigasi	39
2.45	Gambar Contoh Plugins	40
3.1	Google Developers Console	43
3.2	Membuat Proyek Baru	43
3.3	Menu Credentials	44
3.4	Membuat Client ID yang Baru	44
3.5	Tipe Aplikasi	45
3.6	Pengisian Tipe Aplikasi	45
3.7	Client ID	46
3.8	Izin Pihak Pengguna	46
3.9	Skenario Aplikasi SIRM	47
3.10	Output Sintaks Cetak Tebal dan Cetak Miring	48
3.11	Output Sintaks Judul Bab	49
3.12	Output Sintaks Batas Baris	49
3.13	Output Sintaks Paragraf	49
3.14	Output Sintaks Link	49
3.15	Output Sintaks Daftar	49
3.16	Output Keterangan Mahasiswa	50
3.17	Tampilan pilihmahasiswa.php Dengan Zurb Foundation	53
3.18	Use Case Diagram Sistem Informasi Riwayat Mahasiswa	54
3.19	Entity Relationship Diagram	58
3.20	Data Context Diagram Sistem Informasi Riwayat Mahasiswa	59
3.21	Data Flow Diagram Sistem Informasi Riwayat Mahasiswa	59
4.1	Desain Antarmuka Halaman Awal	63
4.2	Desain Antarmuka Pilih Mahasiswa	64
4.3	Desain Antarmuka Info Mahasiswa	64
4.4	Desain Antarmuka Edit Mahasiswa	65
4.5	Desain Antarmuka Lihat Histori	65
4.6	Desain Antarmuka Lihat Versi Ini	66
4.7	Desain Antarmuka Entri Baru	66
4.8	Diagram Sekuens Bagian Satu	70
4.9	Diagram Sekuens Bagian Dua	70
4.10	Diagram Sekuens Bagian Tiga	71
5.1	Membuka Halaman index.php	78
5.2	Login Dengan Email yang Diakhiri "@student.unpar.ac.id"	78
5.3	Konfirmasi Email yang Dikelola oleh student.unpar.ac.id	79
5.4	CAS UNPAR	80
5.5	Izin Akses Dari Pihak Pengguna	80
5.6	Login Dengan Email yang Diakhiri "@gmail.com"	81
5.7	Alert Email yang Digunakan Tidak Dapat Mengakses SIRM	81
5.8	Memilih Mahasiswa	82
5.9	Melihat Info Mahasiswa	82
5.10	Mengedit Info Mahasiswa	82
5.11	Melihat Histori	83
5.12	Keterangan Versi Pertama	83
5.13	Keterangan Versi Kedua	84
5.14	Template Entri Baru	84
5.15	Membuat Entri Baru	85

5.16 Entri Baru Berhasil Dibuat 85

DAFTAR TABEL

2.1	Tabel 2-1 Daftar Pengujian Zurb Foundation	29
3.1	Tabel 3-1 Skenario Login	55
3.2	Tabel 3-2 Skenario Pilih Mahasiswa	56
3.3	Tabel 3-3 Skenario Melihat Info Mahasiswa	56
3.4	Tabel 3-4 Skenario Edit Mahasiswa	57
3.5	Tabel 3-5 Skenario Lihat Histori	57
3.6	Tabel 3-6 Skenario Membuat Entri Baru	58
4.1	Tabel 4-1 Modul Login	67
4.2	Tabel 4-2 Modul Pilih Mahasiswa	67
4.3	Tabel 4-3 Modul Info Mahasiswa	68
4.4	Tabel 4-4 Modul Edit Mahasiswa	68
4.5	Tabel 4-5 Modul Lihat Histori	68
4.6	Tabel 4-6 Modul Entri Baru	69
4.7	Tabel 4-7 Rancangan Tabel Info Mahasiswa	69
4.8	Tabel 4-8 Rancangan Tabel Histori	69
5.1	Tabel 5-1 Hasil Pengujian Fungsional	85
5.2	Tabel 5-2 Catatan Waktu Pengujian Eksperimental Kelompok Pertama	87
5.3	Tabel 5-3 Catatan Waktu Pengujian Eksperimental Kelompok Kedua	88
5.4	Tabel 5-4 Perbandingan Rata-rata Waktu Pengujian Eksperimental	89

BAB 1

PENDAHULUAN

1.1 Latar Belakang

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

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

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

1.2 Rumusan Masalah

Berdasarkan latar belakang maka dapat dirumuskan permasalahan sebagai berikut:

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

8 1.3 Tujuan

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

- 11 • Mengautentikasi pengguna menggunakan Google Authentication.
- 12 • Menggunakan teks dengan format Markdown Syntax.
- 13 • Menampilkan teks dengan format Markdown Syntax ke halaman website.
- 14 • Merancang antarmuka Sistem Informasi Riwayat Mahasiswa menggunakan Zurb Fo-
- 15 undation.
- 16 • Mengimplementasikan Sistem Informasi Riwayat Mahasiswa yang telah dirancang ke-
- 17 dalam *script* PHP.

18 1.4 Batasan Masalah

19 Dalam penelitian ini ditetapkan batasan-batasan yang akan menjadi pedoman dalam pelak-
20 sanaan penelitian:

- 21 • Perangkat lunak akan memiliki 6 fitur yaitu: Login, Pilih mahasiswa, Melihat info
- 22 mahasiswa, Edit mahasiswa, Lihat histori, dan Membuat entri baru.
- 23 • Untuk fitur login hanya untuk dosen yang diakhiri dengan @unpar.ac.id dan *username*
- 24 bukan angka semua.

- 1 • Untuk fitur pilih mahasiswa, pengguna dapat memilih mahasiswa yang ingin dilihat
2 atau dirubah dan pengguna juga bisa menekan tombol "Add" untuk menambah ma-
3 hasiswa baru.
- 4 • Untuk fitur melihat info mahasiswa, pengguna dapat melihat info terkini dari maha-
5 siswa dan aksi ini dicatat dalam log untuk alasan penjagaan privasi.
- 6 • Untuk fitur edit mahasiswa, pengguna dapat mengubah info mahasiswa dan aksi ini
7 juga dicatat dalam log.
- 8 • Untuk fitur lihat histori, pengguna dapat melihat histori setiap aksi perubahan atau
9 aksi *view*.
- 10 • Untuk fitur membuat entri baru, saat membuat entri baru akan dibuatkan *template*
11 sehingga kedepannya isi info setiap mahasiswa seragam.

12 1.5 Metodologi Penelitian

13 Metodologi yang digunakan untuk menyusun penelitian:

- 14 • Melakukan studi pustaka mengenai teknologi yang akan digunakan untuk membangun
15 Sistem Informasi Riwayat Mahasiswa.
- 16 • Menganalisis cara kerja teknologi yang akan digunakan untuk membangun Sistem
17 Informasi Riwayat Mahasiswa.
- 18 • Merancang Sistem Informasi Riwayat Mahasiswa yang akan dibuat.
- 19 • Melakukan implementasi untuk Sistem Informasi Riwayat Mahasiswa yang telah di-
20 rancang ke dalam PHP.
- 21 • Melakukan pengujian perangkat lunak yang telah diimplementasikan.

22 1.6 Sistematika Pembahasan

23 Sistematika pembahasan dalam penelitian ini adalah sebagai berikut:

- 24 • Bab I Pendahuluan
25 Bab ini menjelaskan latar belakang permasalahan, rumusan masalah, tujuan, batasan
26 masalah, metodologi penelitian, dan sistematika pembahasan.

1 • Bab II Dasar Teori

2 Bab ini menjelaskan teori-teori dasar mengenai Google Authentication, Markdown
3 Syntax, StrapdownJS, Zurb Foundation, PHP, dan MySQL yang menjadi refrensi
4 utama dalam pelaksanaan penelitian.

5 • Bab III Analisis

6 Bab ini berisi analisis mengenai Google Authentication, Markdown Syntax, Strapdo-
7 wnJS, Zurb Foundation, PHP, dan MySQL yang akan digunakan pada penelitian ini.

8 • Bab IV Perancangan

9 Bab ini berisi perancangan Sistem Informasi Riwayat Mahasiswa yang akan dibuat.

10 • Bab V Implementasi dan Pengujian

11 Bab ini berisi pengimplementasian dan pengujian Sistem Informasi Riwayat Mahasis-
12 wa.

13 • Bab VI Kesimpulan dan Saran

14 Bab ini berisi kesimpulan dari hasil penelitian dan saran untuk pengembangan lebih
15 lanjut.

BAB 2

LANDASAN TEORI

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

2.1 Google Authentication [1]

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

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

2.1.1 Langkah Dasar

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

1. Mendapatkan kepercayaan OAuth 2.0 dari Google Developers Console

Berkunjung ke Google Developers Console untuk mendapatkan kepercayaan OAuth 2.0 seperti klien id dan kerahasiaan klien yang keduanya dikenal oleh Google dan

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

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

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

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

3. Kirim token akses ke API

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

1 4. Memperbaharui token akses jika diperlukan

2 Token akses memiliki daya tahan yang terbatas. Jika aplikasi yang dibangun membu-
3 tuhkan akses ke Google API melebihi masa aktif token akses, maka dapat memperba-
4 harui token akses tersebut. Hal ini memungkinkan untuk mendapatkan token akses
5 yang baru.

6 2.1.2 Skenario Google Authentication

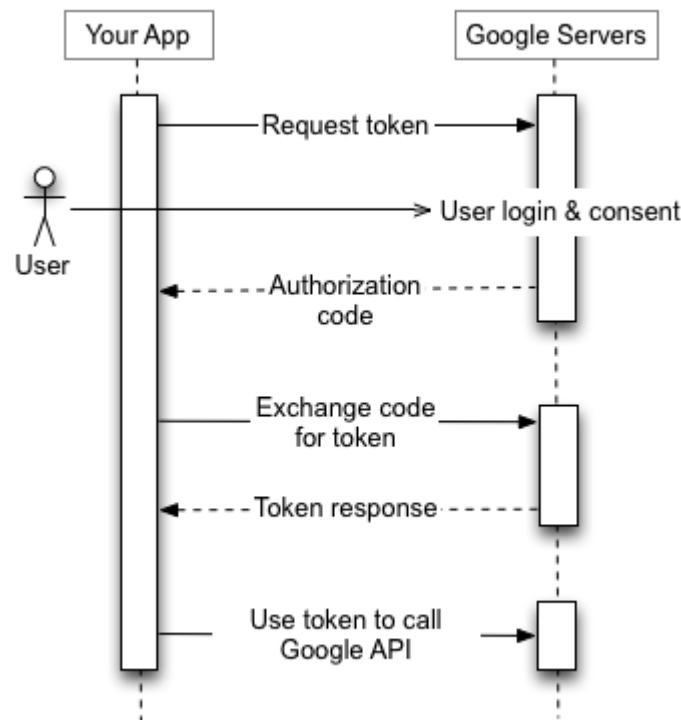
7 Terdapat lima skenario yang dapat digunakan untuk Google Authentication yaitu Skenario
8 Aplikasi Web Server, Skenario Aplikasi yang Terinstal, Skenario Aplikasi Sisi Klien (Java-
9 Script), Skenario Aplikasi Pada Perangkat Dengan Masukan Yang Terbatas, dan Skenario
10 Layanan Akun. Untuk penjelasan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

11 2.1.2.1 Skenario Aplikasi Layanan Web

12 Google OAuth 2.0 mendukung aplikasi Layanan Web yang menggunakan bahasa dan ke-
13 rangka kerja seperti PHP, Java, Python, Ruby, dan ASP.NET.

14 Urutan otorisasi dimulai ketika aplikasi mengarahkan mesin pencari ke URL Google;
15 URL tersebut termasuk parameter permintaan yang menunjukkan jenis akses yang diminta.
16 Google menangani otentikasi pengguna, pemilihan sesi, dan izin dari pihak pengguna. Ha-
17 silnya adalah sebuah kode otorisasi, dimana aplikasi dapat bertukar untuk token akses dan
18 memperbaharui token akses.

19 Aplikasi harus menyimpan pembaharuan token akses untuk penggunaan kedepannya
20 dan menggunakan token akses untuk mengakses Google API. Setelah masa token akses
21 berakhir, maka aplikasi akan memperbaharui token akses untuk mendapatkan yang baru.
22 Untuk gambaran skenario dapat dilihat pada Gambar [2.1](#).



Gambar 2.1: Skenario Aplikasi Layanan Web

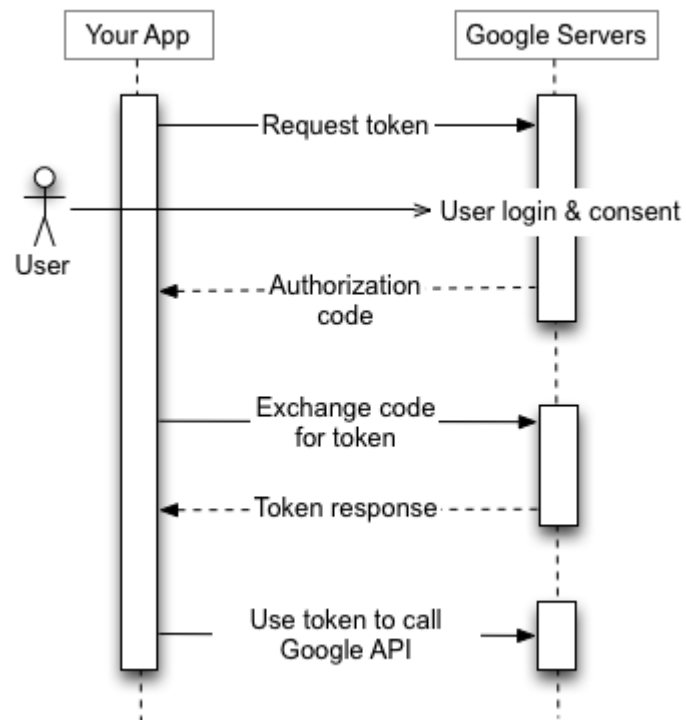
1 2.1.2.2 Skenario Aplikasi yang Terinstal

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

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

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

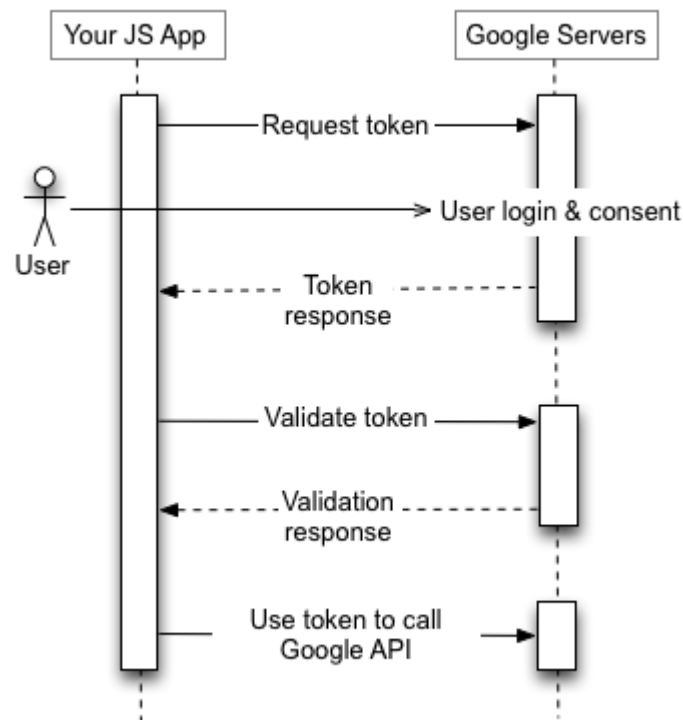
13 Aplikasi harus menyimpan token yang diperbaharui untuk penggunaan masa depan dan
 14 menggunakan token akses untuk mengakses API Google. Setelah masa token akses berakhir,
 15 maka aplikasi akan memperbaharui token untuk mendapatkan yang baru. Untuk gambar
 16 skenario dapat dilihat pada Gambar 2.2.



Gambar 2.2: Skenario Aplikasi yang Terinstal

1 2.1.2.3 Skenario Aplikasi Sisi Klien (JavaScript)

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

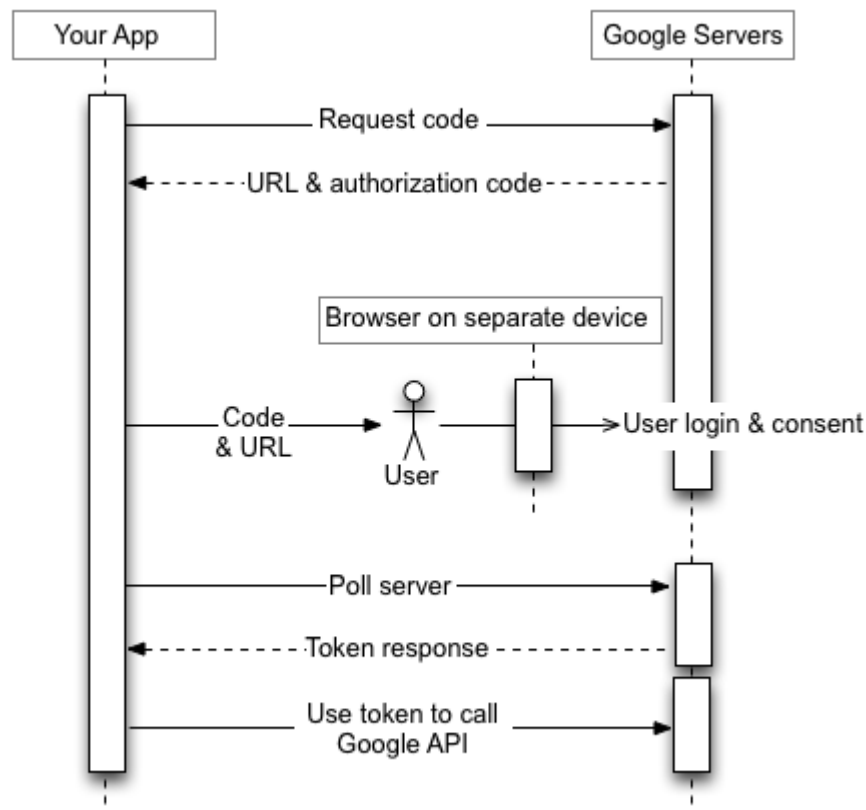


Gambar 2.3: Skenario Aplikasi Sisi Klien (JavaScript)

1 2.1.2.4 Skenario Aplikasi Pada Perangkat Dengan Masukan Yang Terbatas

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

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



Gambar 2.4: Skenario Aplikasi Pada Perangkat Dengan Masukan Yang Terbatas

2.1.2.5 Skenario Layanan Akun

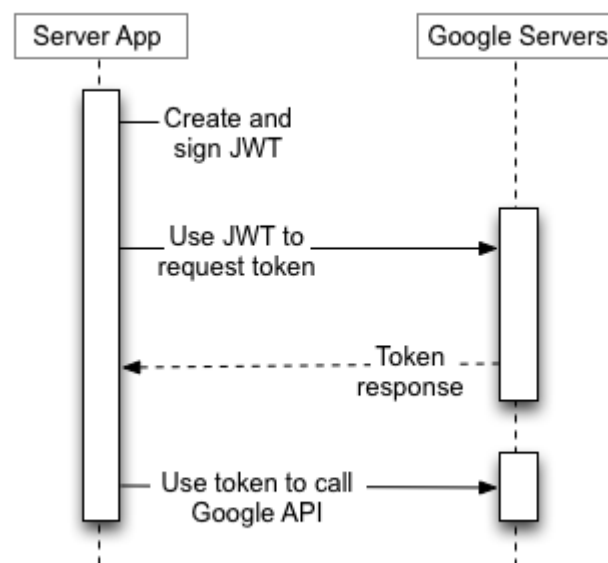
Google API seperti Prediction API dan Google Cloud Storage dapat bertindak atas nama aplikasi yang dibuat tanpa mengakses informasi pengguna. Dalam situasi ini aplikasi perlu membuktikan identitasnya sendiri ke API, tapi tidak diperlukan izin dari pihak pengguna. Demikian pula, dalam skenario perusahaan,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

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

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

1 akan memiliki dampak yang parah pada keamanan aplikasi yang dibangun.

2 Kredensial ayanan akun, yang diperoleh dari Google Developers Console, termasuk ala-
 3 mat email yang dihasilkan yang unik, klien id, dan setidaknya satu pasang kunci publik /
 4 privat. Menggunakan klien id dan satu kunci privat untuk membuat JWT ditandatanga-
 5 ni dan membangun permintaan token akses dalam format yang sesuai. Aplikasi kemudian
 6 mengirimkan permintaan token ke Google OAuth 2.0 Authorization Server, yang mengem-
 7 balikan token akses. Aplikasi menggunakan token untuk mengakses API Google. Ketika
 8 masa token berakhir, aplikasi mengulangi proses. Untuk gambar skenario dapat dilihat
 9 pada Gambar 2.5.



Gambar 2.5: Skenario Layanan Akun

10 2.1.3 Masa Habis Berlaku Token

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

- 14 • Pengguna telah mencabut akses.
- 15 • Token tidak digunakan selama enam bulan.
- 16 • Akun pengguna telah melampaui jumlah tertentu permintaan token.

17 Saat ini batas untuk setiap akun Google adalah 25 token. Jika pengguna akun telah
 18 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 memasukan 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 mewakilkan pengguna yang telah teroten-
29 tikasi disetiap permintaan yang dilakukan.
- 30 • Memungkinkan aplikasi diakses melalui aplikasi android.

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

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

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

8 Lingkup ini juga memungkinkan lintas platform dengan pendaftaran tunggal.

9 2.1.4.2 Lingkup Email

10 | `email`

11 Lingkup ini meminta agar aplikasi diberikan akses ke:

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

14 | `people.getOpenIdConnect`

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

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

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

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

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

22 Lingkup ini meminta aplikasi agar diberikan akses ke:

- 23 • Alamat email Google pengguna yang telah diverifikasi di *profile* Google+. Mengakses
24 email dengan memanggil `people.get` dan mengembalikan hasil email dalam array.
- 25 • Nama domain yang telah didaftarkan di Google jika pengguna memiliki fitur tersebut.

2.1.4.3 Lingkup yang lain

2 | `openid`

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

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

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

9 Untuk keperluan login menggunakan lingkup profil atau lingkup

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

11 karena lingkup

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

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

15 Lingkup ini melakukan hal berikut:

- 16 • Memungkinkan aplikasi untuk mengetahui siapa pengguna yang dikonfirmasi dengan
17 mengganti id pengguna dengan "*me*" yang mewakili pengguna yang telah teroten-
18 tikasi disetiap permintaan yang dilakukan.

19 2.1.4.4 Lingkup yang tidak dipakai lagi

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

21 Ganti dengan lingkup yang setara yaitu lingkup profil. Lingkup ini setara dengan lingkup
22 profil dan meminta akses data yang sama.

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

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

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

30 Lihat juga lingkup terkait:

31 | `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 list. 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.

```
<p><em>hello </em> untuk cetak miring</p>
<p><strong>hello </strong> untuk cetak tebal</p>
```

hello untuk cetak miring

hello untuk cetak tebal

Gambar 2.6: Markdown Cetak Tebal dan Cetak Miring

2.2.2.2 Judul Bab

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

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

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

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

Judul Bab

Judul Sub Bab

Judul Sub Sub Bab

Tingkat ke 4

Tingkat ke 5

Tingkat ke 6

Gambar 2.7: Markdown Judul Bab

2.2.2.3 Batas Baris

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

```
4 | Baris ini dengan
5 | batas baris
```

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

```
9 | <p>Baris ini dengan <br />
10| batas baris</p>
```

Baris ini dengan
batas baris

Gambar 2.8: Markdown Batas Baris Dengan Tiga Spasi

```
11| Baris ini tanpa
12| batas baris
```

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

```
16| <p>Baris ini tanpa
17| batas baris</p>
```

Baris ini tanpa batas baris

Gambar 2.9: Markdown Batas Baris Tanpa Tiga Spasi

2.2.2.4 Paragraf

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

```
21| Ini kalimat pertama. Ini kalimat berikutnya. Ini kalimat terakhir.
22|
23| Ini paragraf baru.
```

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

```
27| <p>Ini kalimat pertama. Ini kalimat berikutnya. Ini kalimat terakhir.</p>
28|
29| <p>Ini paragraf baru.</p>
```

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

Gambar 2.10: Markdown Paragraf

2.2.2.5 Gambar

Untuk menyisipkan gambar pada dokumen Markdown, gunakan sintaks berikut

```
! [ teks ] ( / url _ gambar " judul _ gambar ")
```

Contoh penggunaan sintaks gambar:

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

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

```
< p > < img src = " https : / / www . google . com / logos / doodles / 2014 / googles - 16th - birthday - 4613606054297600 - hp .
gif " alt = " logo " title = " Google " / > < / p >
```



Gambar 2.11: Markdown Menampilkan Gambar

2.2.2.6 Link

Untuk menyisipkan hyperlink pada dokumen Markdown, gunakan sintaks berikut

```
[ link _ teks ] ( / tujuan _ url " judul _ opsional ")
```

Contoh penggunaan sintaks link:

```
[ my _ website ] ( http : / / browsernative . com " Click Here ")
```

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

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



Gambar 2.12: Markdown Link

1 2.2.2.7 Kode

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

```

6 | Definisi dari 'initLabels()' dapat dilihat di bawah ini"
7 |
8 | ```
9 |     function initLabels(){
10 |         function setLabels(elementId, messageId){
11 |             var label = document.querySelector('label[for=' + elementId + ']');
12 |             label.textContent = chrome.i18n.getMessage(messageId);
13 |         }
14 |     }
15 | ```
16 |
17 | Paragraf lain.
```

- 18 'initLabels()' akan menjadi kode pada satu baris kalimat. Sedangkan fungsi yang ada di da-
 19 lam `` akan menjadi blok kode. Sintaks tersebut akan menghasilkan kode HTML sebagai
 20 berikut. Untuk hasil pada HTML dapat dilihat pada Gambar 2.13.

```

21 | <p>Definisi dari 'initLabels()' dapat dilihat di bawah ini"</p>
22 |
23 | <p>```
24 |     function initLabels(){
25 |         function setLabels(elementId, messageId){
26 |             var label = document.querySelector('label[for=' + elementId + ']');
27 |             label.textContent = chrome.i18n.getMessage(messageId);
28 |         }
29 |     }
30 | ```</p>
31 |
32 | <p>Paragraf lain.</p>
```

```

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

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

Paragraf lain.
```

Gambar 2.13: Markdown Kode

2.2.2.8 Kutipan

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

```

4 | Ini merupakan kutipan favorit saya:
5 |
6 | > The weak can never forgive.
7 | > Forgiveness is the attribute of the strong.

```

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

```

10 | <p>Ini merupakan kutipan <strong>favorit</strong> saya:</p>
11 |
12 | <blockquote>
13 |   <p>The weak can never forgive.
14 |   Forgiveness is the attribute of the strong.</p>
15 | </blockquote>

```

Ini merupakan kutipan **favorit** saya:

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

Gambar 2.14: Markdown Kutipan

2.2.2.9 Garis Horisontal

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

```

19 | # Bagian Satu
20 |
21 | The quick brown fox jumps over the lazy dog. The quick brpwn fox jump over the lazy dog.
22 |
23 | ———
24 |
25 | Last Edited on *25th Dec 2014*

```

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

```

28 | <h1>Bagian Satu</h1>
29 |
30 | <p>The quick brown fox jumps over the lazy dog. The quick brpwn fox jump over the lazy dog.</p>
31 |
32 | <hr />
33 |
34 | <p>Last Edited on <em>25th Dec 2014</em></p>

```

Bagian Satu

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

Last Edited on *25th Dec 2014*

Gambar 2.15: Markdown Garis Horisontal

2.2.2.10 Daftar

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

1. Daftar tidak berurutan

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

```
* Item
* Item
* Item

+ Item
+ Item
+ Item

- Item
- Item
- Item
```

Sintaks tersebut akan menghasilkan kode HTML sebagai berikut. Untuk hasil pada HTML dapat dilihat pada Gambar [2.16](#).

```
<ul>
<li>Item</li>
<li>Item</li>
<li><p>Item</p></li>
<li><p>Item</p></li>
<li>Item</li>
<li><p>Item</p></li>
<li><p>Item</p></li>
<li>Item</li>
<li>Item</li>
</ul>
```


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

Gambar 2.16: Daftar Tidak Berurutan

2. Daftar berurutan

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

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

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

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

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

Gambar 2.17: Markdown Daftar Berurutan

2.2.3 GitHub Flavored Markdown [4]

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

2.2.3.1 Beberapa Garis Bawah Pada Kalimat

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

```
wow_great_stuff
do_this_and_do_that_and_another_thing.
```

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

wow_great_stuff do_this_and_do_that_and_another_thing.

Gambar 2.18: GFM Garis Bawah

2.2.3.2 Taut Otomatis URL

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

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

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

<http://www.google.com>

Gambar 2.19: GFM Taut Otomatis URL

2.2.3.3 Tanda Coret

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

```
| ~Mistaken text.~
```

Untuk hasil pada HTML dapat dilihat pada Gambar 2.20.

~~Mistaken text.~~

Gambar 2.20: GFM Tanda Coret

2.2.3.4 Blok Kode

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

```

5 | Here 's an example :
6 |
7 | ```
8 | function test() {
9 |     console.log("notice the blank line before this function?");
10 | }
11 | ```

```

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 code sintaks Ruby.

```

17 | ```ruby
18 | require 'redcarpet'
19 | markdown = Redcarpet.new("Hello World!")
20 | puts markdown.to_html
21 | ```

```

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.

```

27 | First Header | Second Header
28 | -----

```

```

1 | Content Cell | Content Cell
2 | Content Cell | Content Cell

```

3 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

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

```

6 | First Header | Second Header |
7 | _____ | _____ |
8 | Content Cell | Content Cell |
9 | Content Cell | Content Cell |

```

10 Untuk hasil pada HTML dapat dilihat pada Gambar 2.24.

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

Gambar 2.24: GFM Tabel 2

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

```

13 | Name | Description |
14 | _____ | _____ |
15 | Help | Display the help window. |
16 | Close | Closes a window |

```

17 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

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

```

1 | | Name | Description |
2 | |-----|-----|
3 | | Help | ~Display the~ help window. |
4 | | Close | _Closes_ a window |

```

5 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

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

```

8 | | Left-Aligned | Center Aligned | Right Aligned |
9 | | :-----: | :-----: | :-----: |
10 | | col 3 is | some wordy text | $1600 |
11 | | col 2 is | centered | $12 |
12 | | zebra stripes | are neat | $1 |

```

13 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

14 2.3 StrapdownJS [5]

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

19 Untuk penggunaan StrapdownJS gunakan skrip strapdown.js dengan cara langsung meng-
20 arahkan ke website maupun menggunakan path yang mengarahkan dimana file strapdown.js
21 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 adipiscing elit, sed do eiusmod tempor incididunt ut
labore
et dolore magna aliqua.

## Chapter 2

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut
aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse
cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in
culpa qui officia deserunt mollit anim id est laborum.
</xmp>

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

Strapdown.js juga memiliki beberapa fitur :

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

Tabel 2.1: Daftar Pengujian Zurb Foundation¹

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

2.4 Zurb Foundation [6]

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

2.4.1 Kompatibilitas

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

2.4.2 Apa Saja yang Dapat Dibuat Dengan Foundation?

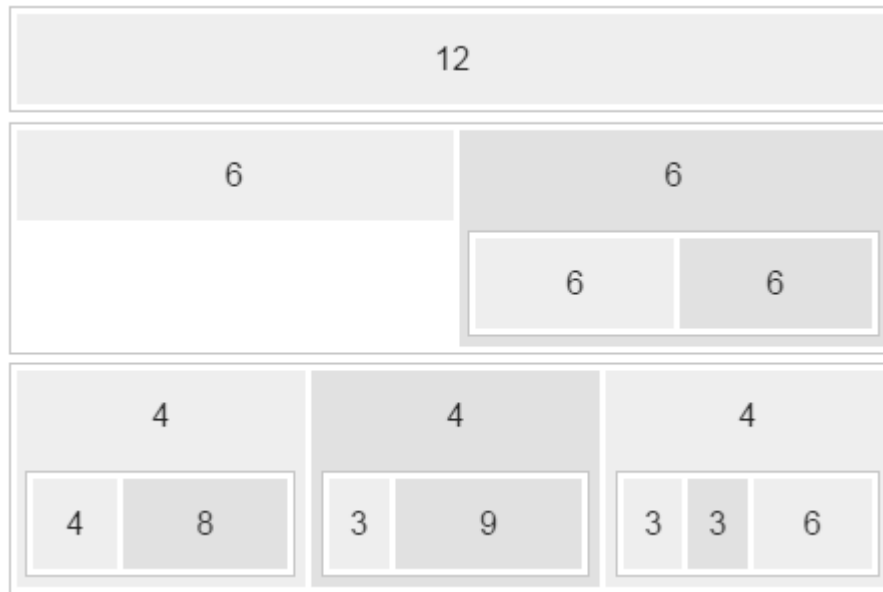
Foundation memiliki banyak komponen dan struktur untuk membantu membangun sebuah situs responsif. Untuk komponen Foundation dapat melihat beberapa gambar dibawah ini :

1 Grid

Grid bekerja pada hampir semua perangkat dan memiliki dukungan untuk menjadi satu kesatuan, sumber pemesanan, offset dan perangkat presentasi. Hal tersebut sedikit mudah dengan waktu yang singkat dapat menciptakan tata letak yang kompleks seperti ini.

Untuk contoh grid dapat dilihat pada Gambar 2.28.

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



Gambar 2.28: Contoh Grid

• Grid Utama

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

```

10 <!-- no. 1 -->
11 <div class="row">
12   <div class="small-2 large-4 columns">2</div>
13   <div class="small-4 large-4 columns">4</div>
14   <div class="small-6 large-4 columns">6</div>
15 </div>
16 <!-- no. 2 -->
17 <div class="row">
18   <div class="large-3 columns">full</div>
19   <div class="large-6 columns">full</div>
20   <div class="large-3 columns">full</div>
21 </div>
22 <!-- no. 3 -->
23 <div class="row">
24   <div class="small-6 large-2 columns">6</div>
25   <div class="small-6 large-8 columns">6</div>
26   <div class="small-12 large-2 columns">full</div>
27 </div>
28 <!-- no. 4 -->
29 <div class="row">
30   <div class="small-3 columns">3</div>
31   <div class="small-9 columns">9</div>
32 </div>

```



```

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

```

1.	2	4	6
2.	full	full	full
3.	6	6	full
4.	3	9	
5.	full	full	
6.	6	6	
7.	full	full	

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.

```

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

```

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

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.

```

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

```



Gambar 2.31: Grid Menengah

• Grid Tingkat Lanjut

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

```

17 <div class="row">
18   <div class="small-8 columns">8
19     <div class="row">
20       <div class="small-8 columns">8 Nested
21         <div class="row">
22           <div class="small-8 columns">8 Nested Again</div>
23           <div class="small-4 columns">4</div>
24         </div>
25       </div>
26       <div class="small-4 columns">4</div>
27     </div>
28   </div>
29   <div class="small-4 columns">4</div>
30 </div>

```



Gambar 2.32: Grid Tingkat Lanjut

• Grid Dengan Offset

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

```

35 <!-- no. 1 -->
36 <div class="row">

```

```

1      <div class="large-1 columns">1</div>
2      <div class="large-11 columns">11</div>
3  </div>
4  <!-- no. 2 -->
5  <div class="row">
6      <div class="large-1 columns">1</div>
7      <div class="large-10 large-offset-1 columns">10, offset 1</div>
8  </div>
9  <!-- no. 3 -->
10 <div class="row">
11     <div class="large-1 columns">1</div>
12     <div class="large-9 large-offset-2 columns">9, offset 2</div>
13 </div>
14 <!-- no. 4 -->
15 <div class="row">
16     <div class="large-1 columns">1</div>
17     <div class="large-8 large-offset-3 columns">8, offset 3</div>
18 </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 memin-dahkan 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.

```

25 <!-- no. 1 -->
26 <div class="row">
27     <div class="medium-3 columns">3</div>
28     <div class="medium-3 columns">3</div>
29     <div class="medium-3 columns">3</div>
30 </div>
31 <!-- no. 2 -->
32 <div class="row">
33     <div class="medium-3 columns">3</div>
34     <div class="medium-3 columns">3</div>
35     <div class="medium-3 columns end">3 end</div>
36 </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.

```
<div class="row medium-uncollapse large-collapse">
  <div class="small-6 columns">
    Removes gutter at large media query
  </div>
  <div class="small-6 columns">
    Removes gutter at large media query
  </div>
</div>
```

On a large screen, I have no gutters!

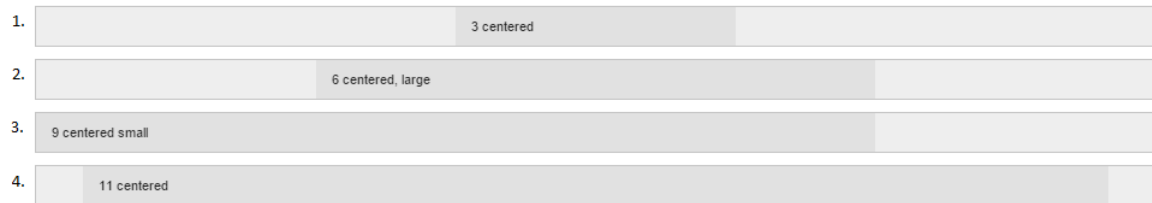
On a large screen, I have no gutters!

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.

```
<!-- no. 1 -->
<div class="row">
  <div class="small-3 small-centered columns">3 centered</div>
</div>
<!-- no. 2 -->
<div class="row">
  <div class="small-6 large-centered columns">6 centered</div>
</div>
<!-- no. 3 -->
<div class="row">
  <div class="small-9 small-centered large-uncentered columns">9 centered</div>
</div>
<!-- no. 4 -->
<div class="row">
  <div class="small-11 small-centered columns">11 centered</div>
</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.

```

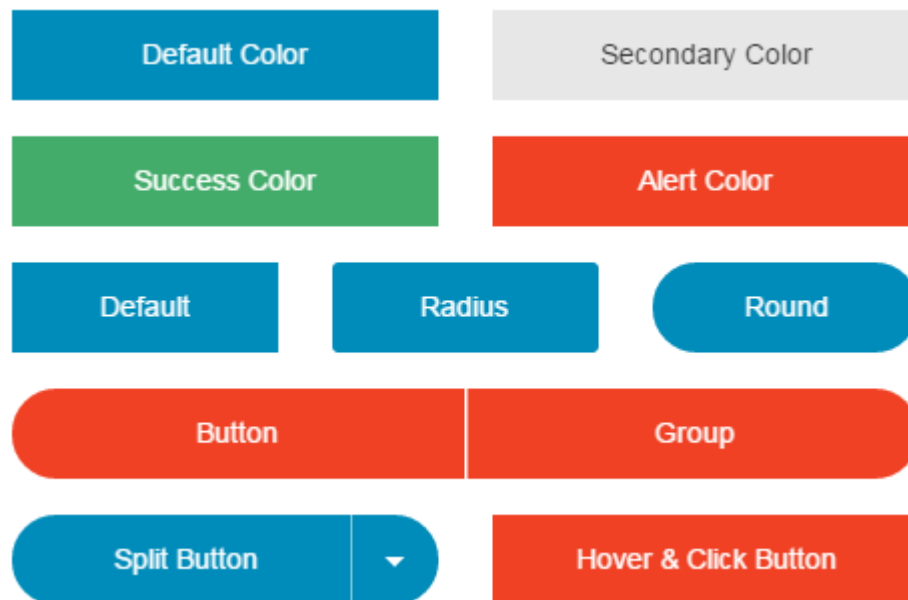
1  <!-- no. 1 -->
2  <div class="row">
3    <div class="small-10 small-push-2 columns">10</div>
4    <div class="small-2 small-pull-10 columns">2, last</div>
5  </div>
6  <!-- no. 2 -->
7  <div class="row">
8    <div class="large-9 large-push-3 columns">9</div>
9    <div class="large-3 large-pull-9 columns">3, last</div>
10 </div>
11 <!-- no. 3 -->
12 <div class="row">
13   <div class="large-8 large-push-4 columns">8</div>
14   <div class="large-4 large-pull-8 columns">4, last</div>
15 </div>
16 <!-- no. 4 -->
17 <div class="row">
18   <div class="small-5 small-push-7 medium-7 medium-push-5 columns">7</div>
19   <div class="small-7 small-pull-5 medium-5 medium-pull-7 columns">5, last</div>
20 </div>
21 <!-- no. 5 -->
22 <div class="row">
23   <div class="medium-6 medium-push-6 columns">6</div>
24   <div class="medium-6 medium-pull-6 columns">6, last</div>
25 </div>

```



Gambar 2.37: Grid Dengan Sumber yang Terurut

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



Gambar 2.38: Contoh Tombol

6 • Tombol Utama

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

9 `Default Button`



Gambar 2.39: Tombol Dasar

10 • Mengubah Ukuran Tombol

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

13 `<!-- Kelas untuk merubah ukuran -->`

14 `1. Tiny Button`

15 `2. Small Button`

16 `3. Default Button`

17 `4. Disabled Button`

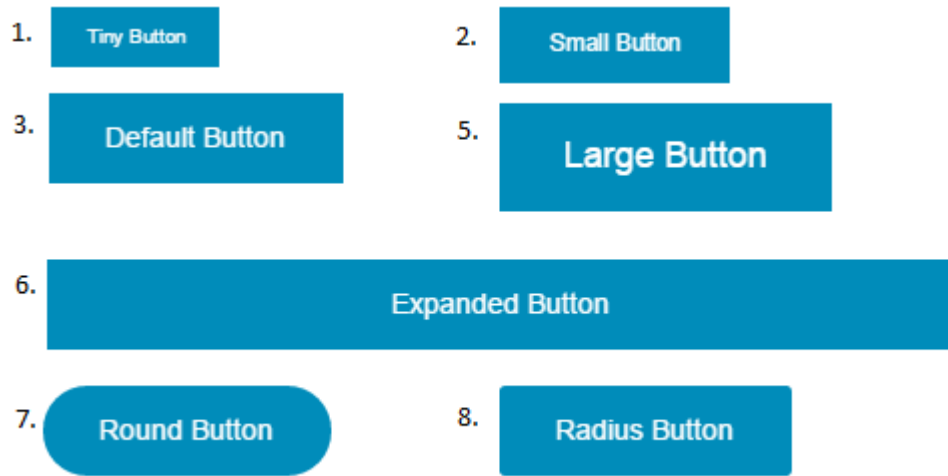
18 `5. Large Button`

19 `6. Expanded Button`

```

1      <!-- Kelas untuk merubah bentuk -->
2      7. <a href="#" class="button round">Round Button</a>
3      8. <a href="#" class="button radius">Radius Button</a>

```



Gambar 2.40: Ukuran dan Bentuk Tombol

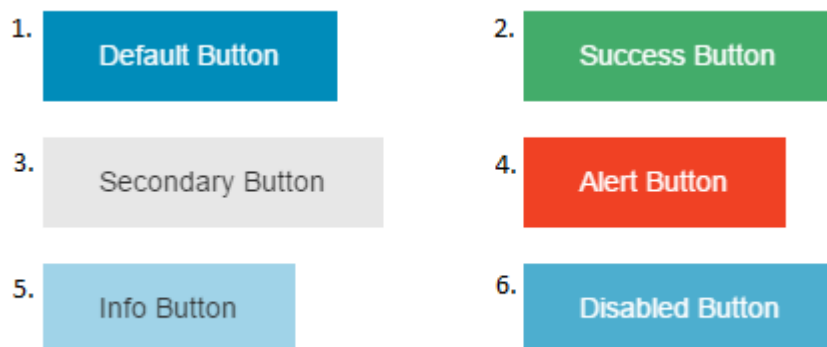
• Warna Tombol

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

```

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

```



Gambar 2.41: Warna-warna Tombol

• Aksesibilitas

Gunakan kode HTML dibawah ini untuk membuat tombol lebih mudah diakses. Dapat menggunakan atribut `aria-label='submit form'` untuk memberikan petunjuk jika

tidak ada keterangan pada tombol. Jika tombol tidak memiliki `` maka cukup menambahkan `tabindex="0"` pada div atau span untuk membuat tombol tersebut dapat difokuskan.

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



Gambar 2.42: Aksesibilitas Tombol

3 Tabel

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

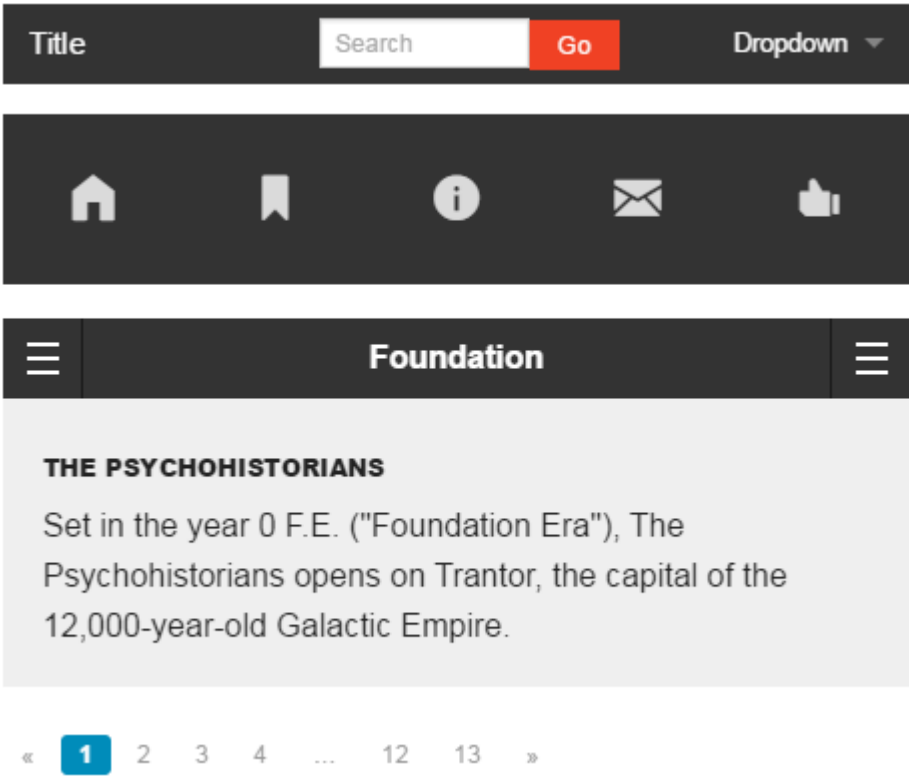
```
<table>
  <thead>
    <tr>
      <th width="200">Table Header</th>
      <th>Table Header</th>
      <th width="150">Table Header</th>
      <th width="150">Table Header</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Content Goes Here</td>
      <td>This is longer content Donec id elit non mi porta gravida at eget metus.</td>
      <td>Content Goes Here</td>
      <td>Content Goes Here</td>
    </tr>
    <tr>
      <td>Content Goes Here</td>
      <td>This is longer Content Goes Here Donec id elit non mi porta gravida at eget metus
        .</td>
      <td>Content Goes Here</td>
      <td>Content Goes Here</td>
    </tr>
    <tr>
      <td>Content Goes Here</td>
      <td>This is longer Content Goes Here Donec id elit non mi porta gravida at eget metus
        .</td>
      <td>Content Goes Here</td>
      <td>Content Goes Here</td>
    </tr>
  </tbody>
</table>
```


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

Gambar 2.43: Tabel Foundation

4 Navigasi

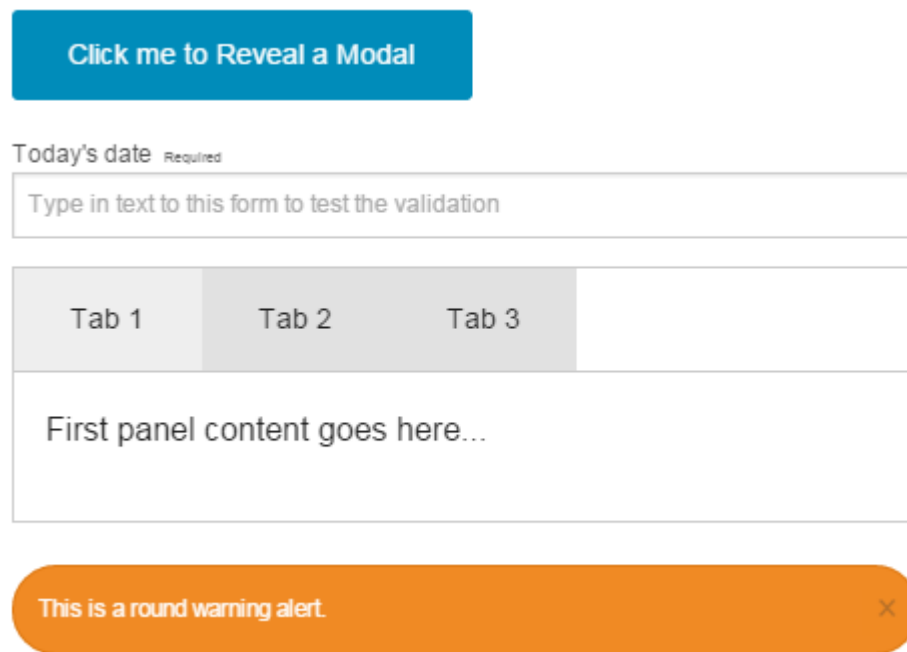
Orang yang mengakses harus bisa berkeliling melihat menu-menu yang ada. Gaya navigasi pada Foundation meliputi : bar bagian atas yang kuat dengan menu dropdown; tombol; bar pencari; ikon bar yang keren; implementasi kanvas yang lepas dari keluhan; dan sekelompok navigasi lainnya. Untuk contoh macam-macam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



Gambar 2.44: Contoh Navigasi

5 Plugins

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



Gambar 2.45: Contoh Plugins

BAB 3

ANALISIS

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

3.1 Analisis Google Authentication

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

3.1.1 Langkah Dasar Penggunaan OAuth 2.0

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

1. Mendapatkan kepercayaan OAuth 2.0 dari Google Developers Console

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

- (b) Buat sebuah proyek baru. Dapat dilihat pada Gambar [3.2](#).

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

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

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

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

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

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

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

```
Client id :
568951368854-ufmbistn0pcaq0khubafo1a133orfge.apps.googleusercontent.com
Client secret :
-cSZ-AUmeQ9PaWWry_IpiBBi
```

Agar lebih jelas dapat dilihat pada Gambar 3.7.

2. Memperoleh token akses dari Google Authorization Server

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

3. Kirim token ke API

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

```

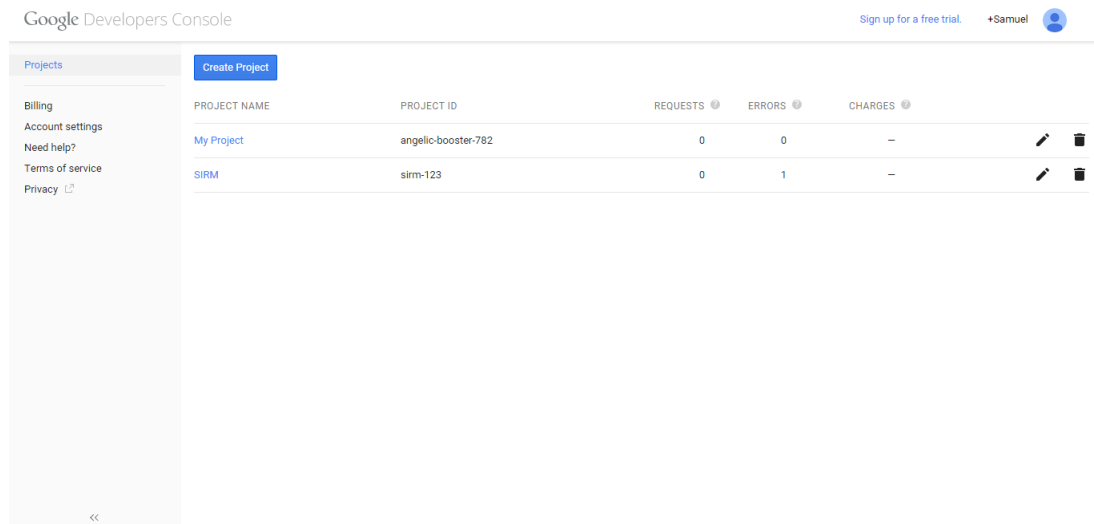
1 | https://www.googleapis.com/auth/plus.login
2 | dan
3 | email

```

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

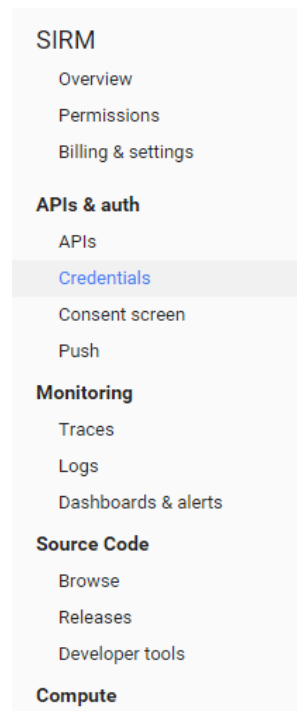
Pada penelitian ini tidak akan menggunakan tahap memperbaharui token akses karena token akses hanya digunakan selama penelitian ini berlangsung.



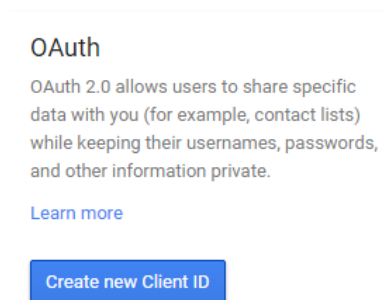
Gambar 3.1: Google Developers Console

The screenshot shows the 'New Project' dialog box. It contains two input fields: 'PROJECT NAME' with the value 'SIRM' and 'PROJECT ID' with the value 'sirm-1234'. Below the input fields is a link 'Show advanced options...'. At the bottom, there are two buttons: 'Create' and 'Cancel'.

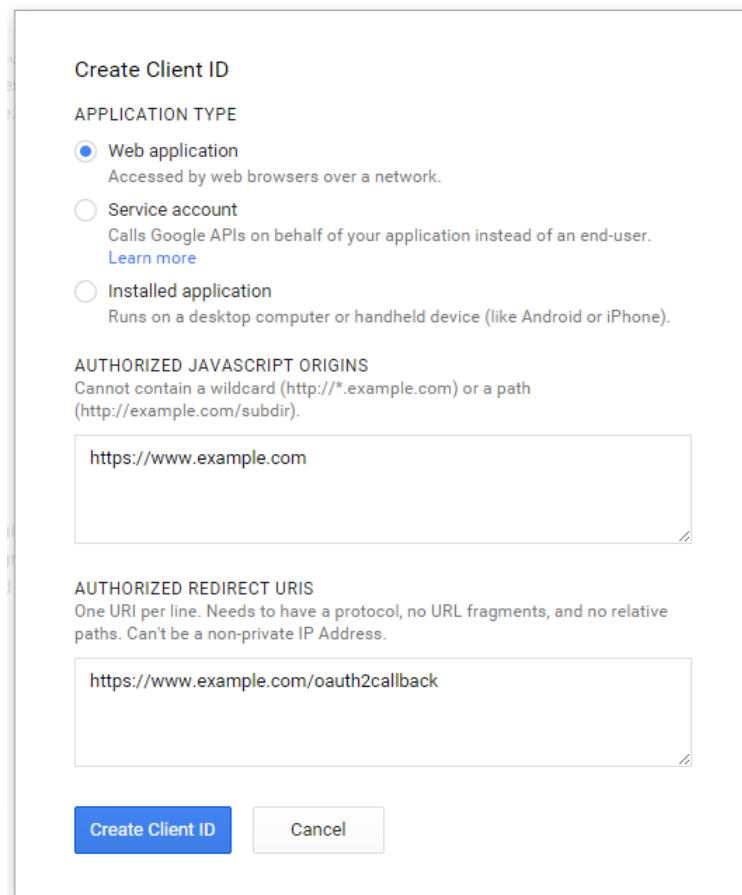
Gambar 3.2: Membuat Proyek Baru



Gambar 3.3: Menu Credentials



Gambar 3.4: Membuat Client ID yang Baru



Create Client ID

APPLICATION TYPE

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

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

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

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

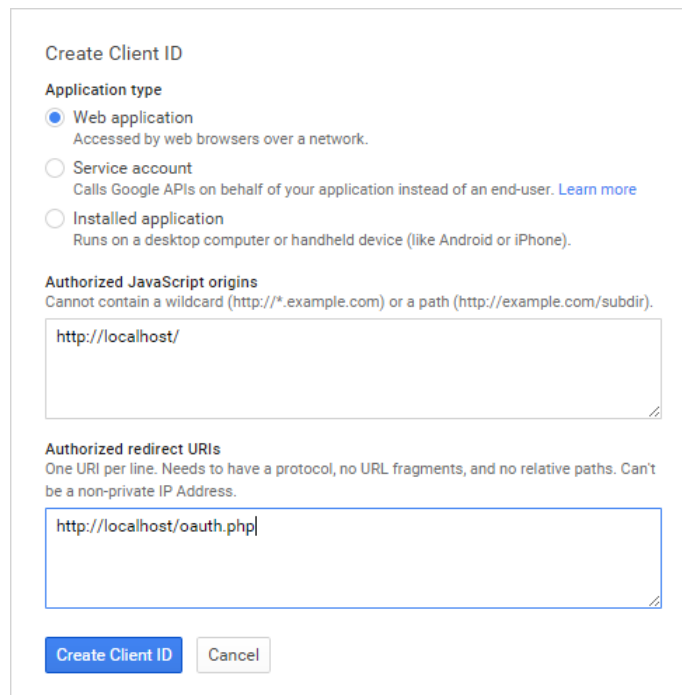
`https://www.example.com`

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

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

Create Client ID **Cancel**

Gambar 3.5: Tipe Aplikasi



Create Client ID

Application type

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

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

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

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

`http://localhost/`

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

`http://localhost/oauth.php`

Create Client ID **Cancel**

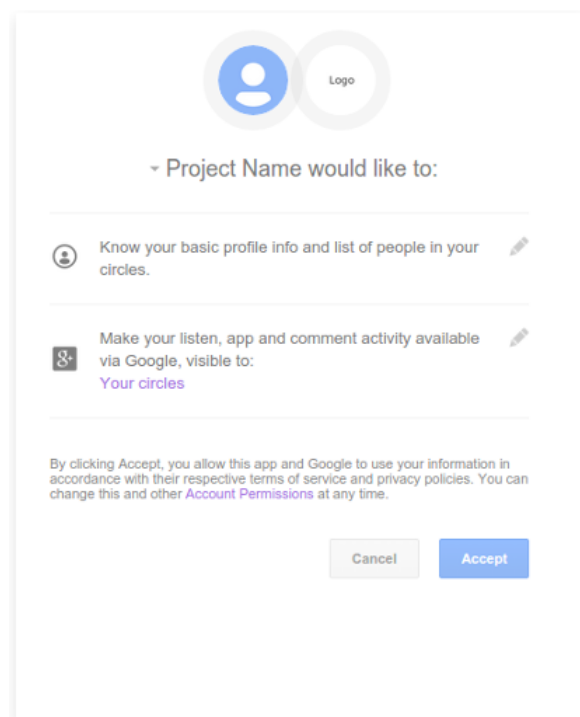
Gambar 3.6: Pengisian Tipe Aplikasi

Client ID for web application

CLIENT ID	568951368854-ufmbistn0pcaq0khubafo1a133orfgve.apps.googleusercontent.com
EMAIL ADDRESS	568951368854-ufmbistn0pcaq0khubafo1a133orfgve@developer.gserviceaccount.com
CLIENT SECRET	-cSZ-AUmeQ9PaWWry_lpiBBi
REDIRECT URIS	http://localhost/pilihmahasiswa.php
JAVASCRIPT ORIGINS	http://localhost/

[Edit settings](#)
[Reset secret](#)
[Download JSON](#)
[Delete](#)

Gambar 3.7: Client ID

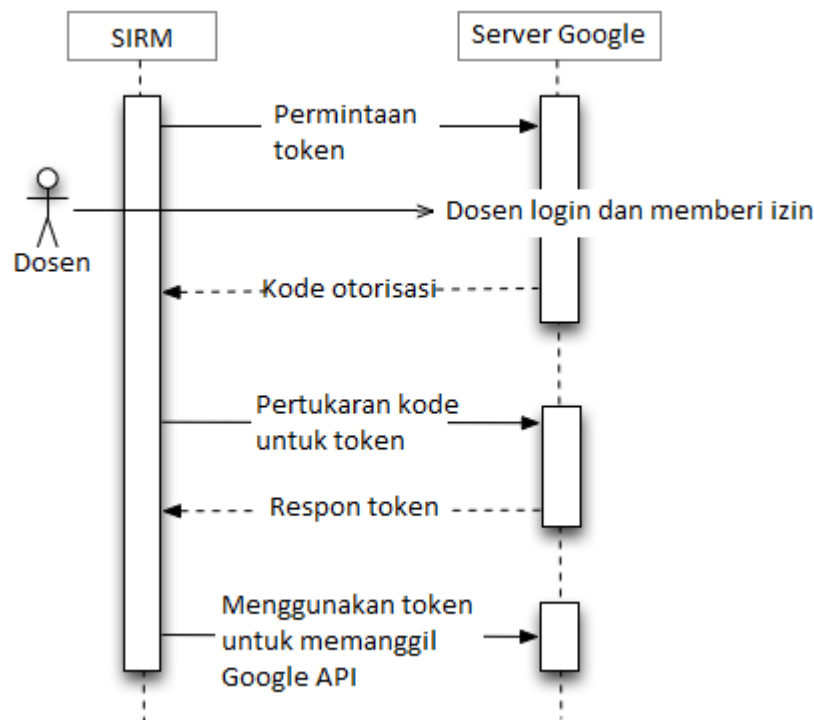


Gambar 3.8: Izin Pihak Pengguna

1 3.1.2 Skenario Aplikasi

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

- 1 akses, maka aplikasi dapat memanggil Google API dengan menggunakan token akses. Untuk
- 2 skenario aplikasi SIRM dapat dilihat pada Gambar 3.9.



Gambar 3.9: Skenario Aplikasi SIRM

3.2 Analisis Markdown

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

- 9 • Sintaks Cetak Tebal dan Cetak Miring

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

12 | ****NPM**** – *2010730013*

- 13 • Sintaks Judul Bab

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

```

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

```

• Sintaks Batas Baris

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

```

11     Baris ini dengan
12     batas baris
13
14     Baris ini tanpa
15     batas baris

```

• Sintaks Paragraf

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

```

19     Samuel adalah seorang mahasiswa yang periang namun terkadang sulit diatur. Dia aktif di
20     himpunan sebagai ketua divisi pelayanan masyarakat.
21
22     Grady adalah seorang mahasiswa yang memiliki jiwa pemimpin. Dia aktif di UKM sebagai ketua
23     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.

```

28     Yang bersangkutan memiliki blog di [http://bletaack.blogspot.com/](http://bletaack.blogspot.
29     com/).

```

• Daftar

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

```

33     * 9 Oktober 2014, bimbingan skripsi
34     * 3 Oktober 2014, bimbingan skripsi
35     * 1 September 2014, perwalian
36     * 1 September 2014, pertama kali dibuat

```

NPM - 2010730013

Gambar 3.10: Output Sintaks Cetak Tebal dan Cetak Miring

Judul 1

Judul 2

Judul 3

Judul 4

Judul 5

Judul 6

Gambar 3.11: Output Sintaks Judul Bab

Baris ini dengan
batas baris

Baris ini tanpa batas baris

Gambar 3.12: Output Sintaks Batas Baris

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

Gambar 3.13: Output Sintaks Paragraf

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

Gambar 3.14: Output Sintaks Link

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

Gambar 3.15: Output Sintaks Daftar

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

```

1  ### NPM
2
3  2010730013
4
5  ### Nama
6
7  Samuel
8
9  ### Umum
10
11 Samuel adalah seorang mahasiswa yang periang namun terkadang sulit diatur. Dia aktif di himpunan
12 sebagai ketua divisi pelayanan masyarakat. Yang bersangkutan memiliki blog di [http://bletack.
13 blogspot.com/](http://bletack.blogspot.com/).
14
15 ### Catatan
16
17 * 9 Oktober 2014, bimbingan skripsi
18 * 3 Oktober 2014, bimbingan skripsi
19 * 1 September 2014, perwalian
20 * 1 September 2014, pertama kali dibuat

```

NPM

2010730013

Nama

Samuel

Umum

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

Catatan

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

Gambar 3.16: Output Keterangan Mahasiswa

21 3.3 Analisis StrapdownJS

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

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

26 Skrip tersebut disisipkan pada skrip infomahasiswa.php yang berfungsi untuk menampilkan
 27 info mahasiswa yang dimana info tersebut ditulis menggunakan sintaks Markdown. Berikut
 28 skrip infomahasiswa.php yang menggunakan strapdown.js.

```

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 | Welcome</title>

```

```

1      <link rel="stylesheet" href="css/foundation.css" />
2      <script src="js/vendor/modernizr.js"></script>
3  </head>
4  <body>
5      <div class="row">
6          <h5>Anda melihat catatan mahasiswa ini sebagai test@unpar.ac.id.</h5>
7      </div>
8      <div class="row">
9          <ul class="button-group">
10             <li><a href="editmahasiswa.php" class="button">Edit</a></li>
11             <li><a href="lihathistori.php" class="button">Lihat Histori</a></li>
12          </ul>
13      </div>
14      <hr/>
15  <xmp style="display:none;">
16  ### NPM
17
18  2010730013
19
20  ### Nama
21
22  Samuel
23
24  ### Umum
25
26  Samuel adalah seorang mahasiswa yang periang namun terkadang sulit diatur. Dia aktif di himpunan
27  sebagai ketua divisi pelayanan masyarakat. Yang bersangkutan memiliki blog di [http://bletaack.
28  blogspot.com/](http://bletaack.blogspot.com/).
29
30  ### Catatan
31
32  * 9 Oktober 2014, bimbingan skripsi
33  * 3 Oktober 2014, bimbingan skripsi
34  * 1 September 2014, perwalian
35  * 1 September 2014, pertama kali dibuat
36
37  </xmp>
38      <script src="js/0.2/strapdown.js"></script>
39  </body>
40  </html>

```

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

42 3.4 Analisis Zurb Foundation

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

```

50 <!doctype html>
51 <html class="no-js" lang="en">
52   <head>

```

```

1      <meta charset="utf-8" />
2      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
3      <title>SIRM | Welcome</title>
4      <link rel="stylesheet" href="css/foundation.css" />
5      <script src="js/vendor/modernizr.js"></script>
6  </head>
7  <body>
8  <?php
9      session_start();
10     ?>
11     <div class="row">
12         <h3>Pilih NPM yang ingin dicari / tambah baru.</h3>
13         <ul class="button-group">
14             <li><a href="new.php" class="button">Add</a></li>
15             <li><a href="index.php?logout" class="button">Logout</a></li>
16         </ul>
17         <hr/>
18     </div>
19     <div class="row">
20     <?php
21         include_once "configDatabase.php";
22         $hasil = mysql_query("SELECT * FROM info_mahasiswa", $id_mysql);
23         if(! $hasil){
24             die("Permintaan gagal");
25         }
26
27         echo "<table>
28         <thead>
29         <tr>
30             <th width='250'>NPM</th>
31             <th width='500'>Nama</th>
32             <th width='250'>Last Update</th>
33         </tr>
34         </thead>";
35
36         while($row = mysql_fetch_array($hasil))
37         {
38             echo "<tr>";
39             //echo "<td><input class='button1' type='submit' name='submit' value='". $row['npm'] .'></td>";
40             echo "<td><a href='view.php?npm='". $row['npm'] .'>' . $row['npm'] . "</a></td>";
41             echo "<td>" . $row['nama'] . "</td>";
42             echo "<td>" . $row['pembaruan_terakhir'] . "</td>";
43             echo "</tr>";
44         }
45         echo "</table>";
46
47         mysql_close($id_mysql);
48     ?>
49     </div>
50 </body>
51 </html>

```

Pilih NPM yang ingin dicari / tambah baru.

Add

Logout

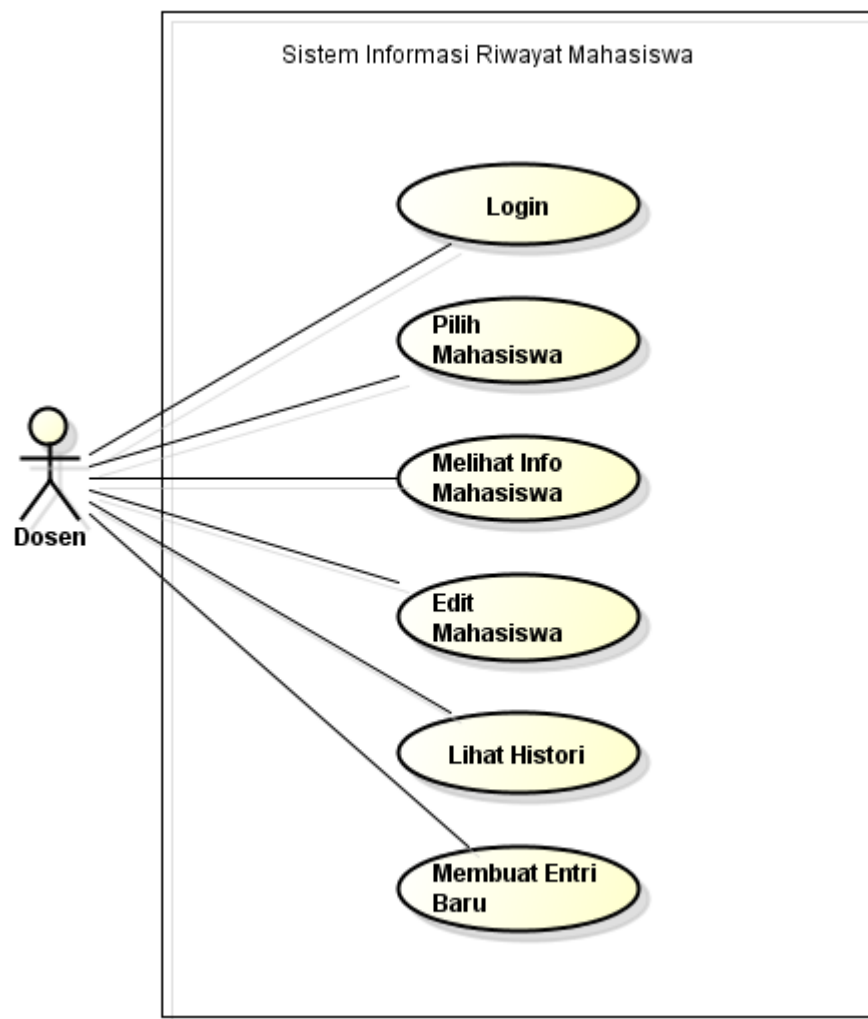
NPM	Nama	Last Update
2010120031	Kenneth Natanael	2015-04-30 14:29:51
2010730001	Andri Agustian	2015-04-29 15:09:33
2010730005	Grady Ireneus	2015-04-30 14:31:19
2010730012	Kevin PL	2015-04-24 00:55:57
2010730013	Samuel Herman	2015-04-24 00:23:34

Gambar 3.17: Tampilan pilihmahasiswa.php Dengan Zurb Foundation

1 3.5 Analisis Kebutuhan Perangkat Lunak

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

1 3.5.1 Use Case Diagram



Gambar 3.18: Use Case Diagram Sistem Informasi Riwayat Mahasiswa

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

5 • Login

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

7 • Pilih Mahasiswa

8 Use case ini memungkinkan pengguna untuk memilih mahasiswa yang ingin dilihat
9 infonya. Selain itu pengguna juga bisa menemukan tombol "Add" untuk menambah
10 entri baru.

11 • Melihat Info Mahasiswa

Tabel 3.1: Skenario Login

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

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

4 • Edit Mahasiswa

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

6 • Lihat Histori

7 Use case ini memungkinkan pengguna untuk melihat histori untuk setiap perubahan
 8 dan aksi view.

9 • Membuat Entri Baru

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

12 3.5.2 Skenario

13 3.5.2.1 Login

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

15 3.5.2.2 Pilih Mahasiswa

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

Tabel 3.2: Skenario Pilih Mahasiswa

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

Tabel 3.3: Skenario Melihat Info Mahasiswa

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

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 merekam 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	-		

1 3.5.2.3 Melihat Info Mahasiswa

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

3 3.5.2.4 Edit Mahasiswa

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

5 3.5.2.5 Lihat Histori

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

7 3.5.2.6 Membuat Entri Baru

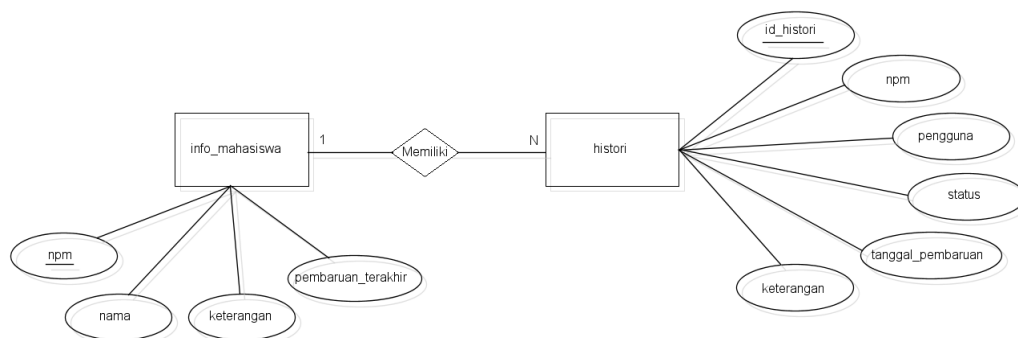
8 Untuk use case Membuat Entri Baru, skenarionya dapat dilihat pada Tabel 3.6.

Tabel 3.6: 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

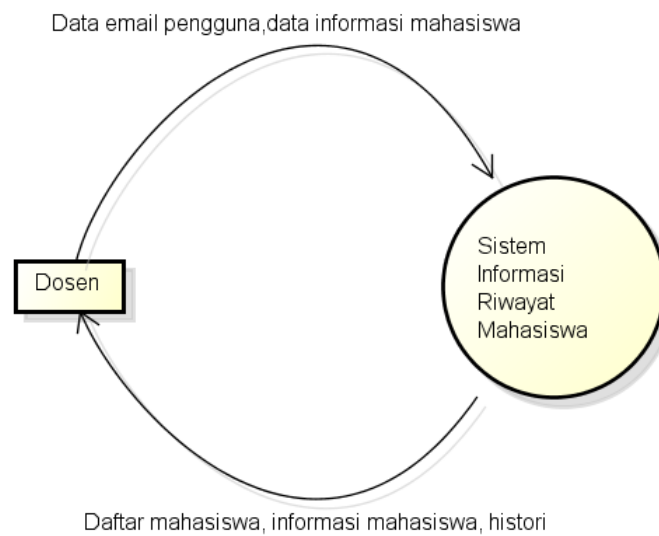
Pada ERD Sistem Informasi Riwayat Mahasiswa terdapat dua entitas yaitu info mahasiswa dan histori. Entitas info mahasiswa dengan relasi memiliki terhubung dengan entitas histori.

3.5.4 Data Flow Diagram

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

1 Data Context Diagram

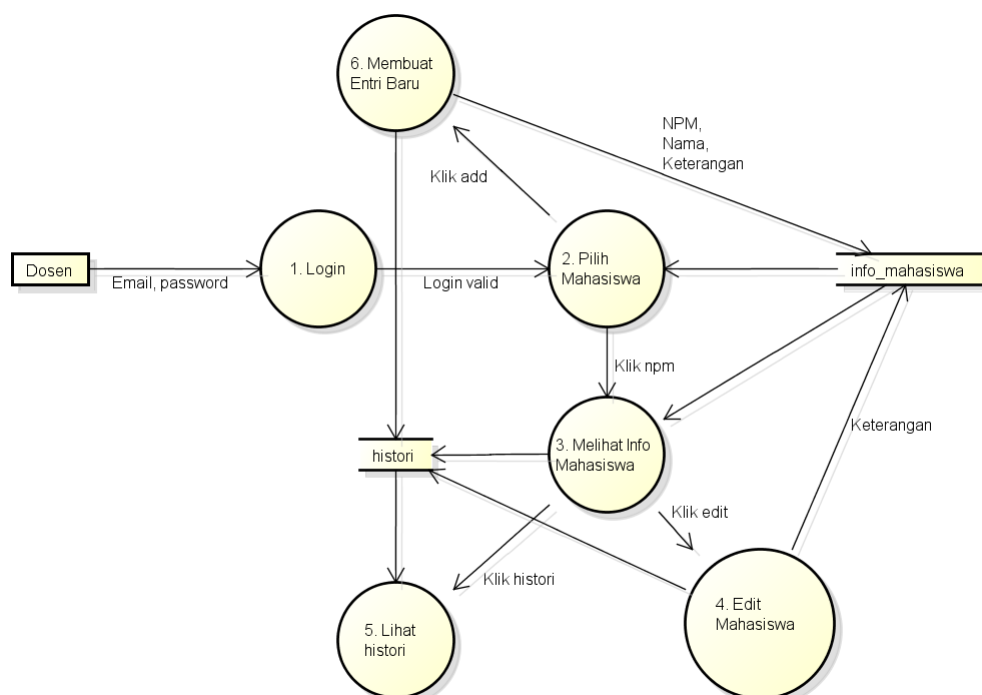
Data Context Diagram Sistem Informasi Riwayat Mahasiswa dapat dilihat pada Gambar 3.20.



Gambar 3.20: Data Context Diagram Sistem Informasi Riwayat Mahasiswa

2 Data Flow Diagram Level 1

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



Gambar 3.21: Data Flow Diagram Sistem Informasi Riwayat Mahasiswa

1 Untuk spesifikasi proses dapat dilihat di bawah ini.

2 • No dan Nama Proses : P1 Login

3 Deskripsi : Proses akses ke perangkat lunak

4 Data Input : Email, password

5 Data Output : Login valid

6 Proses : Mengecek email yang digunakan diakhiri @unpar.ac.id atau tidak. Jika ya
7 maka sistem akan mengarahkan ke halaman utama akan tetapi jika tidak maka
8 sistem akan menampilkan pesan login gagal.

9 • No dan Nama Proses : P2 Pilih Mahasiswa

10 Deskripsi : Proses memilih mahasiswa

11 Data Input : -

12 Data Output : Tabel daftar mahasiswa

13 Proses : pengguna memilih mahasiswa yang ingin dipilih dari tabel daftar mahasiswa.

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 pengguna
19 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 sebelumnya.

32 • No dan Nama Proses : P6 Membuat Entri Baru

-
- 1 Deskripsi : Proses membuat entri baru
 - 2 Data Input : NPM, Nama, Keterangan
 - 3 Data Output : -
 - 4 Proses : pengguna memasukan data NPM, Nama, Keterangan untuk entri baru lalu
 - 5 menyimpannya. Proses ini dicatat pada histori sebagai aksi membuat entri baru.

BAB 4

PERANCANGAN

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

4.1 Perancangan Tampilan Web Yang Digunakan

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

4.1.1 Tampilan Halaman Awal

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



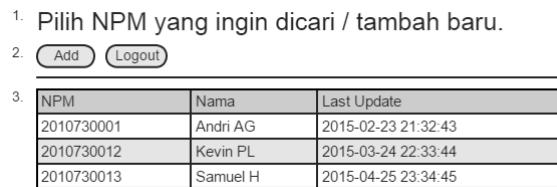
Gambar 4.1: Desain Antarmuka Halaman Awal

Keterangan :

- (1) Bagian ini merupakan judul yang merupakan keterangan dari perangkat lunak.
- (2) Bagian ini merupakan teks yang dapat diklik untuk melakukan login.

4.1.2 Tampilan Web Pilih Mahasiswa

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



Gambar 4.2: Desain Antarmuka Pilih Mahasiswa

1 Keterangan :

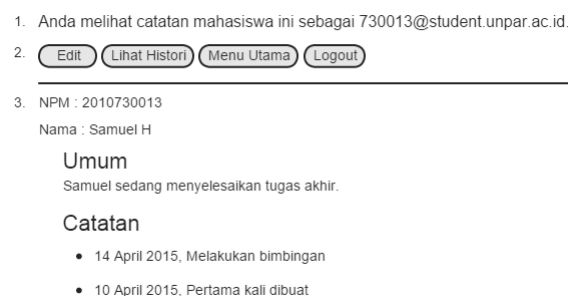
2 (1) Bagian ini merupakan judul dari halaman untuk memilih mahasiswa.

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

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

6 4.1.3 Tampilan Web Info Mahasiswa

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



Gambar 4.3: Desain Antarmuka Info Mahasiswa

8 Keterangan :

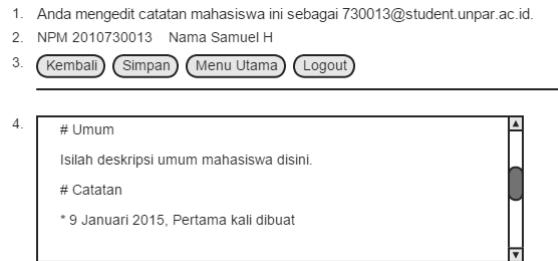
9 (1) Bagian ini merupakan teks yang menampilkan keterangan dan juga pengguna yang
10 sedang menggunakan Sistem Infomasi Riwayat Mahasiswa.

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

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

14 4.1.4 Tampilan Web Edit Mahasiswa

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



Gambar 4.4: Desain Antarmuka Edit Mahasiswa

1 Keterangan :

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

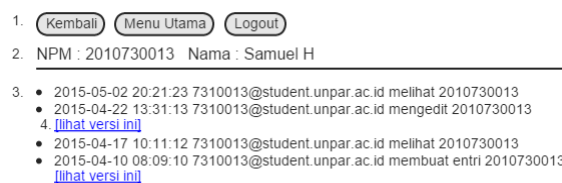
4 (2) Bagian ini merupakan teks yang menampilkan NPM dan nama mahasiswa yang telah
5 dipilih untuk diedit.

6 (3) Bagian ini merupakan tombol untuk melakukan aksi kembali, simpan untuk perubahan
7 yang telah dilakukan, pindah ke menu utama, dan logout.

8 (4) Bagian ini merupakan tempat menampilkan catatan mahasiswa yang berasal dari data-
9 base dan dapat diedit (ditulis dengan format markdown).

10 4.1.5 Tampilan Web Lihat Histori

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



Gambar 4.5: Desain Antarmuka Lihat Histori

12 Keterangan :

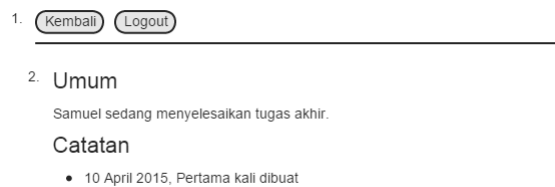
13 (1) Bagian ini merupakan tombol untuk melakukan aksi kembali, pindah ke menu utama,
14 dan logout.

15 (2) Bagian ini merupakan teks yang menampilkan keterangan NPM dan nama mahasiswa
16 yang telah dipilih untuk dilihat historinya.

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

4.1.6 Tampilan Web Lihat Versi Ini

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



Gambar 4.6: Desain Antarmuka Lihat Versi Ini

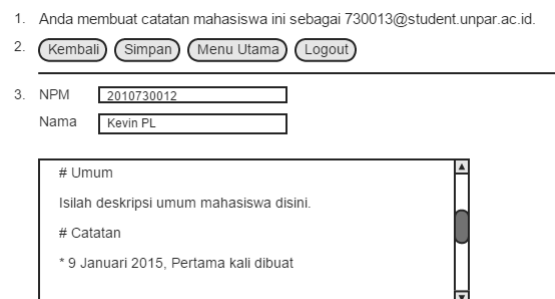
Keterangan :

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

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

4.1.7 Tampilan Web Entri Baru

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



Gambar 4.7: Desain Antarmuka Pilih Entri Baru

Keterangan :

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

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

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

Tabel 4.1: Modul Login

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

Tabel 4.2: Modul Pilih Mahasiswa

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

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

4.2 Perancangan Modul

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

4.2.1 Modul Login

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

4.2.2 Modul Pilih Mahasiswa

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

4.2.3 Modul Info Mahasiswa

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

4.2.4 Modul Edit Mahasiswa

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

Tabel 4.3: Modul Info Mahasiswa

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

Tabel 4.4: Modul Edit Mahasiswa

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

1 4.2.5 Modul Lihat Histori

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

3 4.2.6 Modul Entri Baru

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

5 4.3 Perancangan Tabel Sistem Informasi Riwayat Mahasiswa

6 4.3.1 Perancangan Tabel Info Mahasiswa

7 Untuk rancangan tabel info mahasiswa dapat dilihat pada Tabel 4.7.

Tabel 4.5: Modul Lihat Histori

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

Tabel 4.6: Modul Entri Baru

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

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	

4.3.2 Perancangan Tabel Histori

Untuk rancangan tabel histori dapat dilihat pada Tabel 4.8.

4.4 Diagram Sekuens

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

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

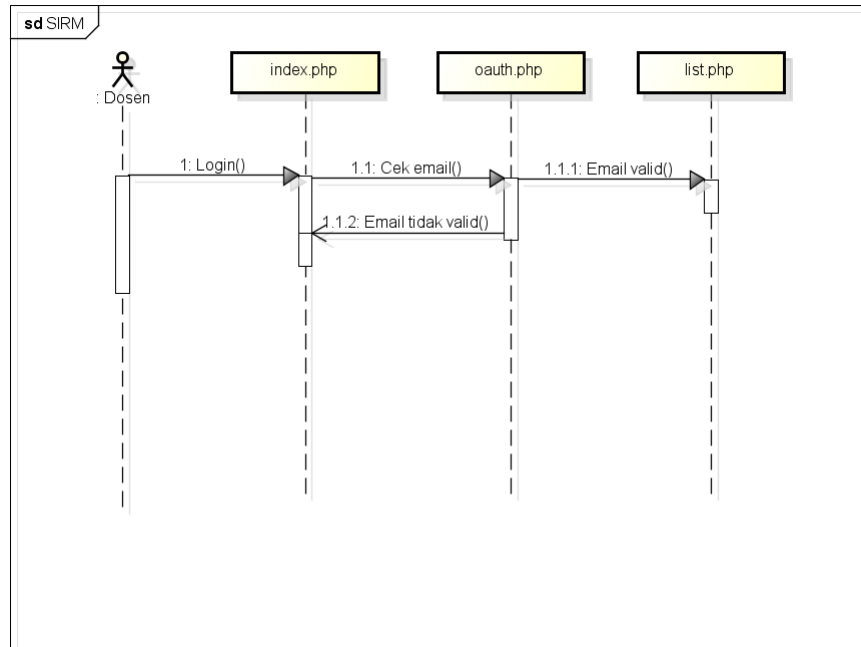
2 Sekuens bagian dua mencakup proses memilih mahasiswa, melihat info mahasiswa, dan membuat entri baru. Dapat dilihat pada Gambar 4.9.

3 Sekuens bagian tiga mencakup melihat info mahasiswa, mengedit info mahasiswa, melihat

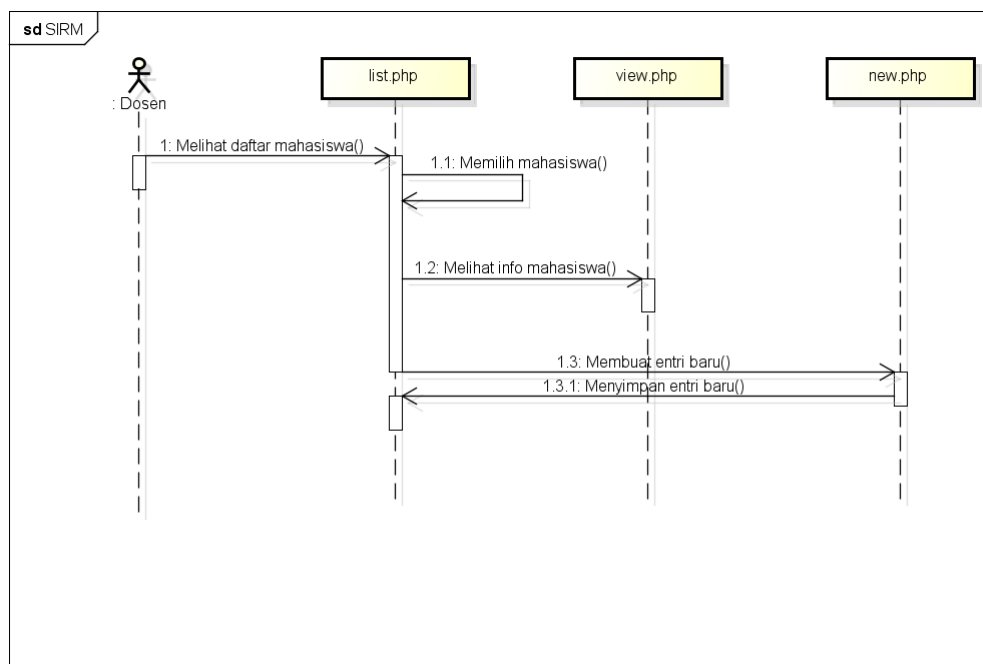
Tabel 4.8: Rancangan Tabel Histori

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	

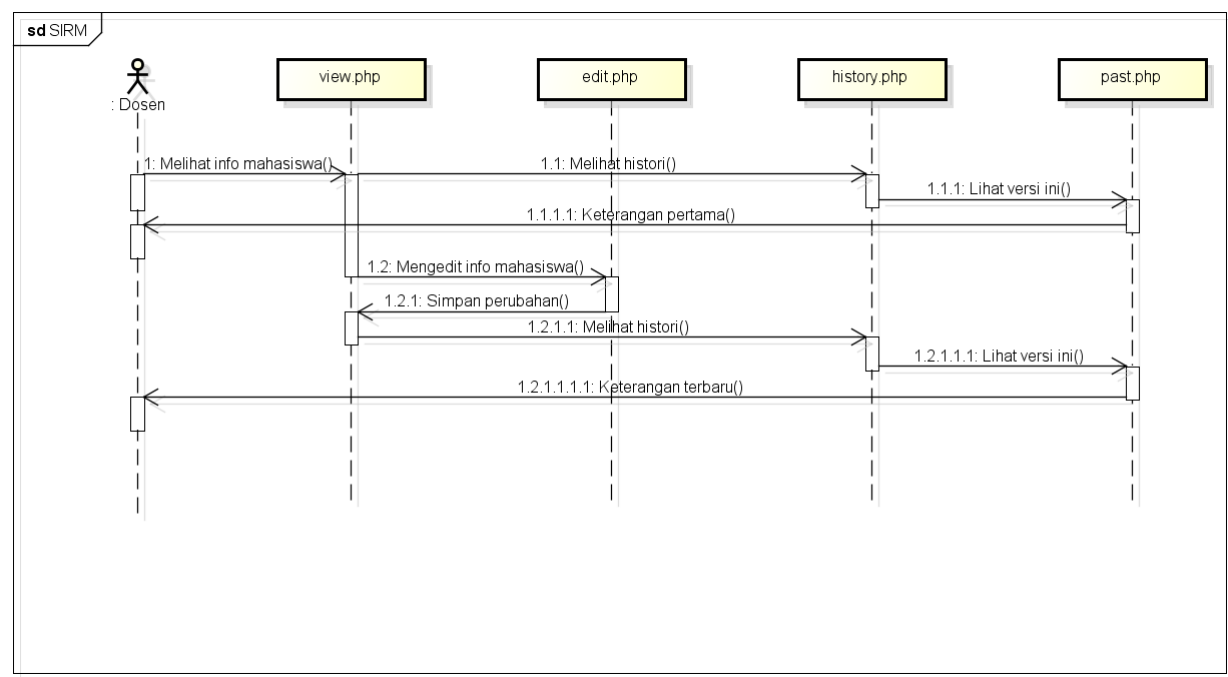
- 1 histori, dan melihat keterangan versi ini. Dapat dilihat pada Gambar 4.10.



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 dua bagian, yaitu Implementasi Perangkat Lunak dan Pengujian Per-
5 angkat Lunak. Bagian implementasi berisi penjelasan lingkungan pengembangan perangkat
6 lunak. Sedangkan bagian pengujian berisi hasil pengujian terhadap perangkat lunak yang
7 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

5.1.2 Lingkungan Implementasi Perangkat Lunak

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

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

5.1.3 Hasil Implementasi Perangkat Lunak

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

5.2 Pengujian Perangkat Lunak

Pada bagian ini akan dibahas mengenai pengujian yang akan dilakukan terhadap perangkat lunak. Pengujian tersebut terdiri dari dua bagian yaitu pengujian fungsional dan pengujian eksperimental. Pengujian fungsional bertujuan untuk memastikan bahwa seluruh fungsi yang dibangun pada perangkat lunak berjalan sesuai dengan yang direncanakan. Sedangkan pengujian eksperimental bertujuan untuk menguji eksepsi-eksepsi yang terdapat pada perangkat lunak. Pada bagian ini juga terdapat perubahan program pada bagian oauth.php, jadi dapat menjalankan pengujian dengan email yang diakhiri @student.unpar.ac.id dikarenakan penulis tidak memiliki email yang diakhiri @unpar.ac.id.

5.2.1 Lingkungan Pengujian Perangkat Keras

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

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

- 1 (b) RAM: 4 GB DDR3
- 2 (c) Harddisk: 1TB
- 3 (d) VGA: AMD Radeon HD 8650G 2GB
- 4 (e) Koneksi Internet: WAN

5 5.2.2 Lingkungan Pengujian Perangkat Lunak

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

- 7 (a) Sistem Operasi: Windows 8.1 Pro 64-bit
- 8 (b) Bahasa Pemrograman: PHP Version 5.6.3
- 9 (c) Aplikasi: XAMPP v5.6.3
- 10 (d) Aplikasi web browser: Google Chrome
- 11 (e) Library: Google APIs Client Library untuk PHP
- 12 (f) Javascript: Strapdown.js
- 13 (g) Framework: Foundation

14 5.2.3 Pengujian Fungsional

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

17 (1) Fungsi login

18 Pengujian fungsi ini dilakukan untuk memastikan perangkat lunak terhubung ke se-
19 rver google untuk melakukan otentikasi dan otorisasi serta memeriksa apakah email
20 yang digunakan untuk login diakhiri "@unpar.ac.id" atau "@student.unpar.ac.id" di-
21 karenakan penulis tidak mempunyai akun dosen. Contoh kasus adalah melakukan lo-
22 gin sebanyak dua kali, yang pertama menggunakan email yang diakhiri "@unpar.ac.id"
23 atau "@student.unpar.ac.id", dan yang kedua menggunakan email yang diakhiri selain
24 "@unpar.ac.id" dan "@student.unpar.ac.id". Pengujian pertama pengguna membuka
25 halaman index.php dapat dilihat pada Gambar 5.1. Lalu pengguna melakukan login
26 menggunakan email "7310013@sudent.unpar.ac.id" dapat dilihat pada Gambar 5.2. La-
27 lu 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 "bletask@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 mahasiswa yang telah dipilih oleh pengguna, dapat dilihat info dari mahasiswa tersebut. Contoh pengujian fungsi merupakan lanjutan dari fungsi memilih mahasiswa, dimana setelah pengguna memilih mahasiswa pada list.php maka akan menampilkan info 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 info mahasiswa dapat diedit. Contoh pengujian mengambil info dari mahasiswa yang telah dilihat infonya pada fungsi melihat info mahasiswa. Dimana keterangan info mahasiswa yang ada akan ditampilkan dan pengguna dapat mengedit lalu menyimpan perubahan dengan menekan tombol "Simpan" dapat dilihat pada Gambar 5.10. Setelah menyimpan perubahan pengguna

1 akan dibawa kembali ke halaman list.php. Hal ini menunjukkan fungsi mengedit info
2 mahasiswa sudah berjalan dengan baik.

3 (5) Fungsi melihat histori

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

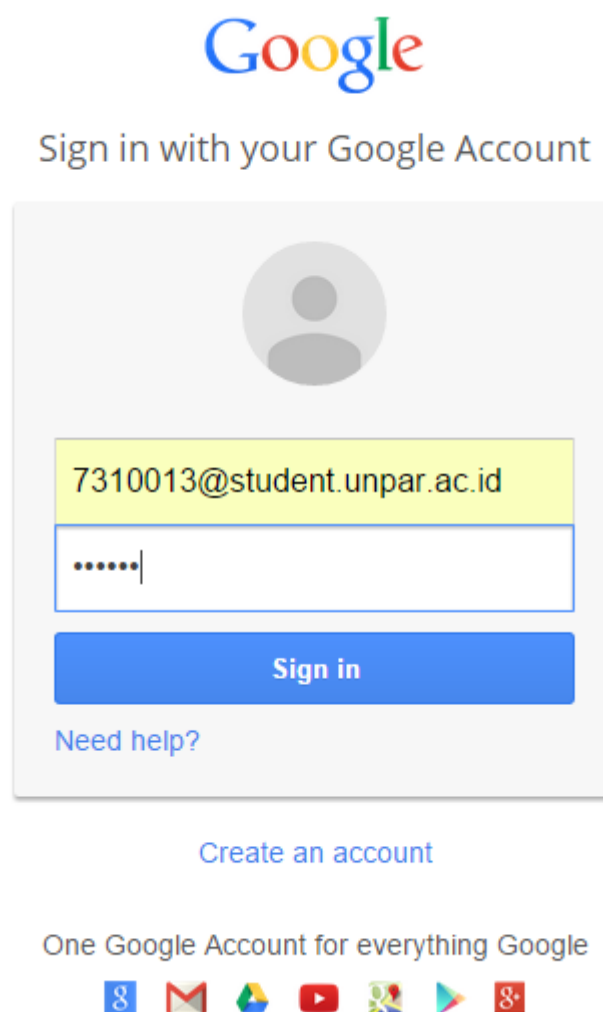
10 (6) Fungsi membuat entri baru

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

WELCOME to SIRM

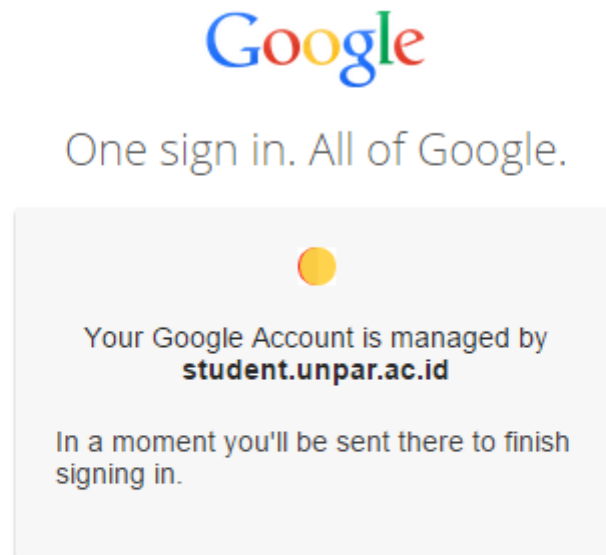
Login with Google

Gambar 5.1: Membuka Halaman index.php




The image shows a Google sign-in interface. At the top is the Google logo. Below it is the text "Sign in with your Google Account". The main part of the interface is a light gray box containing a circular profile picture placeholder. Below the placeholder are two input fields: the first contains the email address "7310013@student.unpar.ac.id" and the second contains masked characters ".....". Below these fields is a blue "Sign in" button. Under the button is a link "Need help?". Below the gray box is a link "Create an account". At the bottom, there is a line of text "One Google Account for everything Google" followed by icons for Google services: Search, Gmail, Drive, YouTube, Maps, and Google Assistant.

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



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



UNPAR Central Authentication Service

Enter your Username and Password

Username:


7310013@student.unpar.ac.id

Password:

•••••

LOGIN


 | [clear](#)





digicert
Trusted
SECURE
SSL Certificate


Powered by
Jasig Central Authentication Service


Gambar 5.4: CAS UNPAR

7310013@student.unpar.ac.id ▼



▼ SIRM would like to:

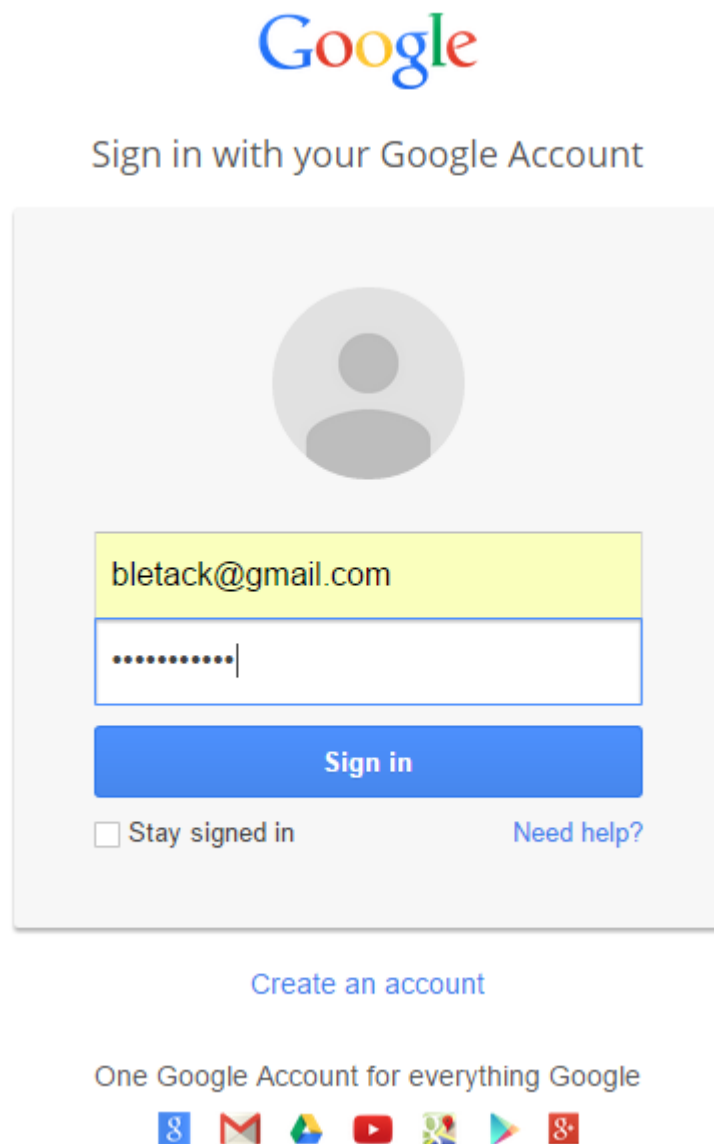
 Have offline access



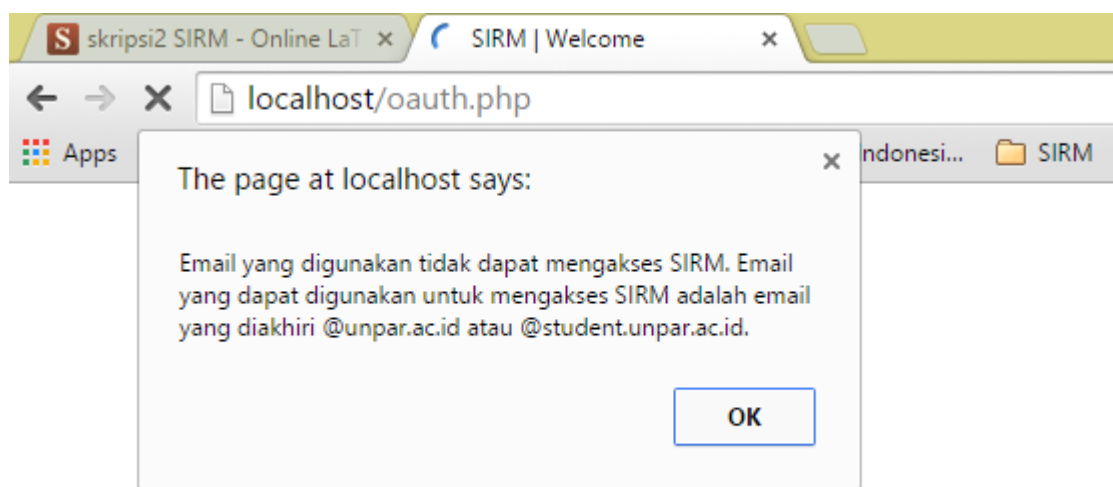
Cancel

Accept

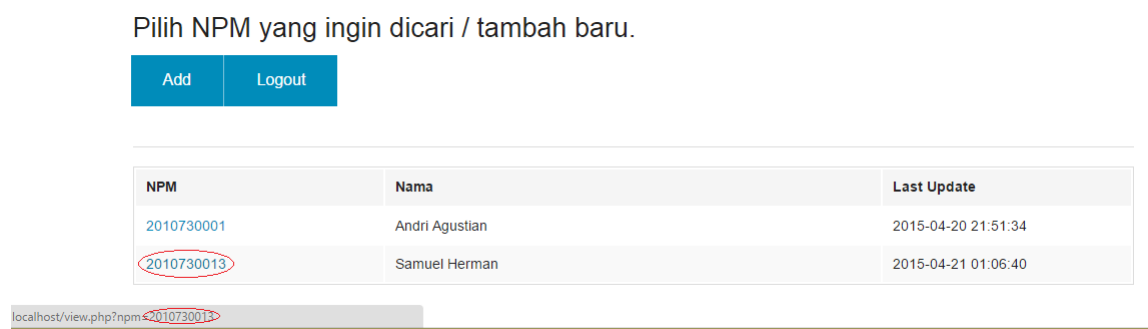
Gambar 5.5: Izin Akses Dari Pihak Pengguna



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



Gambar 5.7: Alert Email yang Digunakan Tidak Dapat Mengakses SIRM



Gambar 5.8: Memilih Mahasiswa

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



Umum

Samuel sedang mengambil tugas akhir

Catatan

- 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



Umum

Samuel sedang mengambil tugas akhir

Catatan

* 21 April 2015, pertama kali dibuat

Gambar 5.10: Mengedit Info Mahasiswa



Gambar 5.11: Melihat Histori



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

Masukan NPM

Nama

Masukan nama

Umum

Isilah deskripsi umum mahasiswa disini.

Catatan

* 9 Oktober 2014, pertama kali dibuat

Gambar 5.14: Template Entri Baru

Tabel 5.1: Hasil Pengujian Fungsional

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

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



NPM

Nama

Umum

Isilah deskripsi umum mahasiswa disini.

Catatan

* 9 Oktober 2014, pertama kali dibuat

Gambar 5.15: Membuat Entri Baru

Pilih NPM yang ingin dicari / tambah baru.



NPM	Nama	Last Update
2010730001	Andri Agustian	2015-04-20 21:51:34
2010730012	Kevin PL	2015-04-24 00:55:57
2010730013	Samuel Herman	2015-04-24 00:23:34

Gambar 5.16: Entri Baru Berhasil Dibuat

1 5.2.4 Hasil Pengujian Fungsional

- Hasil pengujian fungsional dapat dilihat pada Tabel 5.1.

1 5.2.5 Pengujian Eksperimental

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

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

1 Login

$$105 + 52 + 93 + 127/4 = 377/4 = 94.25detik$$

2 Melihat Daftar Mahasiswa

$$31 + 8 + 11 + 7/4 = 57/4 = 14.25detik$$

3 Melihat Info Mahasiswa

$$24 + 12 + 6 + 19/4 = 61/4 = 15.25detik$$

4 Mengedit Info Mahasiswa

$$37 + 43 + 39 + 25/4 = 144/4 = 36detik$$

Tabel 5.2: Catatan Waktu Pengujian Eksperimental Kelompok Pertama

Aksi	Penguji 1	Penguji 2	Penguji 3	Penguji 4
Login	1 menit 45 detik	52 detik	1 menit 33 detik	2 menit 7 detik
Melihat Daftar Mahasiswa	31 detik	8 detik	11 detik	7 detik
Melihat Info 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

5 Melihat Histori Mahasiswa

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

6 Membuat Entri Baru

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

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

1 Login

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

2 Melihat Daftar Mahasiswa

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

Tabel 5.3: Catatan Waktu Pengujian Eksperimental Kelompok kedua

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

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

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

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

Tabel 5.4: Perbandingan Rata-rata Waktu Pengujian Eksperimental

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

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.