

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	viii
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]	19
2.3 StrapdownJS [5]	21
2.4 Zurb Foundation [6]	23
2.4.1 Kompatibilitas	23
2.4.2 Apa Saja yang Hadir Dengan Foundation?	23
3 ANALISIS	29
3.1 Analisis Google Authentication	29
3.1.1 Langkah Dasar Penggunaan OAuth 2.0	29
3.1.2 Skenario Aplikasi	38
3.2 Analisis Markdown	39
3.3 Analisis StrapdownJS	42
3.4 Analisis Zurb Foundation	43
3.5 Analisis Berorientasi Objek	45
3.5.1 Use Case Diagram	46
3.5.2 Skenario	47
4 PERANCANGAN	51
4.1 Perancangan Tampilan Web Yang Digunakan	51
4.1.1 Tampilan Web Pilih Mahasiswa	51

4.1.2	Tampilan Web Info Mahasiswa	52
4.1.3	Tampilan Web Edit Mahasiswa	52
4.1.4	Tampilan Web Lihat Histori	53
4.1.5	Tampilan Web Entri Baru	53
4.2	Perancangan Modul	54
4.2.1	Modul Login	54
4.2.2	Modul Pilih Mahasiswa	55
4.2.3	Modul Info Mahasiswa	55
4.2.4	Modul Edit Mahasiswa	55
4.2.5	Modul Lihat Histori	55
4.2.6	Modul Entri Baru	56
4.3	Diagram Sekuens	56

DAFTAR REFERENSI	59
-------------------------	-----------

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 Contoh Grid	24
2.7	Gambar Contoh Tombol	25
2.8	Gambar Contoh Navigasi	26
2.9	Gambar Contoh Plugins	27
3.1	Google Developers Console	32
3.2	Membuat Proyek Baru	33
3.3	Menu Credentials	33
3.4	Membuat Client ID yang Baru	34
3.5	Tipe Aplikasi	35
3.6	Pengisian Tipe Aplikasi	36
3.7	Client ID	37
3.8	Izin Pihak Pengguna	38
3.9	Skenario Aplikasi SIRM	39
3.10	Output Sintaks Cetak Tebal dan Cetak Miring	40
3.11	Output Sintaks Judul Bab	41
3.12	Output Sintaks Batas Baris	41
3.13	Output Sintaks Paragraf	41
3.14	Output Sintaks Link	41
3.15	Output Sintaks Daftar	41
3.16	Output Keterangan Mahasiswa	42
3.17	Tampilan pilihmahasiswa.php Dengan Zurb Foundation	45
3.18	Use Case Diagram	46
4.1	Desain Antarmuka Pilih Mahasiswa	51
4.2	Desain Antarmuka Info Mahasiswa	52
4.3	Desain Antarmuka Edit Mahasiswa	52
4.4	Desain Antarmuka Lihat Histori	53
4.5	Desain Antarmuka Entri Baru	54
4.6	Diagram Sekuens	57

DAFTAR TABEL

2.1	Tabel 2-1 Daftar Pengujian Zurb Foundation	23
3.1	Tabel 3-1 Skenario Login	47
3.2	Tabel 3-2 Skenario Pilih Mahasiswa	48
3.3	Tabel 3-3 Skenario Melihat Info Mahasiswa	48
3.4	Tabel 3-4 Skenario Edit Mahasiswa	49
3.5	Tabel 3-5 Skenario Lihat Histori	49
3.6	Tabel 3-6 Skenario Membuat Entri Baru	50
4.1	Tabel 4-1 Modul Login	55
4.2	Tabel 4-2 Modul Pilih Mahasiswa	55
4.3	Tabel 4-3 Modul Info Mahasiswa	55
4.4	Tabel 4-4 Modul Edit Mahasiswa	56
4.5	Tabel 4-5 Modul Lihat Histori	56
4.6	Tabel 4-6 Modul Entri Baru	56

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 diedit dan pengguna juga bisa menekan tombol "Add" untuk menambah maha-
3 siswa 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 web server 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 diinstal pada perangkat tidak memiliki fitur mesin pencari maka akan menggunakan *web service*. Beberapa permintaan memerlukan tahap otentikasi dimana pengguna diharuskan login menggunakan akun Google mereka. Setelah login pengguna akan ditanya apakah pengguna akan memberi izin untuk aplikasi yang telah melakukan permintaan tersebut. Proses ini disebut izin dari pihak pengguna. Jika pengguna memberi izin, maka Google Authorization Server akan mengirimkan aplikasi tersebut sebuah token akses. Jika pengguna tidak memberi izin, maka server akan menunjukkan respon yang menyatakan eror.

3. Kirim token akses ke API

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

4. Memperbaharui token akses jika diperlukan

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

2.1.2 Skenario Google Authentication

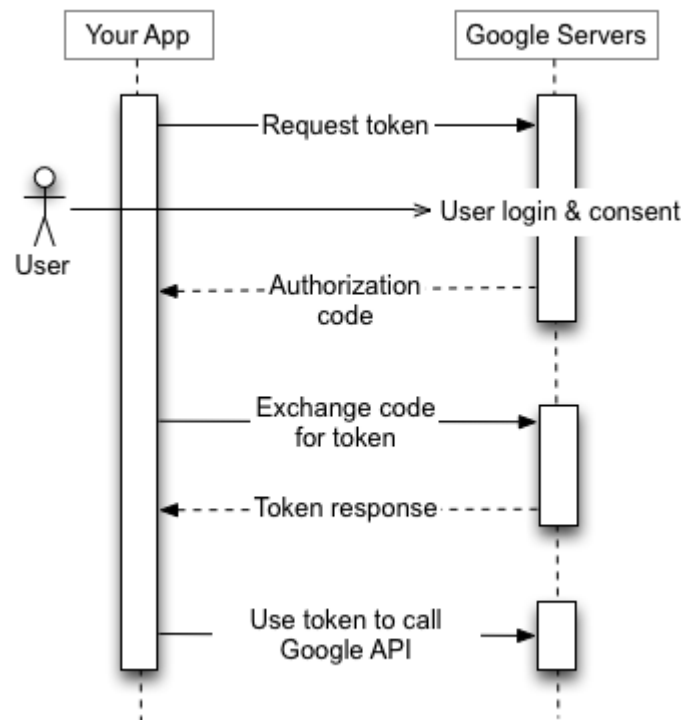
Terdapat lima skenario yang dapat digunakan untuk Google Authentication yaitu Skenario Aplikasi Web Server, Skenario Aplikasi yang Terinstal, Skenario Aplikasi Sisi Klien (JavaScript), Skenario Aplikasi Pada Perangkat Dengan Masukan Yang Terbatas, dan Skenario Layanan Akun. Untuk penjelasan lebih lanjut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

2.1.2.1 Skenario Aplikasi Web Server

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

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

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



Gambar 2.1: Skenario Aplikasi Web Server

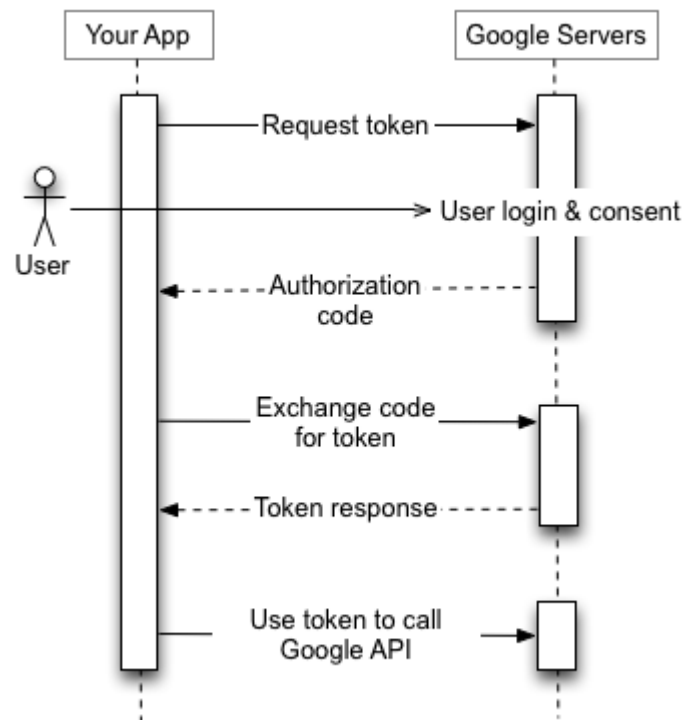
1 2.1.2.2 Skenario Aplikasi yang Terinstal

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.

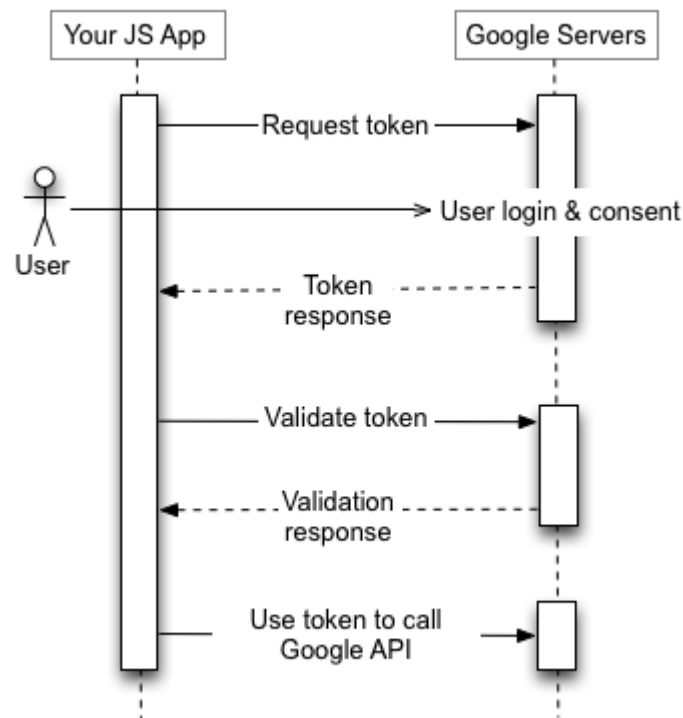
13 Aplikasi harus menyimpan token yang diperbaharui untuk penggunaan masa depan dan
 14 menggunakan token akses untuk mengakses API Google. Setelah masa token akses berakhir,
 15 maka aplikasi akan memperbaharui token untuk mendapatkan yang baru. Untuk gambar
 16 skenario dapat dilihat pada Gambar 2.2.



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.

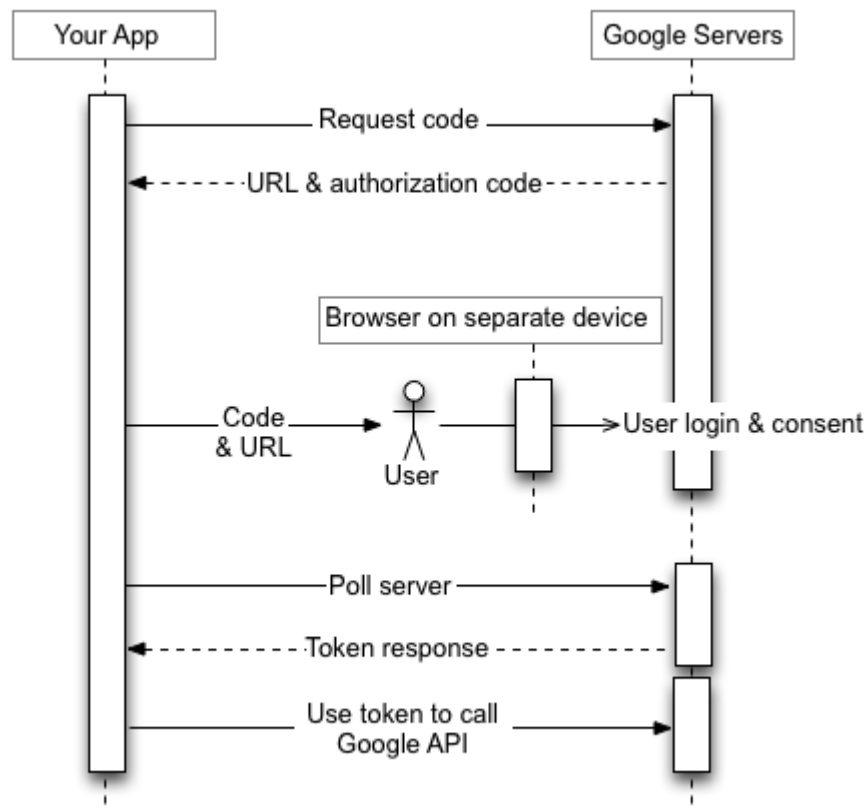


Gambar 2.3: Skenario Aplikasi Sisi Klien (JavaScript)

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 ditandatangani

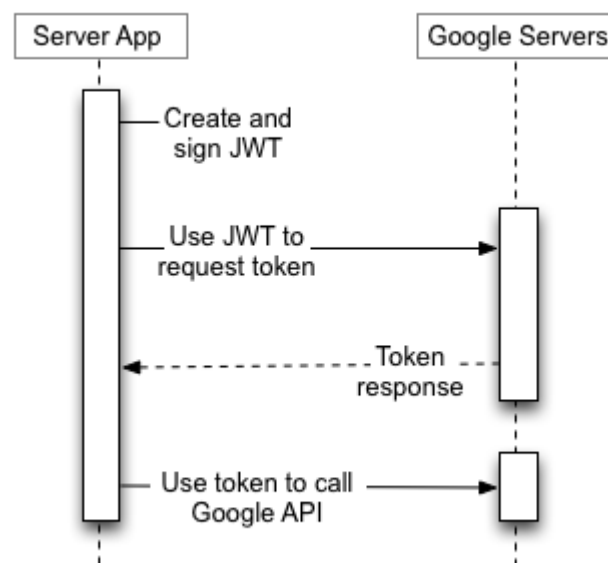
5 ni dan membangun permintaan token akses dalam format yang sesuai. Aplikasi kemudian

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

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 mengembailakan 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.

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

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.

```
# Judul Bab (an <h1> tag)
## Judul Sub Bab (an <h2> tag)
### Judul Sub Sub Bab (an <h3> tag)
#### Tingkat ke 4 (an <h4> tag)
##### Tingkat ke 5 (an <h5> tag)
```


1 | ##### Tingkat ke 6 (an <h6> tag)

2 | 2.2.2.3 Batas Baris

3 | Untuk menyisipkan satu baris baru dalam dokumen, mengakhiri baris dengan dua atau lebih
4 | spasi lalu tekan 'Enter'.

5 | Baris ini dengan
6 | batas baris

7 | Contoh di atas, setelah kata 'dengan' diakhiri dengan tiga spasi lalu tekan 'Enter'.

8 | Baris ini tanpa
9 | batas baris

10 | Contoh di atas, setelah kata 'tanpa' diakhiri tanpa spasi langsung tekan 'Enter'.

11 | 2.2.2.4 Paragraf

12 | Untuk menyisipkan paragraf baru, cukup menyisipkan satu baris kosong.

13 | Ini kalimat pertama. Ini kalimat berikutnya. Ini kalimat terakhir.
14 |
15 | Ini paragraf baru.

16 | Contoh diatas baris pertama adalah paragraf kesatu. Setelah itu ada satu baris kosong.
17 | Kalimat kedua merupakan paragraf kedua.

18 | 2.2.2.5 Gambar

19 | Untuk menyisipkan gambar pada dokumen Markdown, gunakan sintaks berikut

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

21 | Contoh penggunaan sintaks gambar:

22 | ![logo](https://www.google.com/images/srpr/logollw.png "Google")

23 | Url gambar dapat diganti dengan path yang mengarah pada file gambar yang akan digu-
24 | nakan.

25 | 2.2.2.6 Link

26 | Untuk menyisipkan hyperlink pada dokumen Markdown, gunakan sintaks berikut

27 | [link_teks](/ tujuan_url " judul opsional ")

28 | Contoh penggunaan sintaks link:

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

2.2.2.7 Kode

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

```

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.
```

'initLabels()' akan menjadi kode pada satu baris kalimat. Sedangkan fungsi yang ada didalam `'''` akan menjadi blok kode.

2.2.2.8 Kutipan

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

```

23 | Ini merupakan kutipan **favorit** saya:
24 |
25 | > The weak can never forgive.
26 | > Forgiveness is the attribute of the strong.
```

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.

```

30 | # Bagian Satu
31 |
32 | The quick brown fox jumps over the lazy dog. The quick brpwn fox jump over the lazy dog.
33 |
34 | ---
35 |
36 | Last Edited on *25th Dec 2014*
```

Pada contoh di atas `---` akan menjadi 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
```

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
```

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 garis bawah akan dirubah menjadi cetak miring, namun pada GFM garis bawah 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 (*).

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

1 yang mengarah ke URL tersebut.

2 | `http://example.com`

3 Link di atas kan menjadi taut secara otomatis ke URL tersebut.

4 **2.2.3.3 Tanda Coret**

5 GFM menambahkan sintaks untuk membuat teks dicoret, yang dihilangkan dari SM.

6 Untuk membuat teks dicoret gunakan dua karakter tilde (~) antara kata yang akan
7 dicoret.

8 | `~~Mistaken text.~~`

9 **2.2.3.4 Blok Kode**

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

14 | `Here's an example:`
15 |
16 | `'''`
17 | `function test() {`
18 | `console.log("notice the blank line before this function?");`
19 | `}`
20 | `'''`

21 **2.2.3.5 Penandaan Sintaks**

22 Blok kode dapat dilanjutkan dengan menambah sintaks. Dalam blok yang ditandai
23 tambahkan sebuah indentifikasi bahasa apa yang digunakan. Misalnya penandaan code
24 sintaks Ruby.

25 | `'''ruby`
26 | `require 'redcarpet'`
27 | `markdown = Redcarpet.new("Hello World!")`
28 | `puts markdown.to_html`
29 | `'''`

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

33 | `First Header | Second Header`
34 | `-----|-----`
35 | `Content Cell | Content Cell`
36 | `Content Cell | Content Cell`

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

```

1 | First Header | Second Header |
2 | ----- | ----- |
3 | Content Cell | Content Cell |
4 | Content Cell | Content Cell |

```

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

```

9 | Name | Description |
10 | ----- | ----- |
11 | Help | Display the help window. |
12 | Close | Closes a window |

```

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

```

16 | Name | Description |
17 | ----- | ----- |
18 | Help | ~~Display the~~ help window. |
19 | Close | _Closes_ a window |

```

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

```

22 | Left-Aligned | Center Aligned | Right Aligned |
23 | :-----: | :-----: | -----: |
24 | col 3 is | some wordy text | $1600 |
25 | col 2 is | centered | $12 |
26 | zebra stripes | are neat | $1 |

```

2.3 StrapdownJS [5]

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

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

- Cara pertama:

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

2 • Cara kedua:

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

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

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

```
7 | <xmp theme="united" style="display:none;">
8 |
9 | </xmp>
```

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

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

```
13 | <!DOCTYPE html>
14 | <html>
15 | <title>Hello Strapdown</title>
16 |
17 | <xmp theme="united" style="display:none;">
18 | # Markdown text goes in here
19 |
20 | ## Chapter 1
21 |
22 | Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt
23 |     ut labore
24 |     et dolore magna aliqua.
25 |
26 | ## Chapter 2
27 |
28 | Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut
29 | aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit
30 |     esse
31 | cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt
32 |     in
33 | culpa qui officia deserunt mollit anim id est laborum.
34 | </xmp>
35 |
36 | <script src="http://strapdownjs.com/v/0.2/strapdown.js"></script>
37 | </html>
```

38 Strapdown.js juga memiliki beberapa fitur :

- 39 (a) Ramah dengan mesin pencari
- 40 (b) Kompatibel dengan berbagai browser (Sudah diuji dengan ponsel menggunakan
41 Safari, IE 8/9, Firefox, Chrome)
- 42 (c) Github menggunakan Markdown (Tabel, Syntax, Headline)
- 43 (d) 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 Hadir Dengan Foundation?

Foundation memiliki banyak komponen dan struktur untuk membantu membangun sebuah situs responsif. Semua komponen Foundation dapat dilihat pada satu halaman yang disebut Kitchen Sink² atau melihat beberapa gambar dibawah ini :

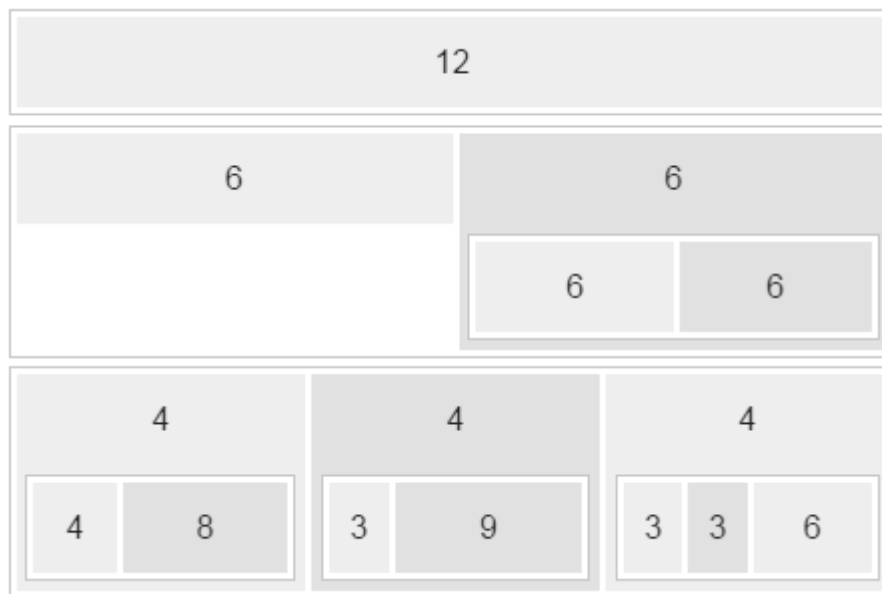
(a) The Grid

Grid bekerja pada hampir semua perangkat dan memiliki dukungan untuk men-

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

²http://foundation.zurb.com/docs/components/kitchen_sink.html.js

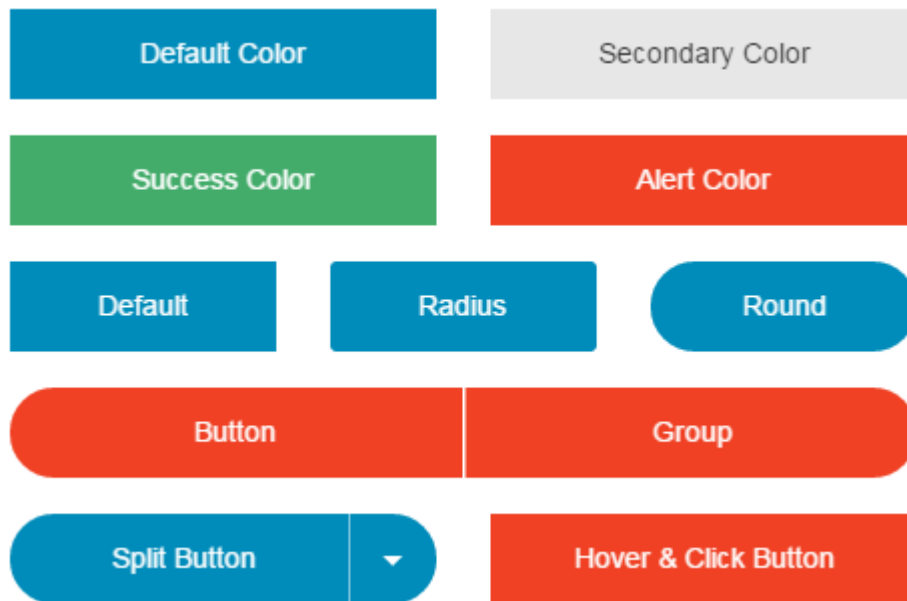
jadi satu kesatuan, sumber pemesanan, offset dan perangkat presentasi. Hal tersebut sedikit terlalu mudah, dalam waktu singkat, dapat menciptakan tata letak yang kompleks seperti ini. Untuk contoh grid dapat dilihat pada Gambar 2.6.



Gambar 2.6: Contoh Grid

(b) Tombol

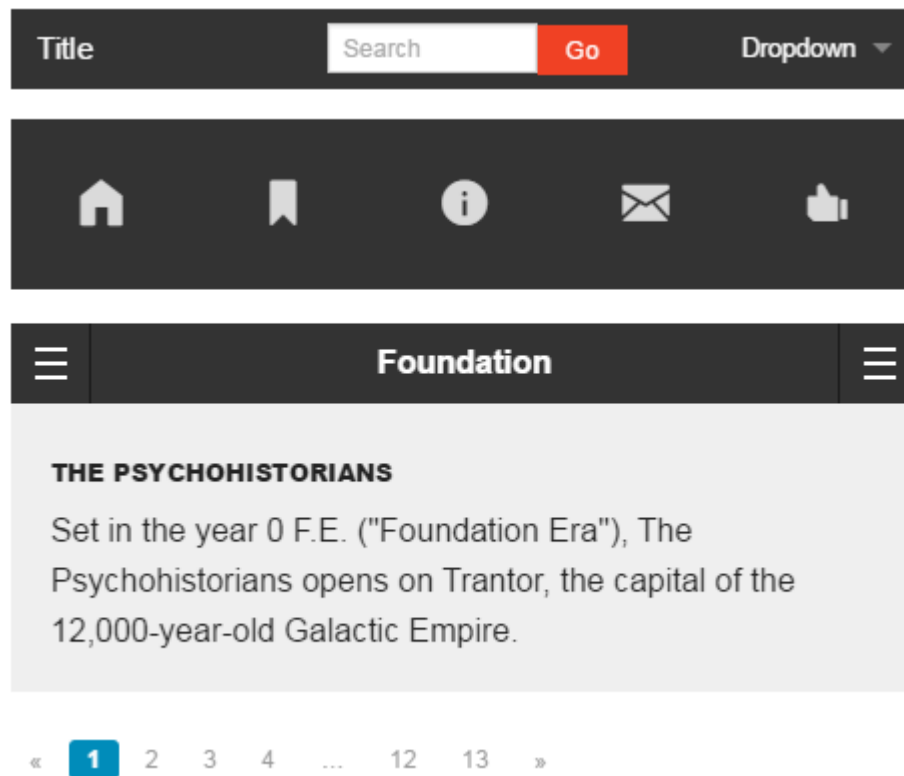
Mengklik tombol dengan material yang bagus merupakan hal yang mengagumkan. Mengklik tombol juga menghubungkan pengguna dengan berbagai aksi. Ada beberapa gaya tombol yang ringan untuk ukuran, presentasi, dan warna untuk menyesuaikan tombol Anda sendiri semudah menambahkan kelas. Untuk contoh macam-macam tombol dapat dilihat pada Gambar 2.7.



Gambar 2.7: Contoh Tombol

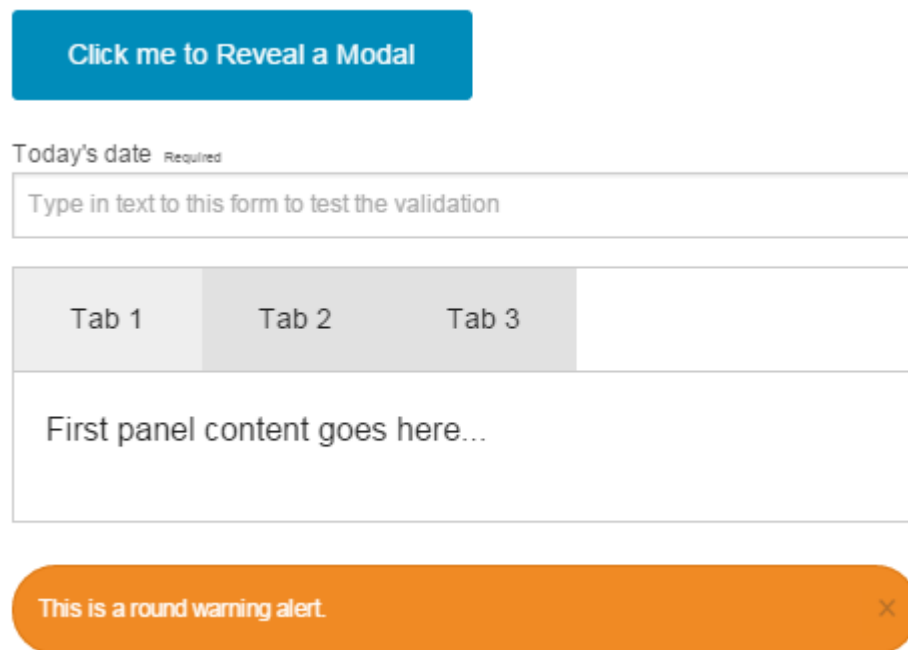
1 (c) Navigasi

2 Orang yang mengakses harus bisa berkeliling melihat menu-menu yang ada. Gaya
3 navigasi pada Foundation meliputi : bar bagian atas yang kuat dengan menu
4 dropdown; tombol; bar pencari; ikon bar yang keren; implementasi kanvas yang
5 lepas dari keluhan; dan sekelompok navigasi lainnya. Untuk contoh macam-
6 macam navigasi dapat dilihat pada Gambar 2.8.



Gambar 2.8: Contoh Navigasi

- 1 (d) Plugins
- 2 Sudah meliputi banyak plugin javascript yang ditulis untuk modal dasar pop-
- 3 up; menambatkan formulir validasi yang diperlukan; membuat tab konten; tanda
- 4 peringatan; dan masih banyak lagi. Untuk contoh macam-macam plugin dapat
- 5 dilihat pada Gambar 2.9.



Gambar 2.9: 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 Berorientasi Objek berisi use case diagram dan skenario 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:

(a) Mendapatkan kepercayaan OAuth 2.0 dari Google Developers Console

- i. Mengunjungi Google Developers Console. Agar lebih jelas dapat dilihat pada Gambar 3.1.

- ii. Buat sebuah proyek baru. Dapat dilihat pada Gambar 3.2.
- iii. Masuk ke proyek yang telah dibuat dan masuk ke menu 'Credentials'. Agar lebih jelas dapat dilihat pada Gambar 3.3.
- iv. Membuat client ID yang baru. Dapat dilihat pada Gambar 3.4.
- v. 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.
- vi. 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.
- vii. 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-ufmbistn0pcaq0khubafola133orfgve.apps.googleusercontent.com
Client secret :
-cSZ-AUmeQ9PaWWry_IpiBBi
```

Agar lebih jelas dapat dilihat pada Gambar 3.7.

(b) 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.

(c) Kirim token ke API

Setelah mendapatkan token akses, maka untuk mengirimkan ke API diperlukan

1 scope. Karena sesuai dengan landasan teori, jika token akses dikeluarkan untuk
2 Google+ API maka token akses tersebut tidak berlaku untuk mengakses Google
3 Contact API. Scope yang akan digunakan pada penelitian ini adalah:

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

7 karena pada penelitian ini dibutuhkan email pengguna dan nama pengguna, ke-
8 duanya itu dapat diperoleh dengan menggunakan kedua scope tersebut. Scope
9 pertama untuk mendapatkan info dari pengguna dan scope kedua untuk menda-
10 patkan email dari pengguna.

11 (d) Memperbaharui token akses jika diperlukan

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

Google Developers Console

Sign up for a free trial.

+Samuel

Create Project

PROJECT NAME	PROJECT ID	REQUESTS	ERRORS	CHARGES
My Project	angelic-booster-782	0	0	—
SIRM	sirm-123	0	1	—

Projects

Billing

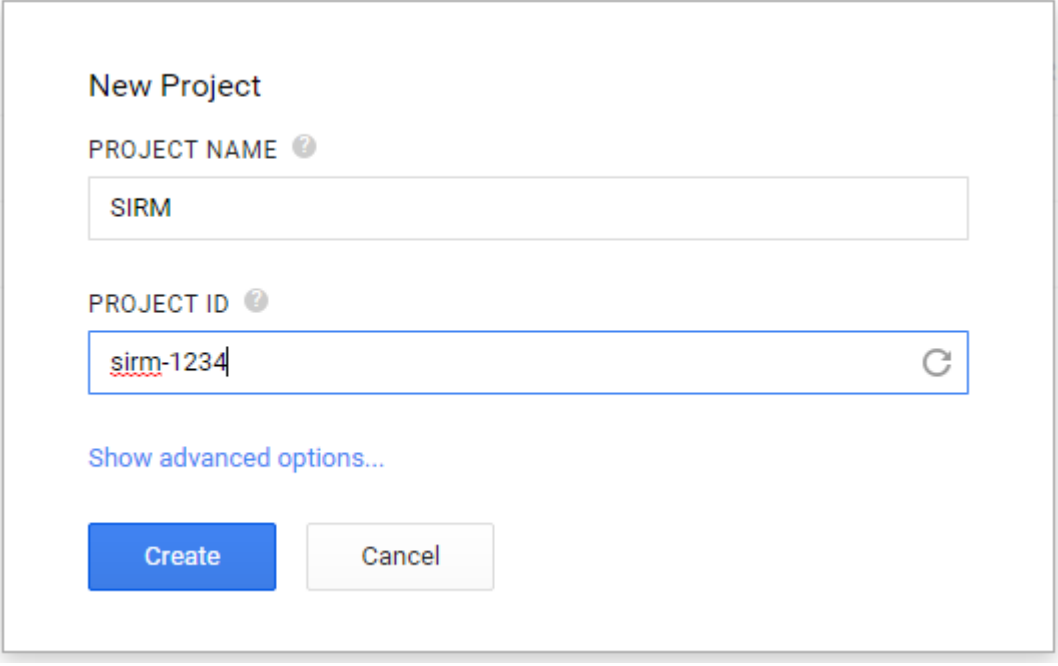
Account settings

Need help?

Terms of service

Privacy

Gambar 3.1: Google Developers Console




New Project

PROJECT NAME ?

SIRM

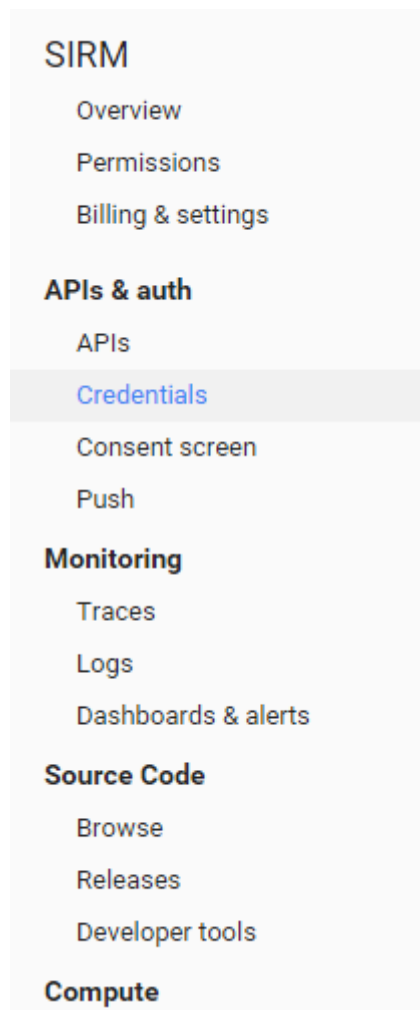
PROJECT ID ?

sirm-1234 

[Show advanced options...](#)

Create **Cancel**

Gambar 3.2: Membuat Proyek Baru



Gambar 3.3: Menu Credentials

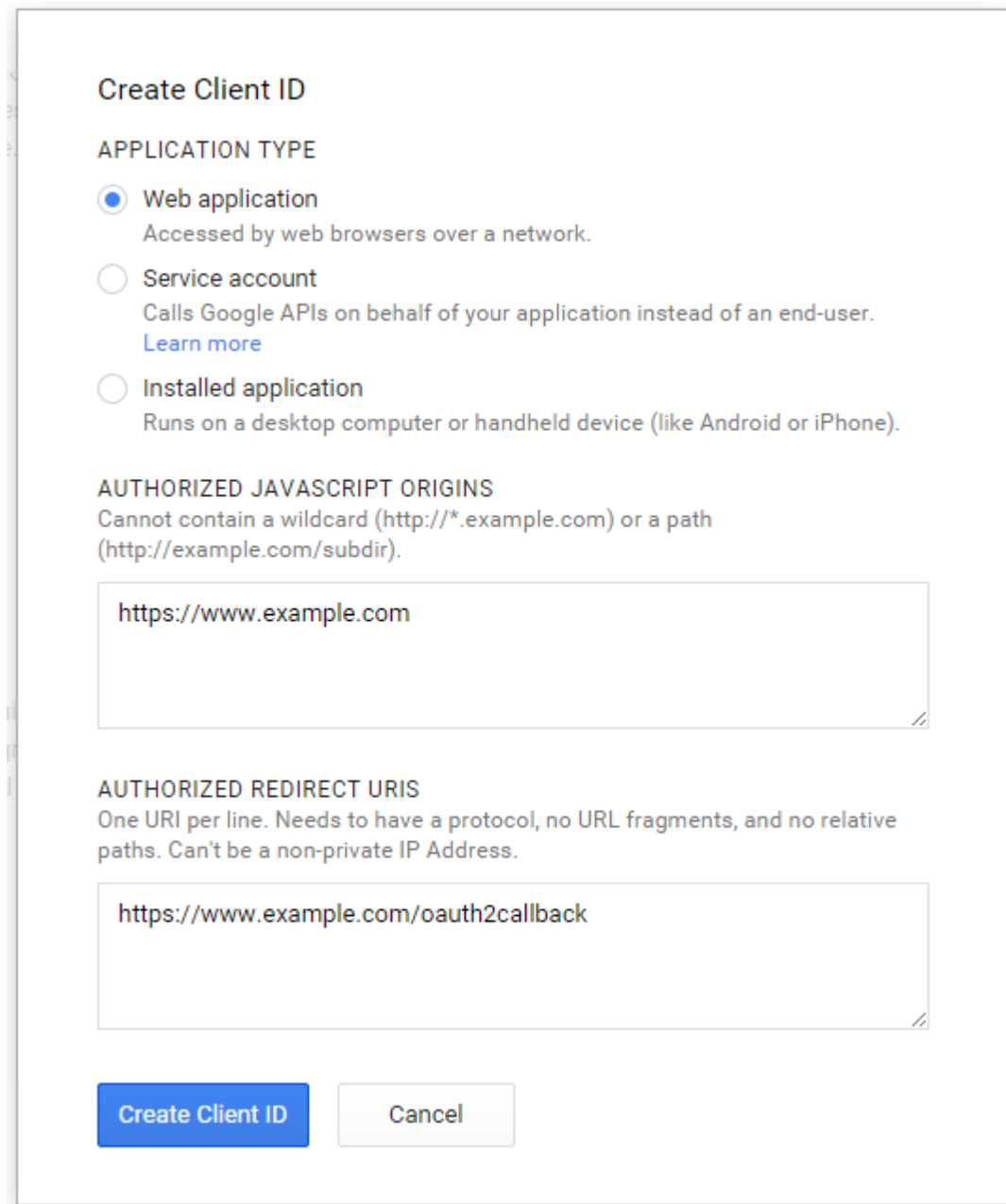
OAuth

OAuth 2.0 allows users to share specific data with you (for example, contact lists) while keeping their usernames, passwords, and other information private.

[Learn more](#)

Create new Client ID

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).

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.

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/

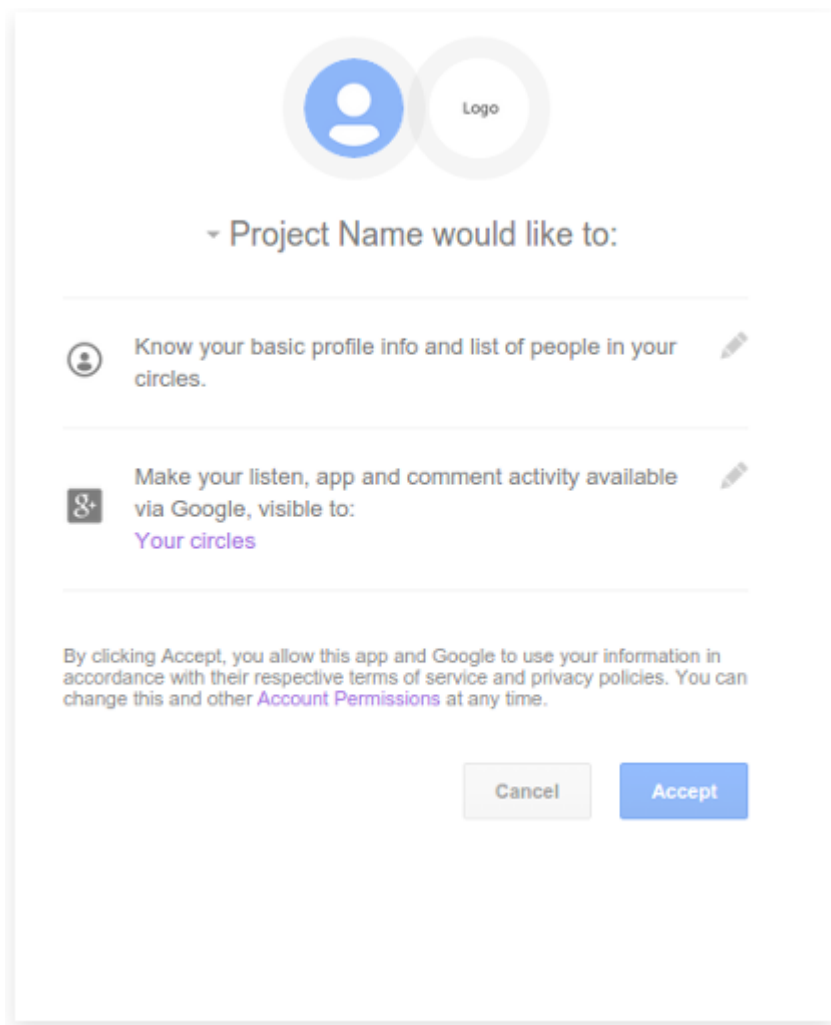
Edit settings

Reset secret

Download JSON

Delete

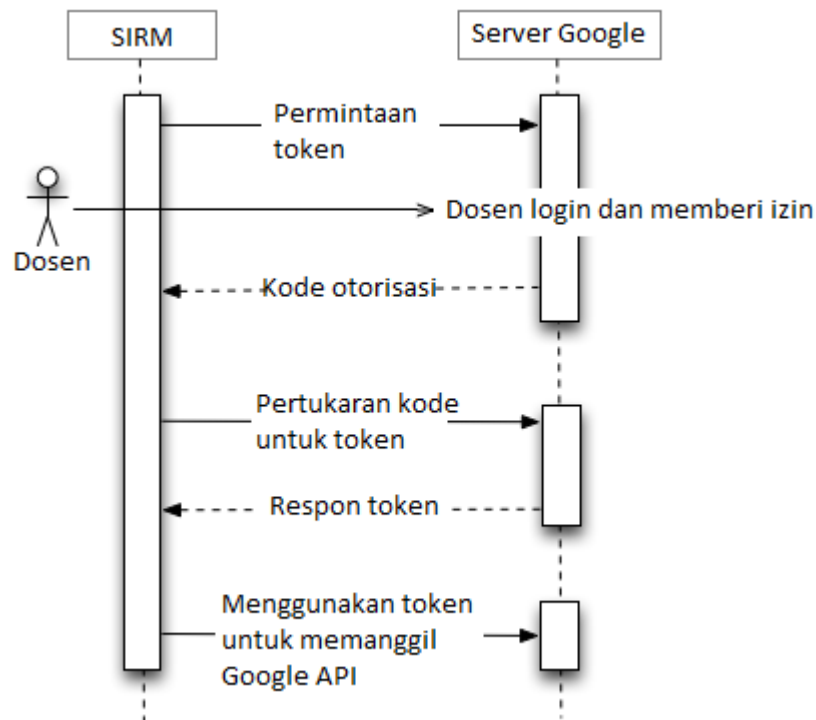
Gambar 3.7: Client ID



Gambar 3.8: Izin Pihak Pengguna

3.1.2 Skenario Aplikasi

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



Gambar 3.9: Skenario Aplikasi SIRM

3.2 Analisis Markdown

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

- Sintask Cetak Tebal dan Cetak Miring

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

```

| **NPM** - *2010730013*

```

- Sintaks Judul Bab

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

```

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
20     |      di himpunan sebagai ketua divisi pelayanan masyarakat.
21
22     | Grady adalah seorang mahasiswa yang memiliki jiwa pemimpin. Dia aktif di UKM sebagai
23     |      ketua 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://bletaek.blogspot.com/](http://bletaek.
29     |      blogspot.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
- 2 mahasiswa. 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
12     himpunan sebagai ketua divisi pelayanan masyarakat. Yang bersangkutan memiliki blog di
13     [http://bletack.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

3.3 Analisis StrapdownJS

StrapdownJS digunakan untuk menampilkan sintaks Markdown ke halaman HTML.

Pada penelitian ini strapdown.js terlebih dahulu diunduh dan untuk menggunakannya menggunakan path seperti di bawah ini.

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

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

```

<!doctype html>
<html class="no-js" lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

```

```

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

```

Untuk baris 17 sampai baris 37 akan diambil dari database.

3.4 Analisis Zurb Foundation

Zurb Foundation digunakan untuk membuat tampilan antarmuka aplikasi yang akan dibangun. Sesuai landasan teori pada bab 2, pada aplikasi ini menggunakan tiga bagian yaitu Grid, Tabel dan Tombol. Grid digunakan untuk mengatur pembagian tata letak kompleks sehingga terlihat rapih. Tabel digunakan untuk menampilkan data yang berasal dari database. Tombol digunakan untuk merubah tombol yang biasa menjadi lebih enak untuk dilihat. Berikut sintaks penggunaan Grid, Tabel, dan Tombol pada pilihmahasiswa.php dan untuk gambar dapat dilihat pada Gambar 4.1.

```
<!doctype html>
```

```

1      <html class="no-js" lang="en">
2          <head>
3              <meta charset="utf-8" />
4              <meta name="viewport" content="width=device-width, initial-scale=1.0" />
5              <title>SIRM | Welcome</title>
6              <link rel="stylesheet" href="css/foundation.css" />
7              <script src="js/vendor/modernizr.js"></script>
8          </head>
9          <body>
10             <div class="row">
11                 <h5>Masukan NPM yang ingin dicari / tambah baru.</h5>
12             </div>
13             <form>
14                 <div class="row">
15                     <div class="large-12 columns">
16                         <div class="row collapse">
17                             <div class="small-8 columns">
18                                 <input type="text" placeholder="NPM">
19                             </div>
20                             <div class="small-2 columns">
21                                 <a href="#" class="button postfix">Search</a>
22                             </div>
23                             <div class="small-2 columns">
24                                 <a href="entribaru.php" class="button postfix">Add</a>
25                             </div>
26                         </div>
27                     </div>
28                 </div>
29             </form>
30             <div class="row">
31                 <?php
32                     $pemakai="admin";
33                     $pass="admin";
34                     $id_mysql=mysql_connect("localhost", $pemakai, $pass);
35
36                     if (!$id_mysql){
37                         die("Database tidak bisa dibuka");
38                     }
39
40                     if (!mysql_select_db("sirm", $id_mysql)){
41                         die("Database tidak bisa dipilih");
42                     }
43
44                     $hasil = mysql_query("SELECT * FROM info_mahasiswa", $id_mysql);
45
46                     if (!$hasil){
47                         die("Permintaan gagal");
48                     }
49
50                     echo "<table>
51 <thead>
52 <tr>
53 <th width='250'>NPM</th>
54 <th width='500'>Nama</th>
55 <th width='250'>Last Update</th>
56 </tr>
57 </thead>";
58
59                     while($row = mysql_fetch_array($hasil))
60                     {
61                         echo "<tr>";
62                         echo "<td><a href='infomahasiswa.php'>" . $row['npm'] . "</a></td>";
63                         echo "<td>" . $row['nama'] . "</td>";
64                         echo "<td>" . $row['log'] . "</td>";

```

```

1      echo "</tr>";
2      }
3      echo "</table>";
4
5      mysql_close($id_mysql);
6  ?>
7  </div>
8  </body>
9  </html>

```

1 2 3

4

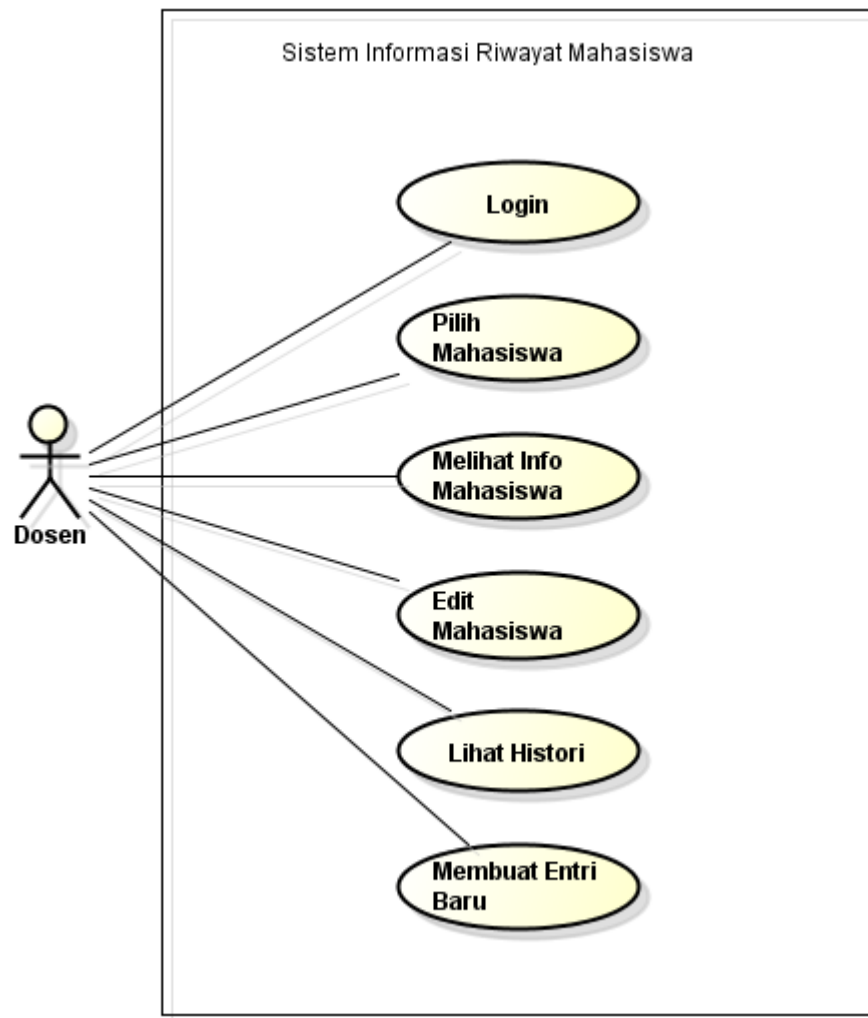
NPM	Nama	Last Update
2010730001	Andri	9 Oktober 2014
2010730012	Kevin	2 Oktober 2014
2010730013	Samuel	7 September 2014

Gambar 3.17: Tampilan pilihmahasiswa.php Dengan Zurb Foundation

3.5 Analisis Berorientasi Objek

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

3.5.1 Use Case Diagram



Gambar 3.18: Use Case Diagram

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

- Login

Use case ini memungkinkan pengguna untuk login via Google OAuth.

- Pilih Mahasiswa

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

- 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		

Use case ini memungkinkan pengguna untuk melihat info mahasiswa. Selain itu pengguna bisa menekan tombol "Edit" untuk mengedit info mahasiswa dan pengguna juga bisa menekan tombol "Lihat Histori" untuk melihat histori.

- Edit Mahasiswa

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

- Lihat Histori

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

- Membuat Entri Baru

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

3.5.2 Skenario

3.5.2.1 Login

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

3.5.2.2 Pilih Mahasiswa

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

Tabel 3.2: Skenario Pilih Mahasiswa

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

Tabel 3.3: Skenario Melihat Info Mahasiswa

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

1 3.5.2.3 Melihat Info Mahasiswa

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

3 3.5.2.4 Edit Mahasiswa

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

5 3.5.2.5 Lihat Histori

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

Tabel 3.4: Skenario Edit Mahasiswa

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

Tabel 3.5: Skenario Lihat Histori

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

Tabel 3.6: Skenario Membuat Entri Baru

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

1 3.5.2.6 Membuat Entri Baru

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

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 lima buah perancangan yaitu pilih mahasiswa, info mahasiswa, edit mahasiswa, lihat histori dan entri baru.

4.1.1 Tampilan Web Pilih Mahasiswa

Perancangan tampilan web untuk pilih mahasiswa dapat dilihat pada Gambar 4.1.

1 2010730013 2 Search 3 Add

NPM	Nama	Last Update
2010730001	Andri	9 Oktober 2014
2010730012	Kevin	2 Oktober 2014
2010730013	Samuel	7 September 2014

Gambar 4.1: Desain Antarmuka Pilih Mahasiswa

Keterangan :

- (1) Bagian ini untuk memasukan input npm yang akan dicari oleh pengguna.
- (2) Bagian ini merupakan tombol untuk melakukan aksi pencarian.
- (3) Bagian ini merupakan tombol untuk melakukan aksi tambah data.
- (4) Bagian ini merupakan tempat menampilkan data mahasiswa dalam bentuk tabel.

4.1.2 Tampilan Web Info Mahasiswa

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



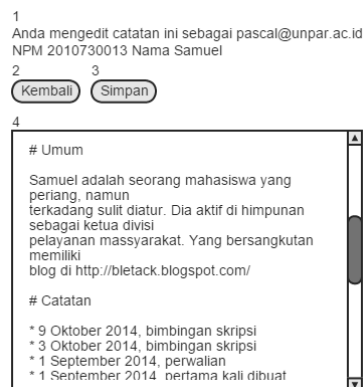
Gambar 4.2: Desain Antarmuka Info Mahasiswa

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 edit.
- (3) Bagian ini merupakan tombol untuk melakukan aksi lihat histori.
- (4) Bagian ini merupakan tempat menampilkan info mahasiswa yang berasal dari database.

4.1.3 Tampilan Web Edit Mahasiswa

Perancangan tampilan web untuk edit mahasiswa dapat dilihat pada Gambar 4.3.



Gambar 4.3: 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 dan juga menampilkan
4 NPM dan nama mahasiswa yang telah dipilih untuk diedit.

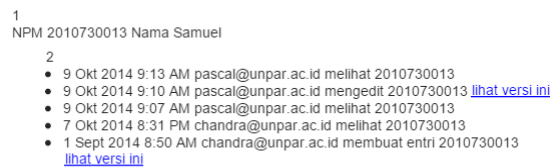
5 (2) Bagian ini merupakan tombol untuk melakukan aksi kembali ke info mahasiswa.

6 (3) Bagian ini merupakan tombol untuk melakukan aksi simpan untuk perubahan
7 yang telah dilakukan.

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

10 4.1.4 Tampilan Web Lihat Histori

11 Perancangan tampilan web untuk lihat histori dapat dilihat pada Gambar 4.4.



Gambar 4.4: Desain Antarmuka Lihat Histori

12 Keterangan :

13 (1) Bagian ini merupakan teks yang menampilkan keterangan NPM dan nama maha-
14 siswa yang telah dipilih untuk dilihat historinya.

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

16 4.1.5 Tampilan Web Entri Baru

17 Perancangan tampilan web untuk entri baru dapat dilihat pada Gambar 4.5.

1
Anda membuat catatan mahasiswa ini sebagai pascal@unpar.ac.id

2 3
Kembali Simpan

4 NPM 2010730013

5 Nama Samuel

6

Umum

Isilah deskripsi umum mahasiswa disini

Catatan

* 9 Oktober 2014, pertama kali dibuat

Gambar 4.5: 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.
- (3) Bagian ini merupakan tombol untuk melakukan aksi simpan untuk penambahan entri baru.
- (4) Bagian ini merupakan area untuk memasukkan NPM mahasiswa yang akan ditambah.
- (5) Bagian ini merupakan area untuk memasukkan nama mahasiswa yang akan ditambah.
- (6) Bagian ini merupakan area untuk memasukkan 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.

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.

4.2.2 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.

4.2.5 Modul Lihat Histori

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

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.

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.

4.2.6 Modul Entri Baru

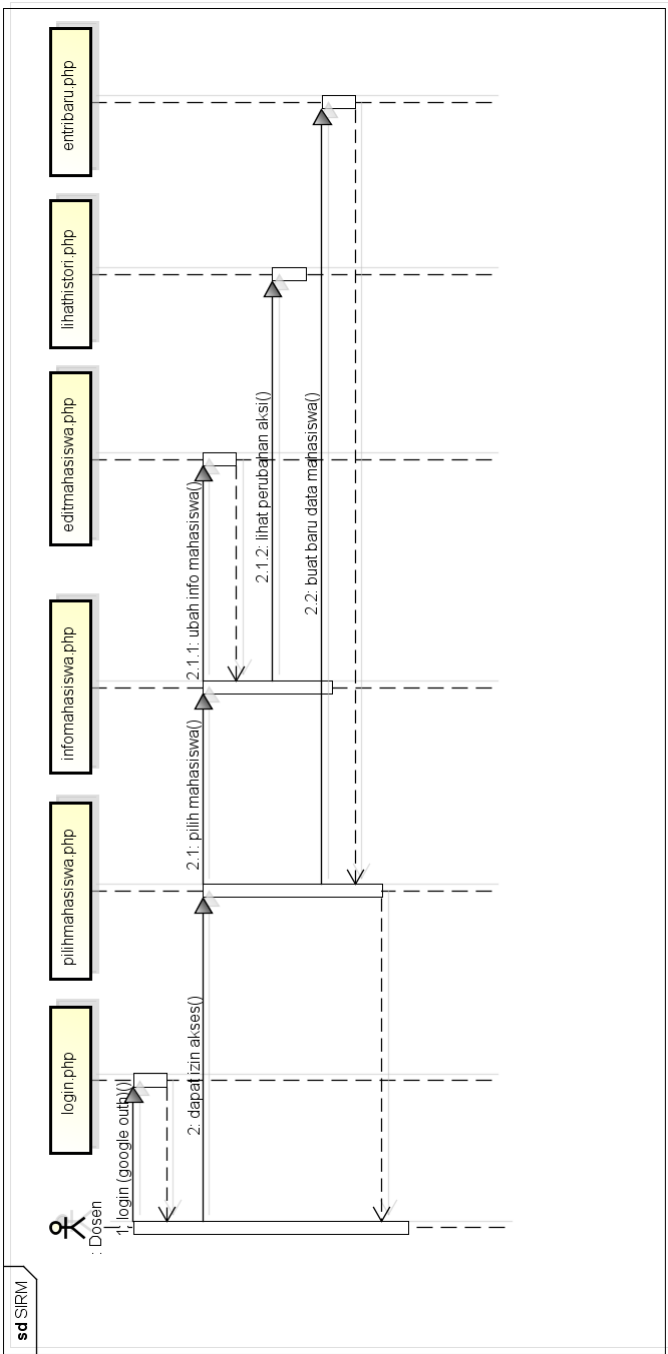
Modul entri baru yang dilakukan oleh user (dosen) dapat dilihat pada Tabel 4.6.

4.3 Diagram Sekuens

Pembuatan diagram sekuens mengacu pada Gambar 3.18. Diagram sekuens dapat dilihat pada Gambar 4.6.

Tabel 4.6: Modul Entri Baru

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



powered by Astah

Gambar 4.6: Diagram Sekuens

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.