

**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

<b>DAFTAR ISI</b>	<b>v</b>
<b>DAFTAR GAMBAR</b>	<b>vii</b>
<b>DAFTAR TABEL</b>	<b>viii</b>
<b>1 PENDAHULUAN</b>	<b>1</b>
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
<b>2 LANDASAN TEORI</b>	<b>5</b>
2.1 Google Authentication [1] . . . . .	5
2.1.1 Langkah Dasar . . . . .	5
2.1.2 Skenario Aplikasi Web Server . . . . .	7
2.1.3 Skenario Aplikasi yang Terinstal . . . . .	8
2.1.4 Skenario Aplikasi Sisi Klien (JavaScript) . . . . .	9
2.1.5 Skenario Aplikasi Pada Perangkat Dengan Masukan Yang Terbatas . . . . .	9
2.1.6 Skenario Layanan Akun . . . . .	10
2.1.7 Masa Habis Berlaku Token . . . . .	11
2.2 Markdown . . . . .	12
2.2.1 Apa itu Markdown? [2] . . . . .	12
2.2.2 Sintaks yang Berguna [2] . . . . .	12
2.2.3 GitHub Flavored Markdown [3] . . . . .	16
2.3 StrapdownJS [4] . . . . .	18
2.4 Zurb Foundation [5] . . . . .	19
2.4.1 Kompatibilitas . . . . .	19
2.4.2 Apa Saja yang Hadir Dengan Foundation? . . . . .	20
<b>3 ANALISIS</b>	<b>25</b>
3.1 Analisis Google Authentication . . . . .	25
3.1.1 Langkah Dasar Penggunaan OAuth 2.0 . . . . .	25
3.1.2 Skenario Aplikasi . . . . .	33
3.2 Analisis Markdown . . . . .	34
3.3 Analisis StrapdownJS . . . . .	37
3.4 Analisis Zurb Foundation . . . . .	38
3.5 Analisis Berorientasi Objek . . . . .	40
3.5.1 Use Case Diagram . . . . .	41
3.5.2 Skenario . . . . .	42



## DAFTAR GAMBAR

2.1	Gambar Skenario Aplikasi Web Server . . . . .	7
2.2	Gambar Skenario Aplikasi yang Terinstal . . . . .	8
2.3	Gambar Skenario Aplikasi Sisi Klien (JavaScript) . . . . .	9
2.4	Gambar Skenario Aplikasi Pada Perangkat Dengan Masukan Yang Terbatas . . . . .	10
2.5	Gambar Skenario Layanan Akun . . . . .	11
2.6	Gambar Contoh Grid . . . . .	21
2.7	Gambar Contoh Tombol . . . . .	21
2.8	Gambar Contoh Navigasi . . . . .	22
2.9	Gambar Contoh Plugins . . . . .	23
3.1	Google Developers Console . . . . .	27
3.2	Membuat Proyek Baru . . . . .	28
3.3	Menu Credentials . . . . .	28
3.4	Membuat Client ID yang Baru . . . . .	29
3.5	Tipe Aplikasi . . . . .	30
3.6	Pengisian Tipe Aplikasi . . . . .	31
3.7	Client ID . . . . .	32
3.8	Izin Pihak Pengguna . . . . .	33
3.9	Skenario Aplikasi SIRM . . . . .	34
3.10	Output Sintaks Cetak Tebal dan Cetak Miring . . . . .	35
3.11	Output Sintaks Judul Bab . . . . .	36
3.12	Output Sintaks Batas Baris . . . . .	36
3.13	Output Sintaks Paragraf . . . . .	36
3.14	Output Sintaks Link . . . . .	36
3.15	Output Sintaks Daftar . . . . .	36
3.16	Output Keterangan Mahasiswa . . . . .	37
3.17	Tampilan pilihmahasiswa.php Dengan Zurb Foundation . . . . .	40
3.18	Use Case Diagram . . . . .	41

## DAFTAR TABEL

2.1	Tabel 2-1 Daftar Pengujian Zurb Foundation . . . . .	20
3.1	Tabel 3-1 Skenario Login . . . . .	42
3.2	Tabel 3-2 Skenario Pilih Mahasiswa . . . . .	43
3.3	Tabel 3-3 Skenario Melihat Info Mahasiswa . . . . .	43
3.4	Tabel 3-4 Skenario Edit Mahasiswa . . . . .	44
3.5	Tabel 3-5 Skenario Lihat Histori . . . . .	44
3.6	Tabel 3-6 Skenario Membuat Entri Baru . . . . .	45



# 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 Console<sup>1</sup>. 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 Connect<sup>2</sup>.

##### 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

---

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

<sup>2</sup><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

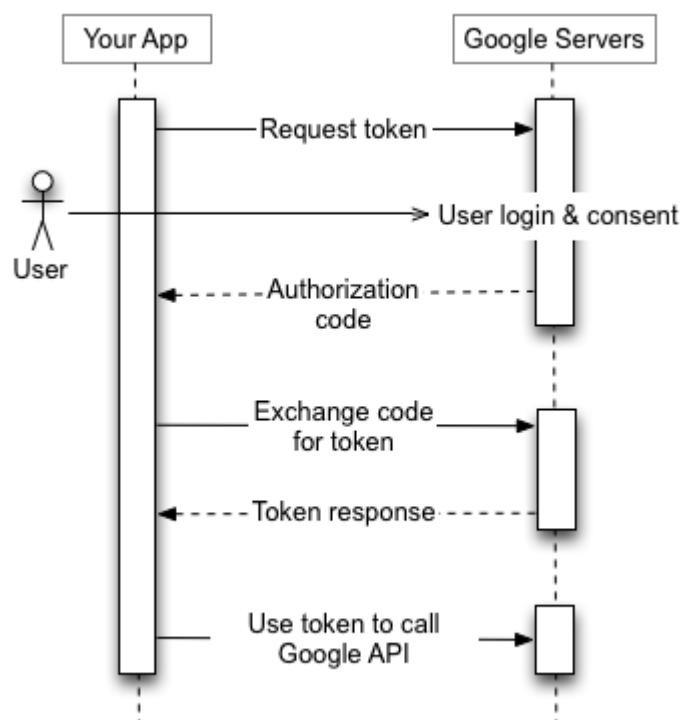
Token akses memiliki daya tahan yang terbatas. Jika aplikasi yang dibangun membutuhkan akses ke Google API melebihi masa aktif token akses, maka dapat memperbaharui token akses tersebut. Hal ini memungkinkan untuk mendapatkan token akses yang baru.

#### 2.1.2 Skenario 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

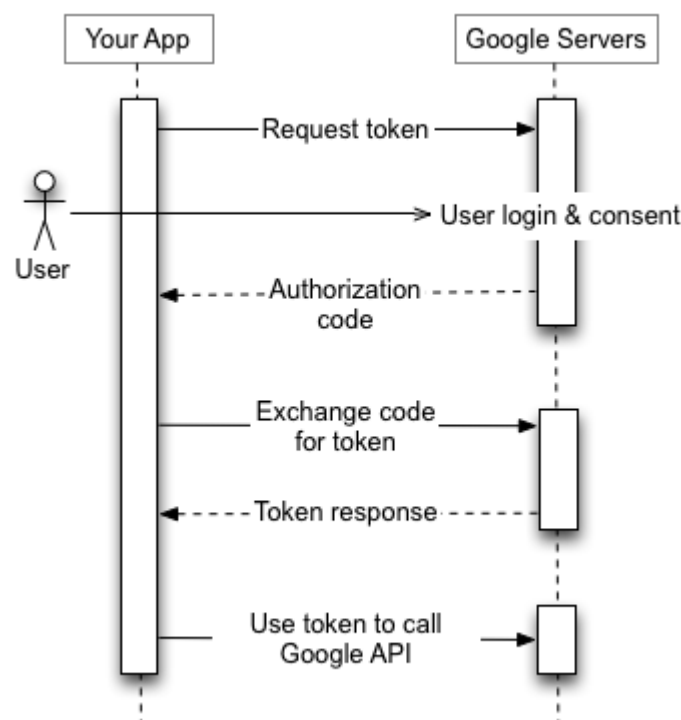
### 2.1.3 Skenario Aplikasi yang Terinstal

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

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

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

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

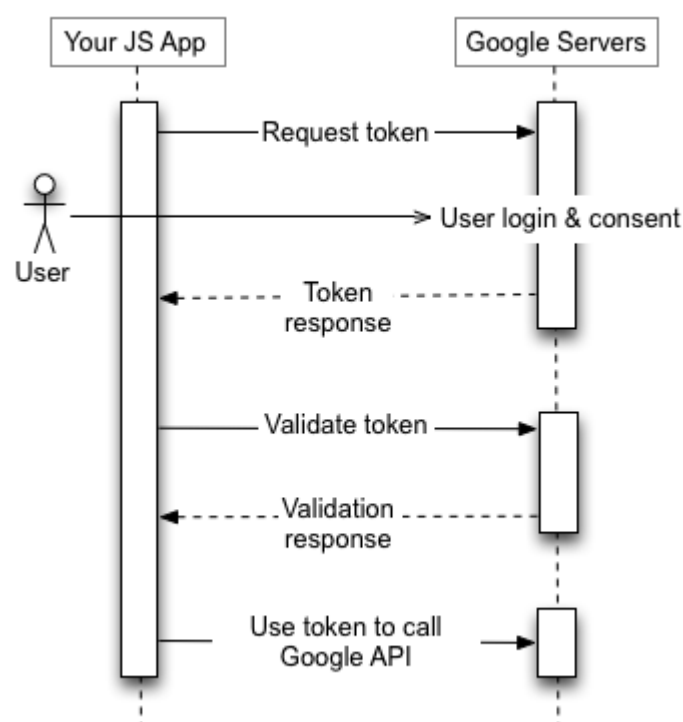


Gambar 2.2: Skenario Aplikasi yang Terinstal



#### 2.1.4 Skenario Aplikasi Sisi Klien (JavaScript)

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



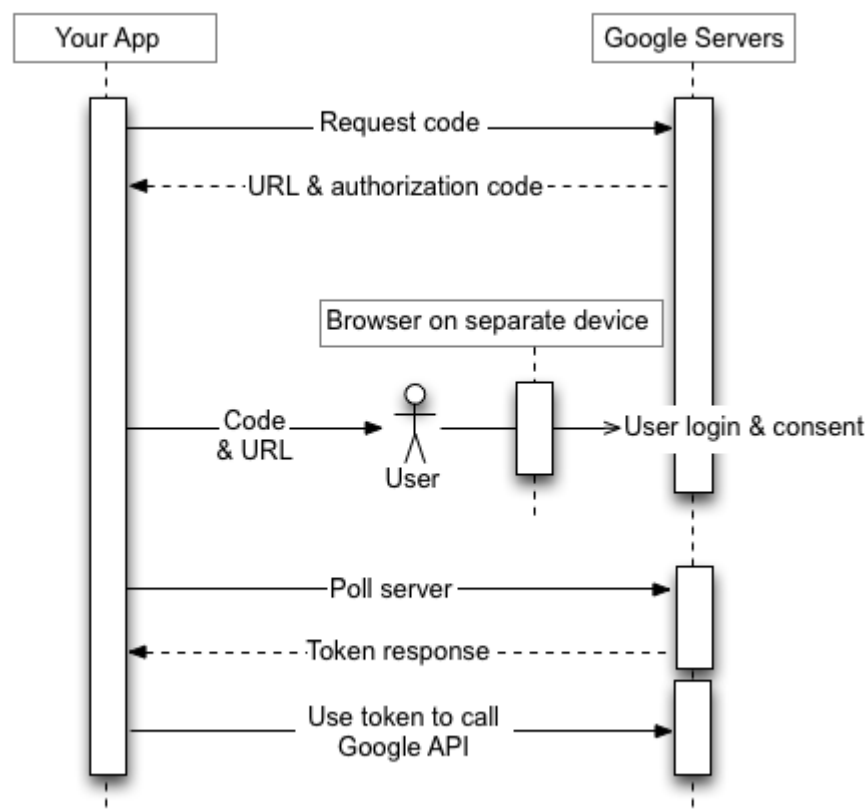
Gambar 2.3: Skenario Aplikasi Sisi Klien (JavaScript)

#### 2.1.5 Skenario Aplikasi Pada Perangkat Dengan Masukan Yang Terbatas

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

1 kode.

2 Sementara itu, aplikasi jajak pendapat dari URL Google pada interval tertentu. Setelah  
 3 pengguna menyetujui akses, respon dari server Google berisi token akses dan memperbaharui  
 4 token. Aplikasi harus menyimpan token yang baru untuk penggunaan masa depan dan  
 5 menggunakan token akses untuk mengakses Google API. Setelah masa token akses berakhir,  
 6 maka aplikasi akan memperbaharui token untuk mendapatkan yang baru. Untuk gambar  
 7 skenario dapat dilihat pada Gambar 2.4.



Gambar 2.4: Skenario Aplikasi Pada Perangkat Dengan Masukan Yang Terbatas

### 8 2.1.6 Skenario Layanan Akun

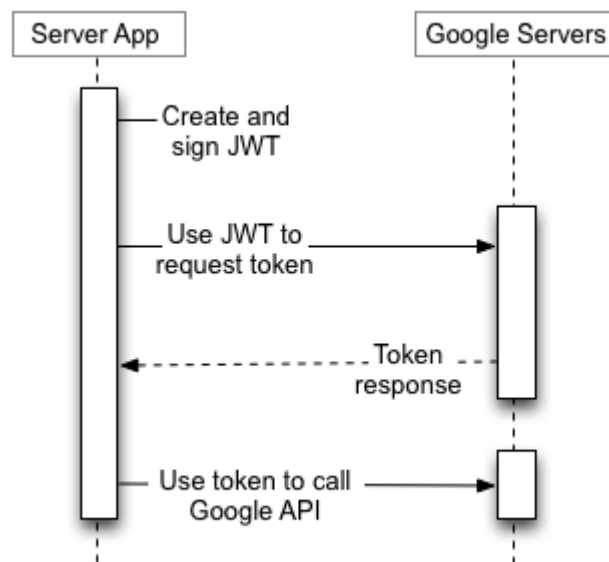
9 Google API seperti Prediction API dan Google Cloud Storage dapat bertindak atas nama  
 10 aplikasi yang dibuat tanpa mengakses informasi pengguna. Dalam situasi ini aplikasi perlu  
 11 membuktikan identitasnya sendiri ke API, tapi tidak diperlukan izin dari pihak pengguna.  
 12 Demikian pula, dalam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

14 Untuk jenis interaksi antara server memerlukan layanan akun, dimana akun tersebut  
 15 terdapat pada aplikasi yang dibuat, bukan individu ke pengguna akhir. Aplikasi memanggil

1 Google API atas nama layanan akun, dan izin dari pihak pengguna tidak diperlukan. (Dalam  
2 skenario tanpa layanan akun, aplikasi memanggil Google API atas nama pengguna akhir,  
3 dan izin dari pihak pengguna kadang-kadang diperlukan.)

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

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



Gambar 2.5: Skenario Layanan Akun

### 17 2.1.7 Masa Habis Berlaku Token

18 Kode token harus ditulis untuk mengantisipasi kemungkinan bahwa token yang diberikan  
19 mungkin tidak lagi bekerja suatu saat. Token mungkin berhenti bekerja untuk beberapa

1 alasan di bawah ini:

- 2     • Pengguna telah mencabut akses.
- 3     • Token tidak digunakan selama enam bulan.
- 4     • Akun pengguna telah melampaui jumlah tertentu permintaan token.

5     Saat ini batas untuk setiap akun Google adalah 25 token. Jika pengguna akun telah  
6 memiliki 25 token, permintaan otentikasi untuk token ke-26 akan berhasil tapi token yang  
7 paling tua atau token ke-1 akan dibuat tidak berlaku tanpa sepengetahuan pengguna. Jika  
8 perlu untuk mengotorisasi beberapa program, mesin, atau perangkat, salah satu solusi ada-  
9 lah untuk membatasi jumlah klien dimana harus mengotorisasi per pengguna akun antara  
10 15 atau 20. Jika Anda adalah admin Google Apps, Anda dapat membuat admin tambahan  
11 untuk mengizinkan beberapa klien.

## 12 2.2 Markdown

### 13 2.2.1 Apa itu Markdown? [2]

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

### 23 2.2.2 Sintaks yang Berguna [2]

24 Terdapat beberapa sintaks untuk penggunaan cetak tebal, cetak miring, judul sub bab, batas  
25 garis, paragraf, gambar, link, kode, kutipan, garis horisontal, dan list. Untuk penjelasan  
26 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)
##### Tingkat ke 6 (an <h6> tag)
```

### 2.2.2.3 Batas Baris

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

```
Baris ini dengan
batas baris
```

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

```
Baris ini tanpa
batas baris
```

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

### 2.2.2.4 Paragraf

Untuk menyisipkan paragraf baru, cukup menyisipkan satu baris kosong.

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

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

### 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/images/srpr/logollw.png "Google")
```

Url gambar dapat diganti dengan path yang mengarah pada file gambar yang akan digunakan.

### 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")
```

### 2.2.2.7 Kode

Untuk menyisipkan kode pada sebuah baris pada dokumen Markdown, gunakan karakter kutip belakang ('). Lampirkan kode yang ingin disisipkan dalam karakter kutip belakang.

Untuk menyisipkan blok kode gunakan tiga kutip belakang (``). Setiap baris kode harus diawali dengan empat spasi.

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

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

### 2.2.2.8 Kutipan

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

```
1 | Ini merupakan kutipan **favorit** saya:  
2 |  
3 | > The weak can never forgive.  
4 | > 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.

```
8 | # Bagian Satu  
9 |  
10 | The quick brown fox jumps over the lazy dog. The quick brpwn fox jump over the lazy dog.  
11 |  
12 | ——  
13 |  
14 | 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.

```
23 | * Item  
24 | * Item  
25 | * Item  
26 |  
27 | + Item  
28 | + Item  
29 | + Item  
30 |  
31 | - Item  
32 | - Item  
33 | - 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.

```
37 | 1. Item 1  
38 | 2. Item 2  
39 | 3. Item 3
```

## 2.2.3 GitHub Flavored Markdown [3]

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 yang mengarah ke URL tersebut.

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

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

### 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.~~
```

### 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.



```

1 | Here's an example:
2 |
3 | ```
4 | function test() {
5 |     console.log("notice the blank line before this function?");
6 | }
7 | ```

```

### 2.2.3.5 Penandaan Sintaks

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

```

12 | ```ruby
13 | require 'redcarpet'
14 | markdown = Redcarpet.new("Hello World!")
15 | puts markdown.to_html
16 | ```

```

subsubsectionTabel 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.

```

20 | First Header | Second Header
21 | -----|-----
22 | Content Cell | Content Cell
23 | Content Cell | Content Cell

```

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

```

26 | | First Header | Second Header |
27 | | -----|-----|
28 | | Content Cell | Content Cell |
29 | | 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.

```

32 | | Name | Description |
33 | | -----|-----|
34 | | Help | Display the help window. |
35 | | 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.

```

39 | | Name | Description |
40 | | -----|-----|
41 | | Help | ~~Display the~~ help window. |
42 | | 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.

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

## 2.3 StrapdownJS [4]

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

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

- Cara pertama:

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

- Cara kedua:

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

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

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

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

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

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

```
<!DOCTYPE html>
<html>
<title>Hello Strapdown</title>
<xmp theme="united" style="display:none;">
```

```

1      # Markdown text goes in here
2
3      ## Chapter 1
4
5      Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt
6          ut labore
7      et dolore magna aliqua.
8
9      ## Chapter 2
10
11     Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut
12     aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit
13         esse
14     cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt
15         in
16     culpa qui officia deserunt mollit anim id est laborum.
17     </xmp>
18
19     <script src="http://strapdownjs.com/v/0.2/strapdown.js"></script>
20     </html>

```

Strapdown.js juga memiliki beberapa fitur :

- (a) Ramah dengan mesin pencari
- (b) Kompatibel dengan berbagai browser (Sudah diuji dengan ponsel menggunakan Safari, IE 8/9, Firefox, Chrome)
- (c) Github menggunakan Markdown (Tabel, Syntax, Headline)
- (d) Dapat menggunakan tema

## 2.4 Zurb Foundation [5]

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.

Tabel 2.1: Daftar Pengujian Zurb Foundation<sup>1</sup>

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	✓	✓	✓

### 2.4.2 Apa Saja yang Hadir Dengan Foundation?

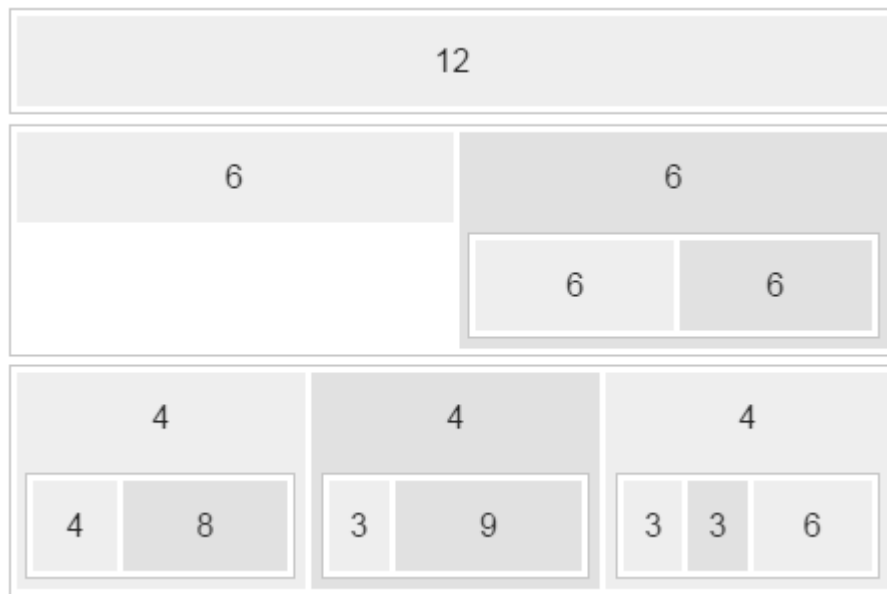
Foundation memiliki banyak komponen dan struktur untuk membantu membangun sebuah situs responsif. Semua komponen Foundation dapat dilihat pada satu halaman yang disebut Kitchen Sink<sup>2</sup> atau melihat beberapa gambar dibawah ini :

#### (a) The Grid

Grid bekerja pada hampir semua perangkat dan memiliki dukungan untuk menjadi 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.

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

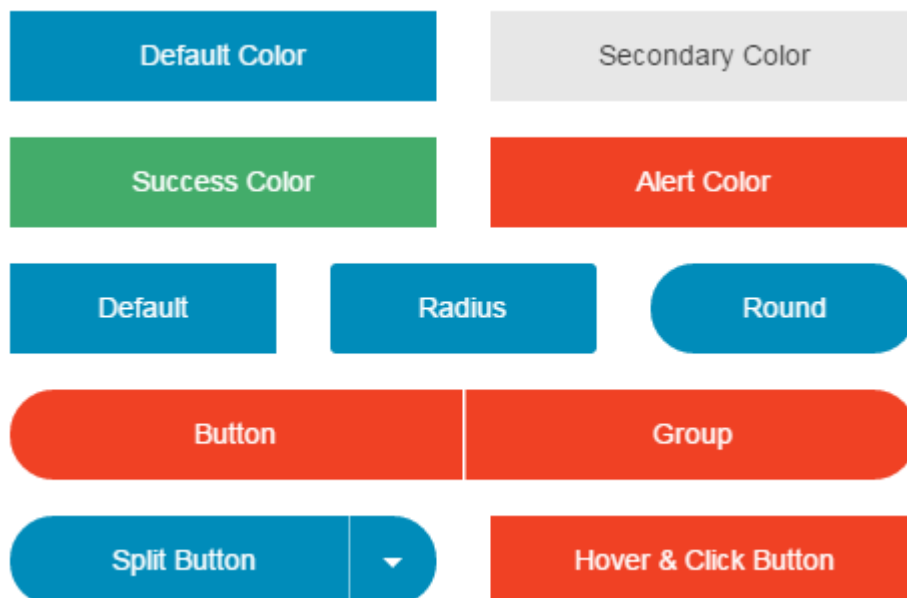
<sup>2</sup>[http://foundation.zurb.com/docs/components/kitchen\\_sink.htmljs](http://foundation.zurb.com/docs/components/kitchen_sink.htmljs)



Gambar 2.6: Contoh Grid

### 1 (b) Tombol

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

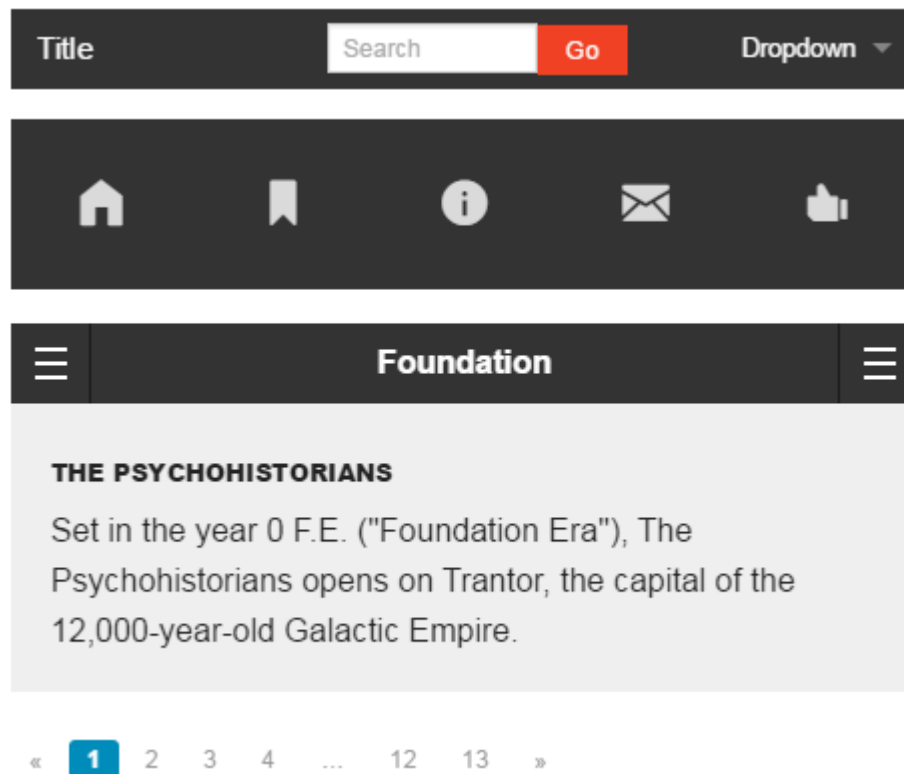


Gambar 2.7: Contoh Tombol

### 7 (c) Navigasi

8 Orang yang mengakses harus bisa berkeliling melihat menu-menu yang ada. Gaya

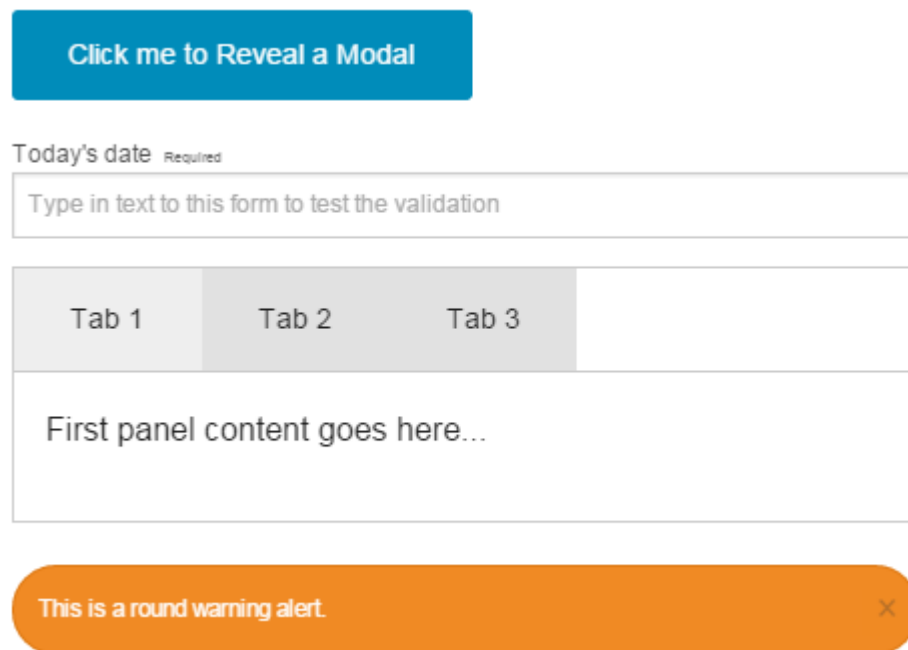
1 navigasi pada Foundation meliputi : bar bagian atas yang kuat dengan menu  
 2 dropdown; tombol; bar pencari; ikon bar yang keren; implementasi kanvas yang  
 3 lepas dari keluhan; dan sekelompok navigasi lainnya. Untuk contoh macam-  
 4 macam navigasi dapat dilihat pada Gambar 2.8.



Gambar 2.8: Contoh Navigasi

5 (d) Plugins

6 Sudah meliputi banyak plugin javascript yang ditulis untuk modal dasar pop-  
 7 up; menambat formulir validasi yang diperlukan; membuat tab konten; tanda  
 8 peringatan; dan masih banyak lagi. Untuk contoh macam-macam plugin dapat  
 9 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. 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'. 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. Dapat dilihat pada Gambar 3.5.
- vi. Isi bagian AUTHORIZED JAVASCRIPT ORIGINS (merupakan path dimana javascript otorisasi akan dijalankan) dan AUTHORIZED REDIRECT URIS (merupakan pengarah jika otorisasi sudah berhasil). Dapat dilihat pada Gambar 3.6.
- vii. Setelah langkah-langkah diatas terpenuhi maka akan mendapatkan client id dan client secret. 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.

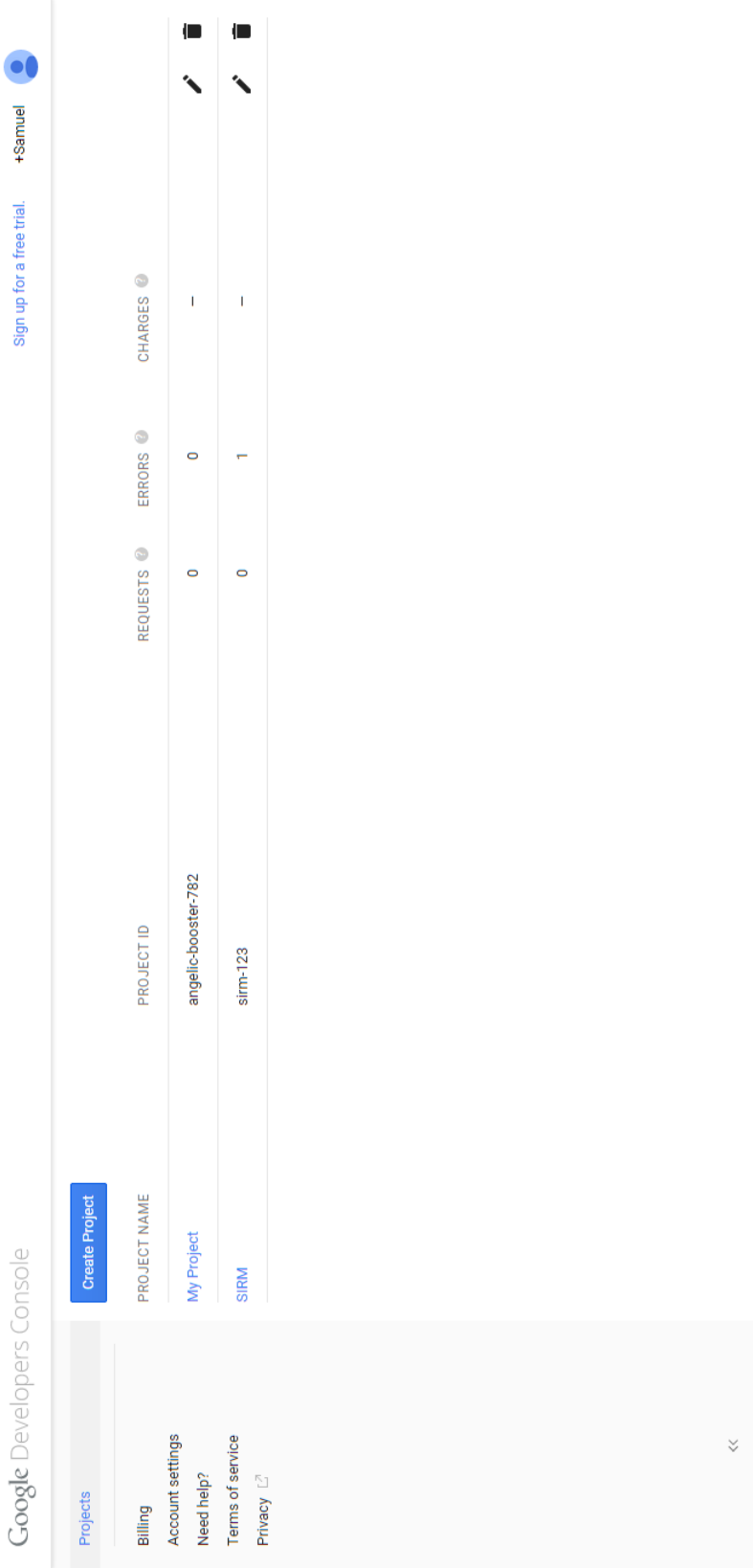
(c) Kirim token ke API

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

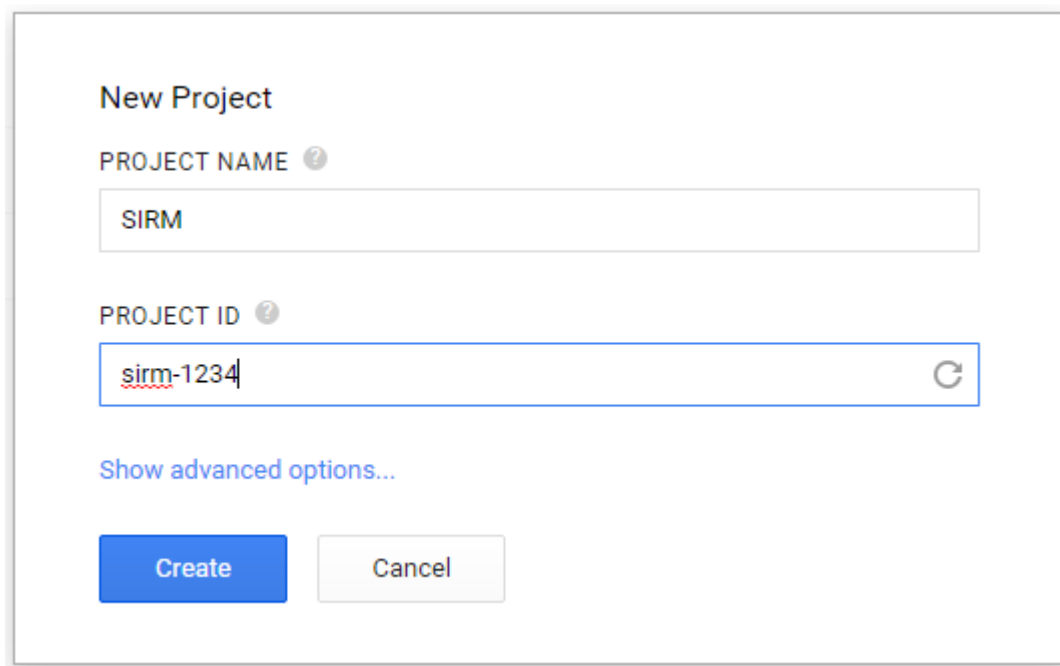
```
https://www.googleapis.com/auth/urlshortener  
https://www.googleapis.com/auth/userinfo.profile
```

(d) 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



**New Project**

PROJECT NAME ?

SIRM

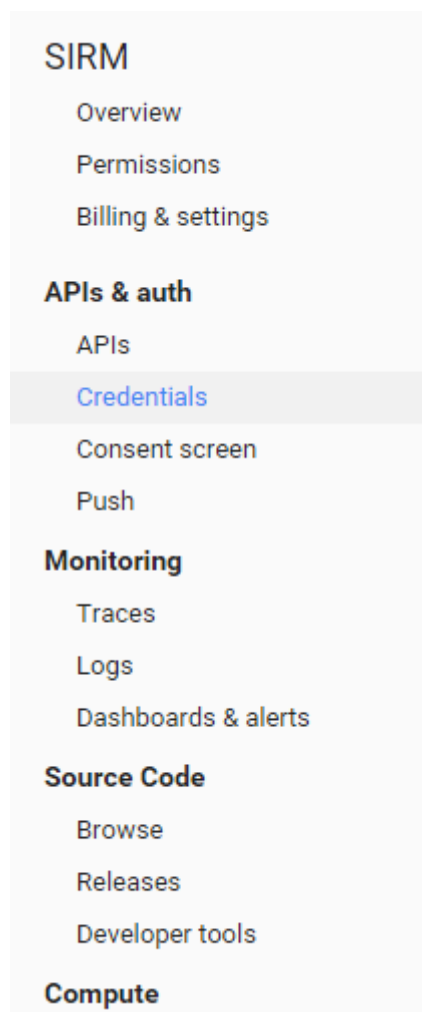
PROJECT ID ?

sirm-1234

[Show advanced options...](#)

Create Cancel

Gambar 3.2: Membuat Proyek Baru



Gambar 3.3: Menu Credentials

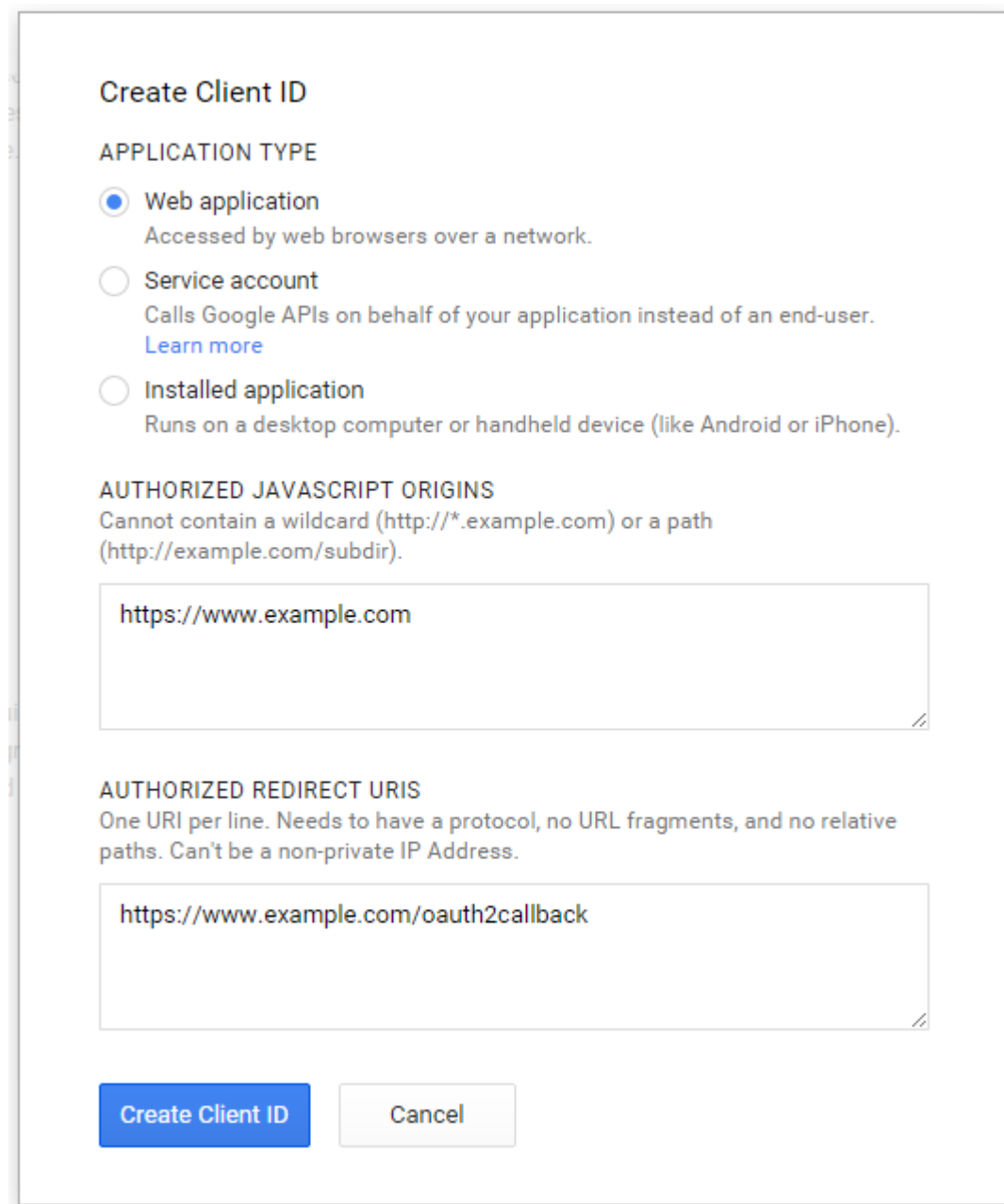
## OAuth

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).

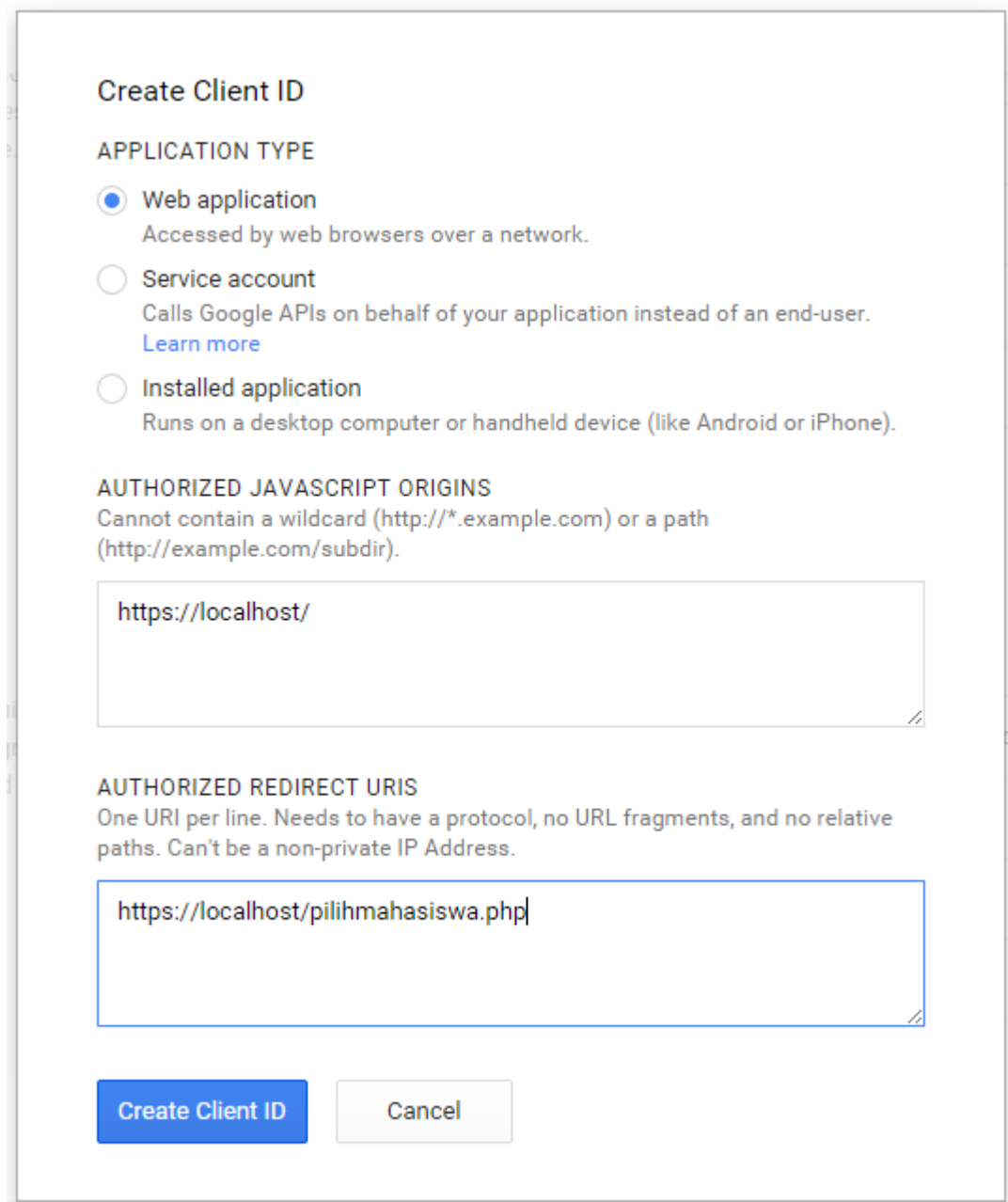
https://www.example.com

**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).

https://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.

https://localhost/pilihmahasiswa.php

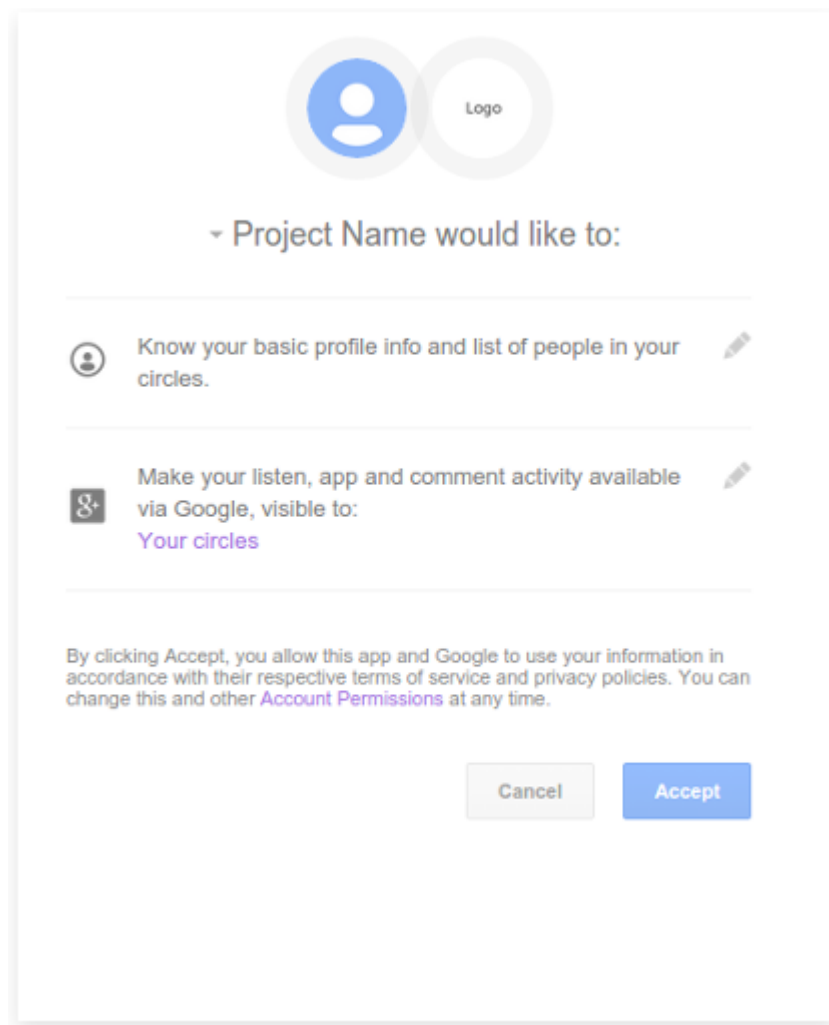
**Create Client ID** **Cancel**

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_IpiBBi
REDIRECT URIS	http://localhost/pilihmahasiswa.php
JAVASCRIPT ORIGINS	http://localhost/
<div>Edit settings</div> <div>Reset secret</div> <div>Download JSON</div> <div>Delete</div>	

Gambar 3.7: Client ID

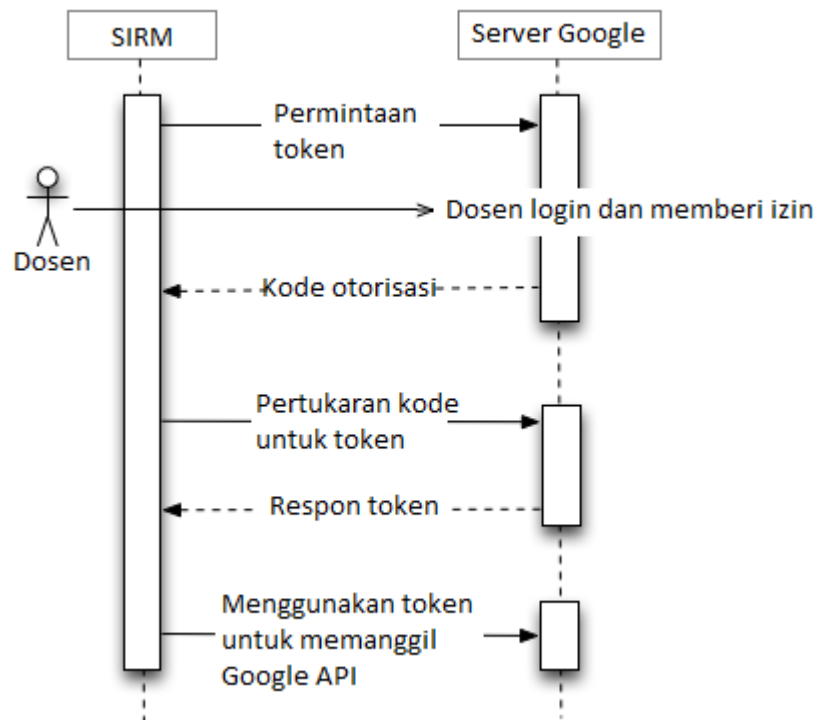




Gambar 3.8: Izin Pihak Pengguna

### 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.

- Sintaks Cetak Tebal dan Cetak Miring

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

```

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

- Sintaks Judul Bab

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

```

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://ble tack .blogspot .com /](http://ble tack .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://ble tack .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>
```

Skip 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://blelack.blogspot.com/](http://blelack.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>

```

### 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 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  <div class="row">
9      <h5>Masukan NPM yang ingin dicari / tambah baru.</h5>
10 </div>
11 <form>
12     <div class="row">
13         <div class="large-12 columns">
14             <div class="row collapse">
15                 <div class="small-8 columns">
16                     <input type="text" placeholder="NPM">
17                 </div>
18                 <div class="small-2 columns">
19                     <a href="#" class="button postfix">Search</a>
20                 </div>
21                 <div class="small-2 columns">
22                     <a href="entribaru.php" class="button postfix">Add</a>
23                 </div>
24             </div>
25         </div>
26     </div>
27 </form>
28 <div class="row">
29 <?php
30     $pemakai="admin ";
31     $pass="admin ";
32     $id_mysql=mysql_connect("localhost", $pemakai, $pass);
33
34     if (! $id_mysql){
35         die("Database tidak bisa dibuka");
36     }
37
38     if (! mysql_select_db("sirm", $id_mysql)){
39         die("Database tidak bisa dipilih");
40     }
41
42     $hasil = mysql_query("SELECT * FROM info_mahasiswa", $id_mysql);
43
44     if (! $hasil){
45         die("Permintaan gagal");
46     }
47
48     echo "<table>
49 <thead>
50 <tr>
51 <th width='250'>NPM</th>
52 <th width='500'>Nama</th>
53 <th width='250'>Last Update</th>
54 </tr>
55 </thead>";
56
57     while($row = mysql_fetch_array($hasil))
58     {
59         echo "<tr>";
60         echo "<td><a href='infomahasiswa.php'>" . $row['npm'] . "</a></td>";
61         echo "<td>" . $row['nama'] . "</td>";
62         echo "<td>" . $row['log'] . "</td>";
63         echo "</tr>";
64     }

```

```

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

```

Masukan NPM yang ingin dicari / tambah baru.

NPM	Search	Add
NPM	Nama	Last Update
<a href="#">2010730001</a>	Andri Agustian	2015-02-28 23:07:01
<a href="#">2010730013</a>	Samuel Herman	2015-02-28 23:20:46
<a href="#">2010730100</a>	Bedu	2015-02-28 21:14:36

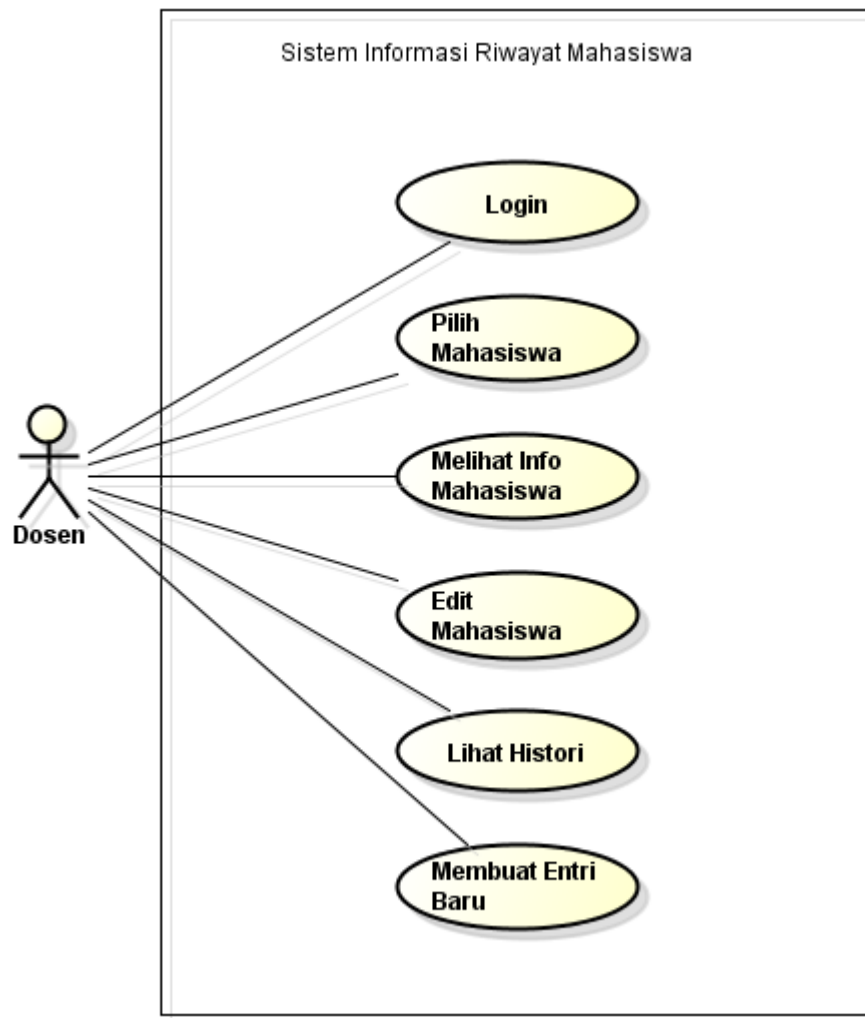
Gambar 3.17: Tampilan pilihmahasiswa.php Dengan Zurb Foundation

### 8 3.5 Analisis Berorientasi Objek

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



## 1      3.5.1 Use Case Diagram



Gambar 3.18: Use Case Diagram

2      Use case diagram merupakan pemodelan yang menunjukkan kegiatan apa saja yang  
3      dapat dilakukan pengguna dan kegiatan yang dilakukan sistem. Berikut adalah desk-  
4      rripsi dari use 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 di-  
9      lihat infonya. Selain itu pengguna juga bisa menemukan tombol "Add" untuk  
10     menambah 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		

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.



## DAFTAR REFERENSI

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