

SKRIPSI

**SISTEM PEREKAM DAN BERBAGI RIWAYAT MAHASISWA
UNTUK DOSEN**



SAMUEL HERMAN

NPM: 2010730013

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

UNDERGRADUATE THESIS

**STUDENT HISTORY RECORDING AND SHARING SYSTEM
FOR LECTURERS**



SAMUEL HERMAN

NPM: 2010730013

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

LEMBAR PENGESAHAN

**SISTEM PEREKAM DAN BERBAGI RIWAYAT MAHASISWA
UNTUK DOSEN**

SAMUEL HERMAN

NPM: 2010730013

Bandung, 27 Mei 2015

Menyetujui,

Pembimbing Tunggal

Dr. rer. nat. Cecilia Esti Nugraheni

Ketua Tim Penguji

Anggota Tim Penguji

Lionov, M.Sc.

Dr. Veronica Sri Moertini

Mengetahui,

Ketua Program Studi

Thomas Anung Basuki, Ph.D.

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

SISTEM PEREKAM DAN BERBAGI RIWAYAT MAHASISWA UNTUK DOSEN

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal 27 Mei 2015

Meterai

Samuel Herman
NPM: 2010730013

ABSTRAK

«Tuliskan abstrak anda di sini, dalam bahasa Indonesia»

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

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

ABSTRACT

«Tuliskan abstrak anda di sini, dalam bahasa Inggris»

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Keywords: Google OAuth, Markdown, StrapdownJS, Zurb Foundation

Dipersembahkan untuk diri sendiri

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Di tahun 2014 terdapat 14 dosen di Falkutas Teknologi Informasi dan Sains [1] yang memiliki tugas untuk mengajar, menjadi dosen wali, dan menjadi dosen pembimbing bagi mahasiswa yang sedang menyusun skripsi. Sedangkan mahasiswa yang berinteraksi dengan dosen pada saat pembelajaran di kelas, perwalian, maupun bimbingan skripsi belum tentu sama. Maka dari itu jumlah dosen dan jumlah mahasiswa menjadi pemasalahan. Permasalahan tersebut disebabkan karena minimnya jumlah dosen yang diharuskan berinteraksi dengan banyak mahasiswa. Minimnya jumlah dosen mengakibatkan seorang dosen kesulitan dalam mengingat perkembangan setiap mahasiswa serta sejarah riwayat setiap mahasiswa, terutama terkaitnya interaksi dengan dosen lain.

Berdasarkan jabaran masalah diatas, maka solusi untuk membantu para dosen adalah membuat sebuah sistem yang dapat mencatat dan membagikan riwayat mahasiswa. Dimana sistem hanya dapat diakses oleh dosen dengan menggunakan akun dosen yang diberikan oleh pihak UNPAR. Dalam sistem ini dosen dapat melakukan beberapa aksi. Pertama, dosen dapat melihat data mahasiswa yang telah terdaftar di basis data. Kedua, dosen dapat melihat data informasi mahasiswa yang ingin dilihat. Ketiga, dosen dapat mengubah data informasi mahasiswa. Keempat, dosen dapat melihat histori yang dimiliki setiap mahasiswa (untuk aksi melihat, mengedit, membuat entri baru yang dilakukan dosen) dan dapat melihat setiap versi data informasi mahasiswa mulai dari pertama kali dibuat sampai yang terakhir. Kelima, dosen dapat membuat entri baru untuk mahasiswa yang belum tercatat pada basis data.

Untuk membangun sistem tersebut digunakan empat teknologi antara lain; Google OAuth, Markdown Syntax, StrapdownJS, Zurb Foundation. Google OAuth akan digunakan untuk fungsi login dalam mengautentikasi dan mengotorisasi pengguna yang menggunakan akun dosen UNPAR. Markdown Syntax akan digunakan untuk membuat informasi mahasiswa dengan format penulisan yang mudah dibaca dan ditulis oleh dosen sehingga penulisan pada sistem akan seragam. StrapdownJS akan digunakan pada sistem untuk mengkonversi format penulisan yang dibuat menggunakan Markdown Syntax menjadi tampilan HTML. Zurb Foundation akan digunakan untuk membuat tampilan antar muka menjadi responsif.

1.2 Rumusan Masalah

Berdasarkan latar belakang maka dapat dirumuskan permasalahan sebagai berikut:

- Bagaimana mengautentikasi akun dosen UNPAR?
- Bagaimana membuat format untuk penulisan teks yang dapat digunakan oleh semua dosen?
- Bagaimana menampilkan teks dengan format yang telah dibuat ke halaman website?
- Bagaimana merancang antarmuka sistem menggunakan Zurb Foundation?

1.3 Tujuan

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

- Mengautentikasi akun dosen UNPAR.
- Membuat format untuk penulisan teks yang dapat digunakan oleh semua dosen.
- Menampilkan teks dengan format yang telah dibuat ke halaman website.
- Merancang antarmuka sistem menggunakan Zurb Foundation.

1.4 Batasan Masalah

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

- Sistem membutuhkan akses internet.
- Sistem tidak terintegrasi dengan Sistem Informasi Akademik.

1.5 Metodologi Penelitian

Metodologi yang digunakan untuk menyusun penelitian:

- Melakukan survei kebutuhan pengguna.
- Melakukan studi pustaka mengenai teknologi yang akan digunakan untuk membangun perangkat lunak.
- Menganalisis cara kerja teknologi yang akan digunakan untuk membangun perangkat lunak.
- Merancang perangkat lunak yang akan dibuat.
- Melakukan implementasi untuk perangkat lunak yang telah dirancang ke dalam *script* PHP.
- Melakukan pengujian perangkat lunak yang telah diimplementasikan.

1.6 Sistematika Pembahasan

Sistematika pembahasan dalam penelitian ini adalah sebagai berikut:

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

BAB 2

LANDASAN TEORI

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

2.1 Google OAuth [2]

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 dukungan dari Google. Rincian tentang cara menggunakan OAuth 2.0 untuk otentikasi (yaitu *sign-in*), dapat dilihat pada OpenID Connect².

2.1.1 Langkah Dasar

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

- (1) Mendapatkan kepercayaan OAuth 2.0 dari Google Developers Console

Berkunjung ke Google Developers Console untuk mendapatkan kepercayaan OAuth 2.0 seperti klien id dan kerahasiaan klien yang keduanya dikenal oleh Google dan aplikasi yang dibuat. Set nilai-nilai yang bervariasi sesuai dengan jenis aplikasi apa yang sedang dibuat. Misalnya, sebuah aplikasi JavaScript tidak memerlukan sebuah rahasia, tapi apakah aplikasi layanan *web* memerlukannya.

- (2) Memperoleh token akses dari Google Authorization Server

Sebelum aplikasi dapat mengakses data privat dengan menggunakan Google API, terlebih dahulu diperlukan token akses untuk mengakses API tersebut. Satu token akses dapat memberikan berbagai tingkat akses ke beberapa API. Izin token akses merupakan parameter untuk variabel

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

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

ruang lingkup yang mengontrol sumber daya dan operasi. Selama ada permintaan untuk token akses, maka aplikasi akan mengirimkan satu atau lebih nilai pada parameter ruang lingkup.

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

(3) Kirim token akses ke API

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

(4) Memperbaharui token akses jika diperlukan

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

2.1.2 Skenario Google OAuth

Terdapat lima skenario yang dapat digunakan untuk Google OAuth yaitu Skenario Aplikasi *Web* Server, Skenario Aplikasi yang Terinstal, Skenario Aplikasi Sisi Klien (JavaScript), Skenario Aplikasi Pada Perangkat Dengan Masukan Yang Terbatas, dan Skenario Layanan Akun. Untuk penjelasan lebih lanjut dapat dilihat pada daftar di bawah ini.

- Skenario Aplikasi Layanan *Web*

Google OAuth 2.0 mendukung aplikasi Layanan *Web* yang menggunakan bahasa dan kerangka kerja seperti PHP, Java, Python, Ruby, dan ASP.NET. Urutan otorisasi dimulai ketika aplikasi mengarahkan mesin pencari ke URL Google; URL tersebut termasuk parameter permintaan yang menunjukkan jenis akses yang diminta. 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.

- 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.

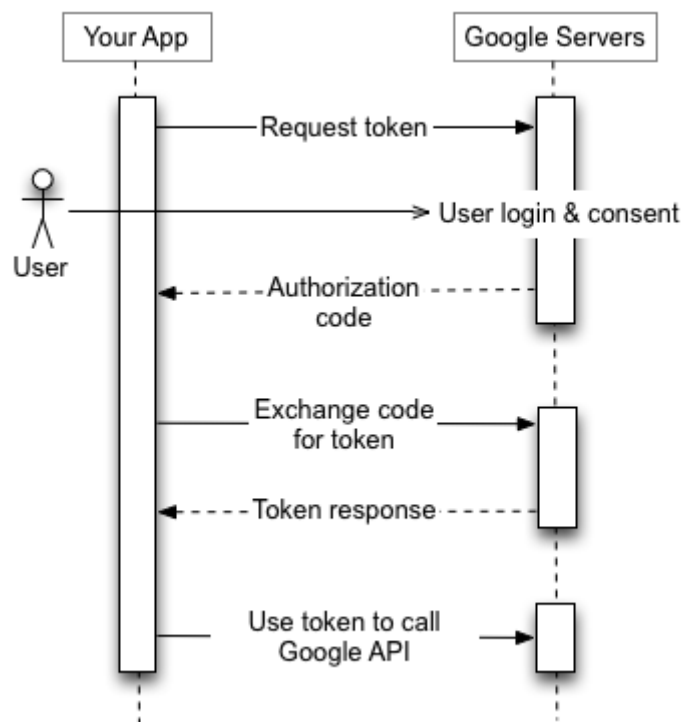
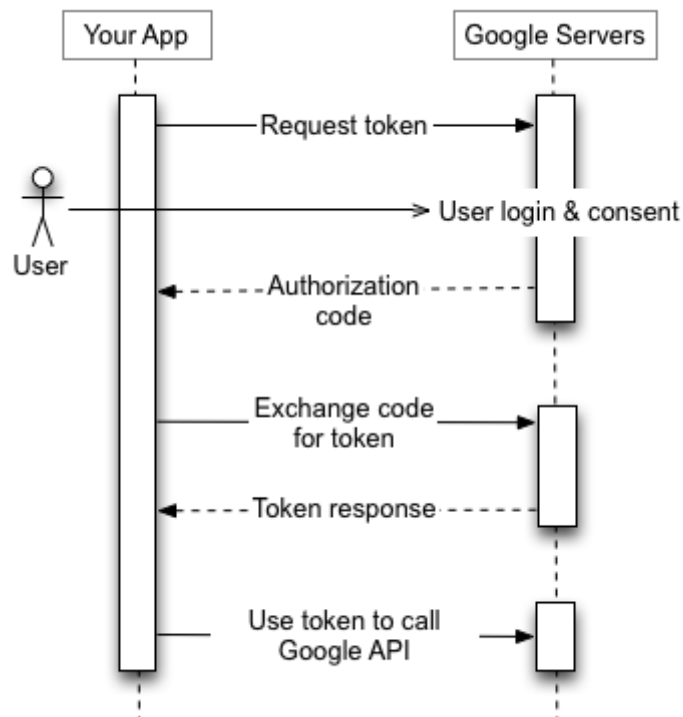
- Skenario Aplikasi Sisi Klien (JavaScript)

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 otentikasi 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.

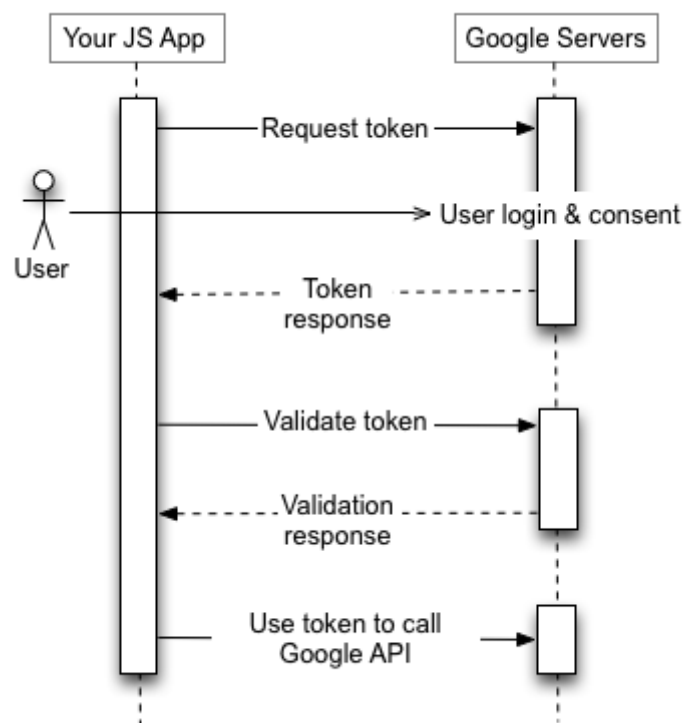
- Skenario Aplikasi Pada Perangkat Dengan Masukan Yang Terbatas

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

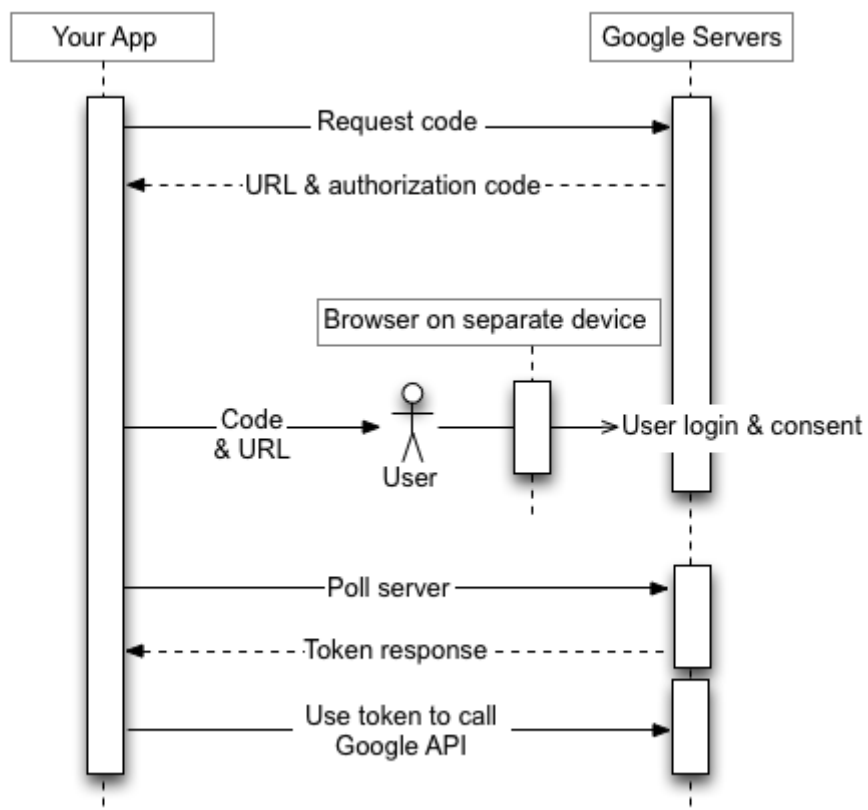
Sementara itu, aplikasi jajak pendapat dari URL Google pada interval tertentu. Setelah pengguna menyetujui akses, respon dari server Google berisi token akses dan memperbaharui token. Aplikasi harus menyimpan token yang baru untuk penggunaan masa depan dan menggunakan token akses untuk mengakses Google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

Gambar 2.1: Skenario Aplikasi Layanan *Web*

Gambar 2.2: Skenario Aplikasi yang Terinstal



Gambar 2.3: Skenario Aplikasi Sisi Klien (JavaScript)



Gambar 2.4: Skenario Aplikasi Pada Perangkat Dengan Masukan Yang Terbatas

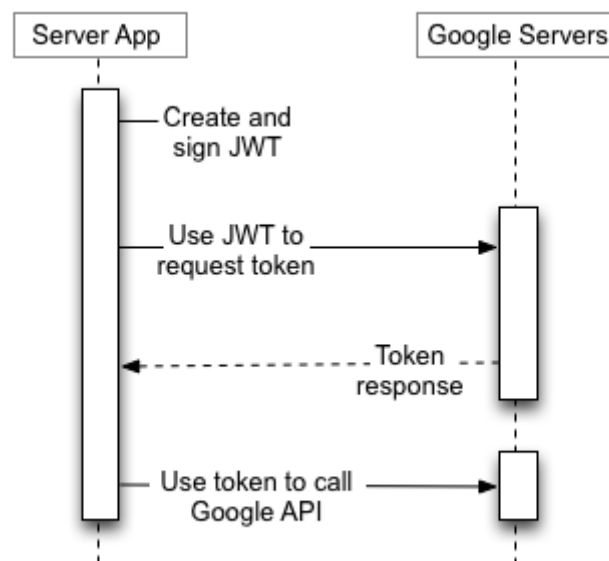
- Skenario Layanan Akun

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

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

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

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



Gambar 2.5: Skenario Layanan Akun

2.1.3 Masa Habis Berlaku Token

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

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

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

2.1.4 Lingkup Otorisasi [3]

Lingkup disini merupakan sebuah *string* yang memungkinkan akses ke sumber daya tertentu, misalnya akses ke data pengguna. Dengan memasukan lingkup tertentu pada saat permintaan otorisasi, kemudian mendapatkan izin sesuai dengan teks yang akan ditampilkan ke pengguna. Setelah mendapat persetujuan dari pihak pengguna untuk izin atas lingkup tersebut, maka Google mengirimkan token untuk aplikasi yang mengidentifikasi untuk memberikan otorisasi khusus. Dengan kata lain, lingkup dan token menentukan apa saja data pengguna yang diberi izin oleh pengguna untuk diakses.

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

- Lingkup Profil

1 | profile

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

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

- Memungkinkan aplikasi diakses melalui aplikasi android.

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

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

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

Lingkup ini juga memungkinkan lintas platform dengan pendaftaran tunggal.

- Lingkup Email

1| `email`

Lingkup ini meminta agar aplikasi diberikan akses ke:

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

1| `people.getOpenIdConnect`

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

- Nama domain Google Apps jika ada yang dimiliki pengguna. Nama domain dikembalikan sebagai kepemilikan domain dari `people.get` atau properti `hd` dari `getOpenIdConnect`.
- Lingkup email ini setara dan menggantikan lingkup di bawah ini.

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

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

Lingkup ini meminta aplikasi agar diberikan akses ke:

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

- Lingkup yang lain

1| `openid`

Lingkup `openid` menginformasikan server otorisasi bahwa klien membuat permintaan OpenID Connect dan meminta akses ke id pengguna yang terotentikasi tersebut. Lingkup ini harus disertakan lingkup OpenId Connect.

Metode `getOpenIdConnect` mengembalikan profil pengguna dengan format OIDC mengikuti jalur permintaan HTTP:

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

Untuk keperluan login menggunakan lingkup profil atau lingkup

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

karena lingkup

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

tidak dianjurkan sebagai lingkup login dikarenakan pengguna yang belum *upgrade* ke Google+ tidak akan mengembalikan nama atau alamat email pengguna.

Lingkup ini melakukan hal berikut:

- Memungkinkan aplikasi untuk mengetahui siapa pengguna yang dikonfirmasi dengan mengganti id pengguna dengan "me" yang mewakili pengguna yang telah terotentikasi disetiap permintaan yang dilakukan.

- Lingkup yang tidak dipakai lagi

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

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

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

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

Lihat juga lingkup terkait:

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

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

2.1.5 API *Client Library* untuk PHP [4]

Google OAuth menggunakan API *Client Library* untuk diimplementasikan ke dalam bahasa pemrograman PHP. Sebelum itu ada beberapa langkah yang perlu diselesaikan sebelum menggunakan *library* ini. Beberapa langkah tersebut dapat dilihat di bawah ini.

- Jika belum memiliki akun Google, maka harus melakukan pendaftaran terlebih dahulu.
- Jika belum pernah membuat proyek Google API, maka buat proyek di Developer Console Google.
- Melakukan instal *library*.

Diperlukan pemahaman dasar-dasar bagaimana autentikasi dan otorisasi yang ditangani oleh API. Semua panggilan API harus menggunakan izin akses. Banyak metode API yang membutuhkan izin akses, tetapi beberapa ada yang tidak membutuhkan izin akses. Beberapa metode API yang dapat digunakan untuk beberapa perilaku, tergantung pada akses yang digunakan adalah akses sederhana atau akses dengan izin. Berikut dua metode untuk menentukan jenis akses.

- Akses sederhana API (kunci API)

Metode ini tidak dapat mengakses data pribadi pengguna. Aplikasi harus mengotentikasi sendiri sebagai aplikasi milik Google Developer Console *project*. Metode ini diperlukan untuk mengukur penggunaan proyek untuk tujuan akuntansi. Konsep penting untuk kunci API pada saat mengotentikasi aplikasi, gunakan kunci API untuk proyek Google Developer Console yang dimiliki. Setiap akses sederhana untuk memanggil aplikasi harus menyertakan kunci ini. Peringatan dalam menggunakan kunci API, jika seseorang memperoleh kunci yang anda miliki maka mereka bisa menggunakan untuk memakan kuota atau dikenakan tuduhan terhadap proyek anda.

- Akses otorisasi API (OAuth 2.0)

Metode ini digunakan untuk mengakses data pribadi pengguna. Sebelum mengakses otorisasi API, pengguna yang memiliki akses ke data pribadi harus memberikan akses ke aplikasi yang digunakan. Oleh karena itu, aplikasi yang digunakan harus disahkan, pengguna harus memberikan akses untuk aplikasi tersebut. Semua ini dapat dicapai dengan OAuth 2.0 dan *library*. Konsep penting untuk OAuth 2.0 sebagai berikut.

- Ruang lingkup

Setiap API mendefinisikan satu atau lebih lingkup yang menyatakan seperangkat operasi diizinkan. Sebagai contoh, sebuah API mungkin telah membaca saja dan baca tulis lingkup. Ketika aplikasi meminta akses ke data pengguna, permintaan harus menyertakan satu atau lebih ruang lingkup. Pengguna perlu untuk menyetujui akses ruang lingkup yang diminta aplikasi anda.

- Pembaruan dan akses token

Bila pengguna memberikan akses pada aplikasi, server otorisasi OAuth 2.0 menyediakan aplikasi dengan pembaruan dan akses token. Token ini hanya berlaku untuk ruang lingkup yang diminta. Aplikasi yang menggunakan token akses untuk mengotorisasi panggilan API. Token akses berakhir, tapi pembaruan token tidak. Aplikasi dapat menggunakan pembaruan token untuk mendapatkan token akses baru. Peringatan untuk menjauhkan pembaruan dan token akses pribadi. Jika seseorang memperoleh token anda, mereka bisa menggunakannya untuk mengakses data pribadi pengguna.

- ID klien dan rahasia klien

ID klien dan rahasia klien merupakan sebuah string unik yang mengidentifikasi aplikasi yang digunakan dan digunakan untuk memperoleh token. Keduanya diciptakan untuk proyek Google Developers Console pada panel API Access yang terdapat pada Developer Console Google. Ada tiga jenis ID klien, jadi pastikan jenis yang digunakan sesuai dengan aplikasi yang akan digunakan. Berikut ketiga jenis aplikasi.

- * Web application client IDs
- * Installed application client IDs
- * Service Account client IDs

Peringatan untuk menjauhkan rehasia klien anda pribadi. Jika seseorang memperoleh rehasia klien anda, mereka bisa menggunakannya untuk mengkonsumsi kuota, dikenakan tuduhan terhadap proyek Google Developer Console, dan meminta akses ke data pengguna.

Pada bagian terakhir ini menggambarkan bagaimana membangun suatu objek layanan API yang spesifik, membuat panggilan ke layanan, dan proses respon. Terdapat empat bagian untuk membangun dan memanggil layanan.

- Membangun objek klien objek

Klient objek adalah wadah utama untuk kelas dan konfigurasi di *library*.

```
$client = new Google_Client();
$client->setApplicationName("My Application");
$client->setDeveloperKey("MY_SIMPLE_API_KEY");
```

- Membangun layanan objek

Layanan dipanggil melalui permintaan untuk layanan onjek tertentu. Ini diciptakan dengan membangun layanan objek, dan melewati sebuah contoh dari `Google_Client` untuk itu. `Google_Client` berisi IO, otentikasi dan kelas-kelas lain yang diperlukan oleh layanan berfungsi, dan layanan menginformasikan klien yang menggunakan ruang lingkup untuk memberikan default ketika otentikasi pengguna.

```
$service = new Google_Service_Books($client);
```

- Memanggil API

Setiap API menyediakan sumber dan metode, biasanya dalam rantai. Ini dapat diakses dari objek layanan dalam bentuk `$service->resource->method(args)`. Kebanyakan metode memerlukan beberapa argumen, maka menerima parameter akhir dari array yang berisi parameter opsional. Misalnya, dengan Google Books API, kita dapat membuat panggilan ke daftar volume pencocokan string tertentu, dan menambahkan parameter filter yang opsional.

```
$optParams = array('filter' => 'free-ebooks');
$results = $service->volumes->listVolumes('Henry David Thoreau', $optParams);
```

- Penanganan hasil

Ada dua jenis utama dari respon yaitu item dan koleksi item. Masing-masing dapat diakses baik sebagai objek atau sebagai array. Koleksi penerapan antarmuka iterator sehingga dapat digunakan dalam konstruksi itu sendiri dan konstruksi lainnya.

```
foreach ($results as $item) {
    echo $item['volumeInfo']['title'], "<br /> \n";
}
```

2.2 Markdown

2.2.1 Apa itu Markdown? [?]

John Gruber pembuat Markdown, memperkenalkan Markdown sebagai alat konferensi 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 konferensi ke HTML dengan banyak perangkat lunak yang mendukung. Untuk contoh, jika menulis **hello** pada Markdown dan konferensi ke HTML menggunakan teks editor yang mendukung, teks tersebut akan menjadi `hello` yang akan terlihat **hello**.

2.2.2 Sintaks yang Berguna [?]

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

* Cetak Tebal dan Cetak Miring

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

```
1 | Alice melaporkan bahwa dirinya mengidap penyakit *Amyotrophic Lateral
   | Sclerosis* sehingga mengganggu proses belajar.
2 |
3 | Bob datang terlambat di kelas algoritma, dan **menolak untuk keluar**
   | saat diusir dosen kelas.
```

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

```
1 | <p><em>hello </em> untuk cetak miring </p>
2 |
3 | <p><strong>hello </strong> untuk cetak tebal </p>
```

hello untuk cetak miring

hello untuk cetak tebal

Gambar 2.6: Markdown Cetak Tebal dan Cetak Miring

* Judul Bab

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

```

1 # Judul Bab
2
3 ## Judul Sub Bab
4
5 ### Judul Sub Sub Bab
6
7 #### Tingkat ke 4
8
9 ##### Tingkat ke 5
10
11 ##### Tingkat ke 6

```

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

```

1 <h1>Judul Bab</h1>
2
3 <h2>Judul Sub Bab</h2>
4
5 <h3>Judul Sub Sub Bab</h3>
6
7 <h4>Tingkat ke 4</h4>
8
9 <h5>Tingkat ke 5</h5>
10
11 <h6>Tingkat ke 6</h6>

```

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

Gambar 2.7: Markdown Judul Bab

- * Batas Baris

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

```

1 | Baris ini dengan

```

2 | batas baris

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

```
1 | <p>Baris ini dengan <br />
2 | batas baris</p>
```

Baris ini dengan
batas baris

Gambar 2.8: Markdown Batas Baris Dengan Tiga Spasi

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

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

```
1 | <p>Baris ini tanpa
2 | batas baris</p>
```

Baris ini tanpa batas baris

Gambar 2.9: Markdown Batas Baris Tanpa Tiga Spasi

* Paragraf

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

```
1 | Ini kalimat pertama. Ini kalimat berikutnya. Ini kalimat terakhir.
2 |
3 | Ini paragraf baru.
```

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

```
1 | <p>Ini kalimat pertama. Ini kalimat berikutnya. Ini kalimat terakhir.</p>
   |     p>
2 |
3 | <p>Ini paragraf baru.</p>
```

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

Gambar 2.10: Markdown Paragraf

* Gambar

Untuk menyisipkan gambar pada dokumen Markdown, gunakan sintaks berikut

```
1 | ![ teks ]( / url_gambar "judul_gambar")
```

Contoh penggunaan sintaks gambar:

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

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

```
1 | <p></p>
```



Gambar 2.11: Markdown Menampilkan Gambar

* Link

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

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

Contoh penggunaan sintaks link:

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

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

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



Gambar 2.12: Markdown Link

* 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.

```

1 Definisi dari 'initLabels()' dapat dilihat di bawah ini"
2
3 ```
4     function initLabels() {
5         function setLabels(elementId, messageId) {
6             var label = document.querySelector('label[for=' + elementId
7                 + ']');
8             label.textContent = chrome.i18n.getMessage(messageId);
9         }
10    }
11    ```
12 Paragraf lain.
```

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

```

1 <p>Definisi dari 'initLabels()' dapat dilihat di bawah ini"</p>
2
3 <p>```
4     function initLabels() {
5         function setLabels(elementId, messageId) {
6             var label = document.querySelector('label[for=' + elementId
7                 + ']');
8             label.textContent = chrome.i18n.getMessage(messageId);
9         }
10    }
11    ```</p>
12 <p>Paragraf lain.</p>
```

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

```

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

Paragraf lain.

Gambar 2.13: Markdown Kode

* 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 .
```

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

```
1 | <p>Ini merupakan kutipan <strong>favorit </strong> saya:</p>
2 |
3 | <blockquote>
4 |   <p>The weak can never forgive .
5 |   Forgiveness is the attribute of the strong.</p>
6 | </blockquote>
```

Ini merupakan kutipan **favorit** saya:

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

Gambar 2.14: Markdown Kutipan

* 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.

```
1 | # Bagian Satu
2 |
3 | The quick brown fox jumps over the lazy dog. The quick brpwn fox jump
   | over the lazy dog.
4 |
5 | —
6 |
7 | Last Edited on *25th Dec 2014*
```

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

```
1 | <h1>Bagian Satu</h1>
2 |
3 | <p>The quick brown fox jumps over the lazy dog. The quick brpwn fox
   | jump over the lazy dog.</p>
4 |
5 | <hr />
6 |
7 | <p>Last Edited on <em>25th Dec 2014</em></p>
```

Bagian Satu

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

Last Edited on 25th Dec 2014

Gambar 2.15: Markdown Garis Horisontal

* 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.

```

1 * Item
2 * Item
3 * Item
4
5 + Item
6 + Item
7 + Item
8
9 - Item
10 - Item
11 - Item

```

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

```

1 <ul>
2 <li>Item</li>
3 <li>Item</li>
4 <li><p>Item</p></li>
5 <li><p>Item</p></li>
6 <li>Item</li>
7 <li><p>Item</p></li>
8 <li><p>Item</p></li>
9 <li>Item</li>
10 <li>Item</li>
11 </ul>

```


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

Gambar 2.16: Daftar Tidak Berurutan

(2) Daftar berurutan

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

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

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

```
1 | <ol>
2 | <li>Item 1</li>
3 | <li>Item 2</li>
4 | <li>Item 3</li>
5 | </ol>
```

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

Gambar 2.17: Markdown Daftar Berurutan

2.2.3 GitHub Flavored Markdown [?]

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.

* Beberapa Garis Bawah Pada Kalimat

Pada Markdown kata yang berada dianantara garis bawah akan dirubah menjadi cetak

miring, namun pada GFM garis bawah tidak memiliki pasangan maka garis bawah tersebut diabaikan sehingga akan tetap tampil sebagai karakter garis bawah.

```
1|wow_great_stuff
2|do_this_and_do_that_and_another_thing.
```

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

wow_great_stuff do_this_and_do_that_and_another_thing.

Gambar 2.18: GFM Garis Bawah

* 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.

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

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

<http://www.google.com>

Gambar 2.19: GFM Taut Otomatis URL

* 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.

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

Untuk hasil pada HTML dapat dilihat pada Gambar 2.20.

~~Mistaken text.~~

Gambar 2.20: GFM Tanda Coret

* Blok Kode

SM mengkonferensi 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 | '''

```

Untuk hasil pada HTML dapat dilihat pada Gambar 2.21.

Here's an example:

```

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

```

Gambar 2.21: GFM Blok Kode

* Penandaan Sintaks

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

```

1 | '''ruby
2 | require 'redcarpet'
3 | markdown = Redcarpet.new("Hello World!")
4 | puts markdown.to_html
5 | '''

```

Untuk hasil pada HTML dapat dilihat pada Gambar 2.22.

```

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

```

Gambar 2.22: GFM Penandaan Sintaks

* Tabel

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

```

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

```

Untuk hasil pada HTML dapat dilihat pada Gambar 2.23.

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

Gambar 2.23: GFM Tabel 1

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

```

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

```

Untuk hasil pada HTML dapat dilihat pada Gambar 2.24.

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

Gambar 2.24: GFM Tabel 2

Untuk membagi judul tabel dan isi tabel dapat menggunakan tanda hubung (-) yang tidak perlu disesuaikan dengan panjang judul tabel maupun isi tabel. Untuk contoh dapat dilihat di bawah ini.

```

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

```

Untuk hasil pada HTML dapat dilihat pada Gambar 2.25.

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

Gambar 2.25: GFM Tabel 3

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

```

1 | Name | Description |

```

```

2 | | _____ | _____ |
3 | | Help      | ~~Display the~~ help window. |
4 | | Close     | _Closes_ a window      |

```

Untuk hasil pada HTML dapat dilihat pada Gambar 2.26.

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

Gambar 2.26: GFM Tabel 4

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

```

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

```

Untuk hasil pada HTML dapat dilihat pada Gambar 2.27.

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

Gambar 2.27: GFM Tabel 5

2.3 StrapdownJS [?]

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>

```

* Cara kedua:

```
1|<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.

```
1|<xmp theme="united" style="display:none;">
2|
3|</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:

```
1|<!DOCTYPE html>
2|<html>
3|<title>Hello Strapdown</title>
4|
5|<xmp theme="united" style="display:none;">
6|# Markdown text goes in here
7|
8|## Chapter 1
9|
10|Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
11|    tempor incididunt ut labore
12|    et dolore magna aliqua.
13|
14|## Chapter 2
15|
16|Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut
17|    aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in
18|    voluptate velit esse
19|    cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat
20|    non proident, sunt in
21|    culpa qui officia deserunt mollit anim id est laborum.
22|</xmp>
23|
24|<script src="http://strapdownjs.com/v/0.2/strapdown.js"></script>
25|</html>
```

Strapdown.js juga memiliki beberapa fitur :

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

Tabel 2.1: Daftar Pengujian Zurb Foundation¹

<i>Browser/OS</i>	<i>The Grid</i>	<i>Layout/UI</i>	<i>JS</i>
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 [?]

Zurb Foundation merupakan alat bantu dalam membuat aplikasi baru maupun membuat *website* yang responsif, tampilan antarmuka mengikuti lebar mesin pencari dan/atau layar komputer tanpa melakukan permintaan tambahan ke server. Jutaan desainer dan teknisi menggunakan Foundation sebagai bagian dari alur kerja mereka. Zurb Foundation adalah *framework* pertama yang memperkenalkan konsep responsif, semantik, mobile dan parsial. Zurb Foundation juga kompatibel dengan kebanyakan mesin pencari dan perangkat. Maka dari itu Zurb Foundation merupakan pilihan profesional bagi para desainer dan teknisi.

2.4.1 Kompatibilitas

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

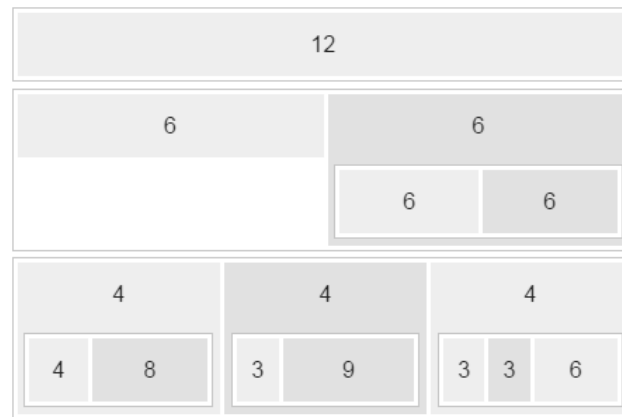
2.4.2 Apa Saja yang Dapat Dibuat dengan Foundation?

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

(1) *Grid*

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

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

Gambar 2.28: Contoh *Grid* Zurb Foundation

* *Grid* Utama

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

```

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

```



```

26 <div class="large-4 columns">full </div>
27 <div class="large-8 columns">full </div>
28 </div>
29 <!-- no. 6 -->
30 <div class="row">
31 <div class="small-6 large-5 columns">6</div>
32 <div class="small-6 large-7 columns">6</div>
33 </div>
34 <!-- no. 7 -->
35 <div class="row">
36 <div class="large-6 columns">full </div>
37 <div class="large-6 columns">full </div>
38 </div>

```

Gambar 2.29: *Grid* Utama* *Grid* Kecil

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

```

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

```

Gambar 2.30: *Grid* Kecil* *Grid* Menengah

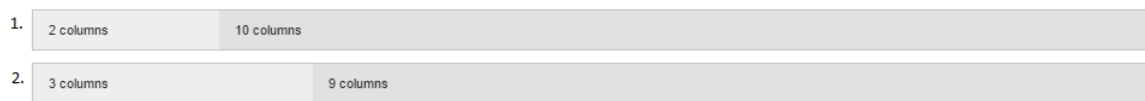
Layar berukuran sedang akan mendapatkan bagian dari gaya yang kecil, kecuali

ditentukan tata letak yang berbeda dengan menggunakan *grid* menengah. Berikut kode HTML untuk membuat *grid* menengah dan dapat dilihat hasilnya pada Gambar 2.31.

```

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

```



Gambar 2.31: *Grid* Menengah

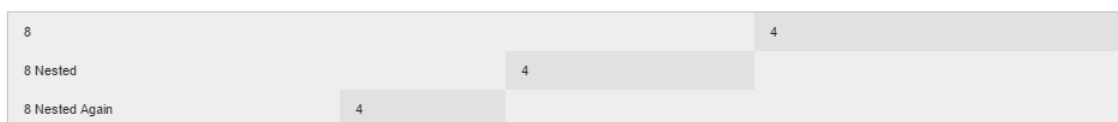
* *Grid* Tingkat Lanjut

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

```

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

```



Gambar 2.32: *Grid* Tingkat Lanjut

* *Grid* Dengan Offset

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

```

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

```

Gambar 2.33: *Grid* Dengan Offset

* *Grid* Dengan Baris yang Tidak Lengkap

Untuk mengatasi perbedaan pada beberapa mesin pencari, Foundation akan memindahkan kolom terakhir berturut-turut ke kanan sehingga sejajar dengan tepi. Jika jumlah kolom tidak mencapai 12 maka pada kolom terakhir perlu ditandai dengan kelas *end*. Berikut contoh penggunaan pada kode HTML dan dapat dilihat hasilnya pada Gambar 2.34.

```

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

```

1.	3	3		3
2.	3	3	3 end	

Gambar 2.34: *Grid* Dengan Baris yang Tidak Lengkap

item *Grid* Dengan Baris yang Mengempis/Tidak Mengempis

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

```

1 <div class="row medium-uncollapse large-collapse">
2 <div class="small-6 columns">
3 Removes gutter at large media query
4 </div>
5 <div class="small-6 columns">
6 Removes gutter at large media query
7 </div>
8 </div>

```

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

Gambar 2.35: *Grid* Dengan Baris yang Mengempis/Tidak Mengempis

* *Grid* Dengan Kolom yang Berpusat

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

```

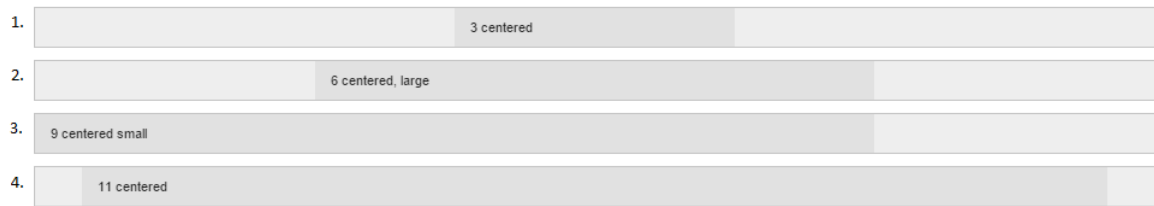
1 <!-- no. 1 -->
2 <div class="row">
3 <div class="small-3 small-centered columns">3 centered</div>
4 </div>
5 <!-- no. 2 -->
6 <div class="row">
7 <div class="small-6 large-centered columns">6 centered</div>
8 </div>
9 <!-- no. 3 -->
10 <div class="row">
11 <div class="small-9 small-centered large-uncentered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

```

```

14 <div class="row">
15 <div class="small-11 small-centered columns">11 centered </div>
16 </div>

```

Gambar 2.36: *Grid* Dengan Kolom yang Berpusat

* *Grid* Dengan Sumber yang Terurut

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

```

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

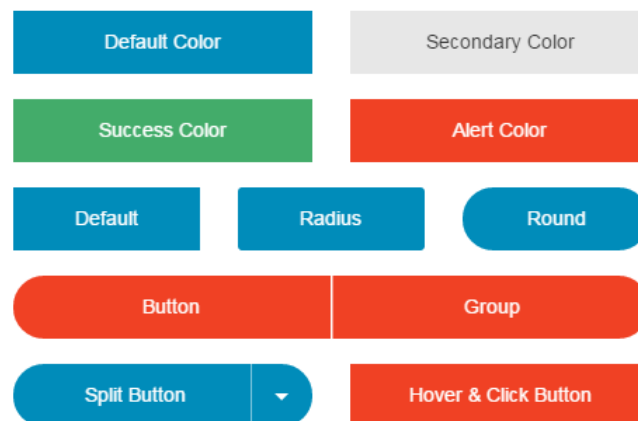
```

1.	2, last	10
2.	3, last	9
3.	4, last	8
4.	5, last	7
5.	6, last	6

Gambar 2.37: *Grid* Dengan Sumber yang Terurut

(2) Tombol

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

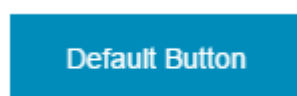


Gambar 2.38: Contoh Tombol Zurb Foundation

* Tombol Utama

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

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



Gambar 2.39: Tombol Dasar

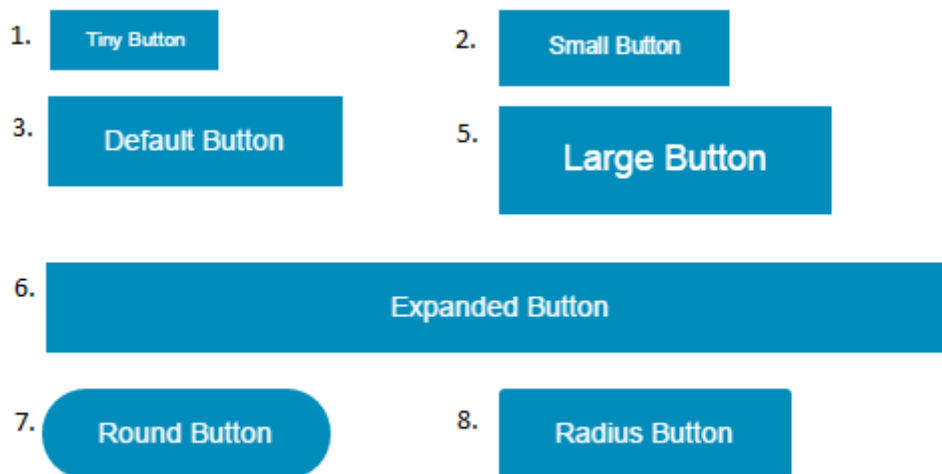
* Mengubah Ukuran Tombol

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

```

1 <!-- Kelas untuk merubah ukuran -->
2 1. <a href="#" class="button tiny">Tiny Button</a>
3 2. <a href="#" class="button small">Small Button</a>
4 3. <a href="#" class="button">Default Button</a>
5 4. <a href="#" class="button disabled">Disabled Button</a>
6 5. <a href="#" class="button large">Large Button</a>
7 6. <a href="#" class="button expand">Expanded Button</a>
8 <!-- Kelas untuk merubah bentuk -->
9 7. <a href="#" class="button round">Round Button</a>
10 8. <a href="#" class="button radius">Radius Button</a>

```



Gambar 2.40: Ukuran dan Bentuk Tombol

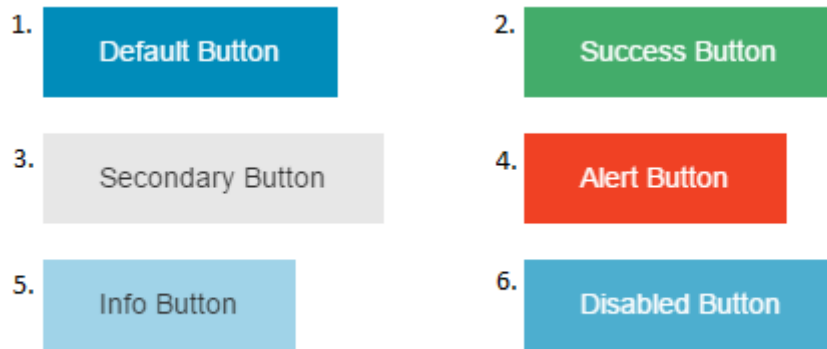
* Warna Tombol

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

```

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

```



Gambar 2.41: Warna-warna Tombol

* Aksesibilitas

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

```

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

```



Gambar 2.42: Aksesibilitas Tombol

(3) Tabel

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

```

1 | <table>
2 |   <thead>
3 |     <tr>
4 |       <th width="200">Table Header</th>
5 |       <th>Table Header</th>
6 |       <th width="150">Table Header</th>
7 |       <th width="150">Table Header</th>
8 |     </tr>
9 |   </thead>
10 |  <tbody>
11 |    <tr>
12 |      <td>Content Goes Here</td>

```



```

13      <td>This is longer content Donec id elit non mi porta gravida at
      eget metus.</td>
14      <td>Content Goes Here</td>
15      <td>Content Goes Here</td>
16  </tr>
17  <tr>
18      <td>Content Goes Here</td>
19      <td>This is longer Content Goes Here Donec id elit non mi porta
      gravida at eget metus.</td>
20      <td>Content Goes Here</td>
21      <td>Content Goes Here</td>
22  </tr>
23  <tr>
24      <td>Content Goes Here</td>
25      <td>This is longer Content Goes Here Donec id elit non mi porta
      gravida at eget metus.</td>
26      <td>Content Goes Here</td>
27      <td>Content Goes Here</td>
28  </tr>
29  </tbody>
30 </table>

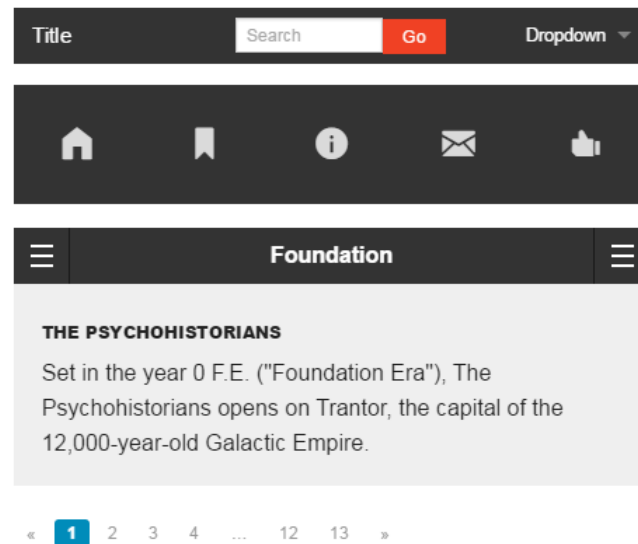
```

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

Gambar 2.43: Tabel Foundation

(4) Navigasi

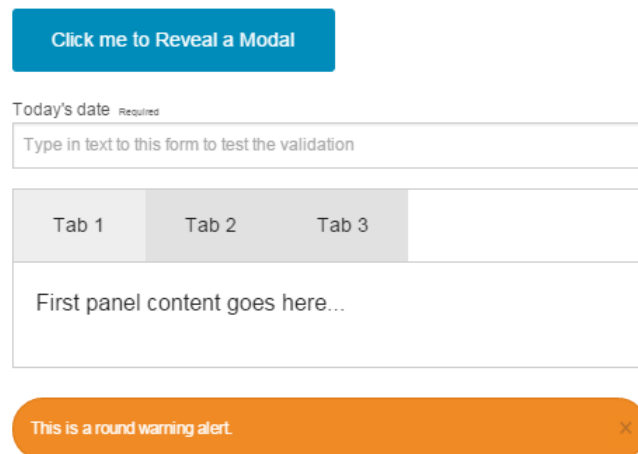
Orang yang mengakses harus bisa berkeliling melihat menu-menu yang ada. Gaya navigasi pada Foundation meliputi : bar bagian atas yang kuat dengan menu dropdown; tombol; bar pencari; ikon bar yang keren; implementasi kanvas yang lepas dari keluhan; dan sekelompok navigasi lainnya. Untuk contoh macam-macam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



Gambar 2.44: Contoh Navigasi Zurb Foundation

(5) *Plugins*

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



Gambar 2.45: Contoh Plugins Zurb Foundation

BAB 3

ANALISIS

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

3.1 Analisis Kebutuhan Pengguna

Kegiatan pengumpulan data merupakan salah satu tahap yang berperan penting dalam sebuah penelitian. Pada penelitian ini digunakan teknik pengumpulan data berupa kuesioner. Kuesioner yang disebarkan sebanyak 6 kuesioner dan yang menjadi responden adalah dosen Falkutas Teknik Informasi dan Sains. Untuk kuesioner kebutuhan pengguna dapat dilihat pada Lampiran ???. Hasil kuesioner dari 6 orang dosen tersebut menghasilkan data yang dapat dilihat pada Tabel 3.1-3.8.

Tabel 3.1: Jawaban Pertanyaan Pertama Survei Kebutuhan Pengguna

Dosen	Sangat Setuju	Setuju	Netral	Tidak Setuju	Sangat Tidak Setuju
PAS		✓			
RDL		✓			
VSM			✓		
CAN	✓				
VAN		✓			
LNV		✓			

Tabel 3.2: Jawaban Pertanyaan Kedua Survei Kebutuhan Pengguna

Dosen	Sangat Setuju	Setuju	Netral	Tidak Setuju	Sangat Tidak Setuju
PAS		✓			
RDL			✓		
VSM			✓		
CAN		✓			
VAN		✓			
LNV			✓		

Tabel 3.3: Jawaban Pertanyaan Ketiga Survei Kebutuhan Pengguna

Dosen	Sangat Setuju	Setuju	Netral	Tidak Setuju	Sangat Tidak Setuju
PAS			✓		
RDL			✓		
VSM			✓		
CAN		✓			
VAN			✓		
LNV				✓	

3.2 Analisis Google OAuth

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

3.2.1 Langkah Dasar Penggunaan OAuth 2.0

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

- (1) Mendapatkan kepercayaan OAuth 2.0 dari Google Developers Console
 - (a) Mengunjungi Google Developers Console. Agar lebih jelas dapat dilihat pada Gambar 3.1.
 - (b) Buat sebuah proyek baru. Dapat dilihat pada Gambar 3.2.
 - (c) Masuk ke proyek yang telah dibuat dan masuk ke menu '*Credentials*'. Agar lebih jelas dapat dilihat pada Gambar 3.3.
 - (d) Membuat *client id* yang baru. Dapat dilihat pada Gambar 3.4.
 - (e) Pilih tipe aplikasi sesuai aplikasi yang dibangun, pada penelitian ini menggunakan tipe aplikasi web karena aplikasi yang akan dibangun berbasis web. Agar lebih jelas dapat dilihat pada Gambar 3.5.
 - (f) Isi bagian AUTHORIZED JAVASCRIPT ORIGINS (merupakan path dimana javascript otorisasi akan dijalankan) pada penelitian ini bagian AUTHORIZED JAVASCRIPT ORIGINS akan diisi dengan `http://localhost/` karena aplikasi yang akan dibangun pada penelitian ini terletak pada localhost dan AUTHORIZED

Tabel 3.4: Jawaban Pertanyaan Keempat Survei Kebutuhan Pengguna

Dosen	Sangat Setuju	Setuju	Netral	Tidak Setuju	Sangat Tidak Setuju
PAS			✓		
RDL		✓			
VSM		✓			
CAN		✓			
VAN		✓			
LNV	✓				

Tabel 3.5: Jawaban Pertanyaan Kelima Survei Kebutuhan Pengguna

Dosen	Sangat Setuju	Setuju	Netral	Tidak Setuju	Sangat Tidak Setuju
PAS				✓	
RDL				✓	
VSM			✓		
CAN			✓		
VAN					✓
LNV		✓			

REDIRECT URIS (merupakan pengarah jika otorisasi sudah berhasil) pada penelitian ini bagian AUTHORIZED REDIRECT URIS akan diisi dengan

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

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

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

```
1 | Client id:
2 | 568951368854-ufmbistn0pcaq0khubafo1a133orfgve.apps.googleusercontent
   | .com
3 | Client secret:
4 | -cSZ-AUmeQ9PaWWry_IpiBBi
```

Agar lebih jelas dapat dilihat pada Gambar 3.7.

- (2) Memperoleh token akses dari Google Authorization Server

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

- (3) Kirim token ke API

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

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

Tabel 3.6: Jawaban Pertanyaan Keenam Survei Kebutuhan Pengguna

Dosen	Sangat Setuju	Setuju	Netral	Tidak Setuju	Sangat Tidak Setuju
PAS	✓				
RDL	✓				
VSM		✓			
CAN	✓				
VAN		✓			
LNV	✓				

Tabel 3.7: Jawaban Pertanyaan Ketujuh Survei Kebutuhan Pengguna

Dosen	Sangat Setuju	Setuju	Netral	Tidak Setuju	Sangat Tidak Setuju
PAS		✓			
RDL	✓				
VSM			✓		
CAN	✓				
VAN		✓			
LNV			✓		

2 | dan
3 | email

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

(4) Memperbaharui token akses jika diperlukan

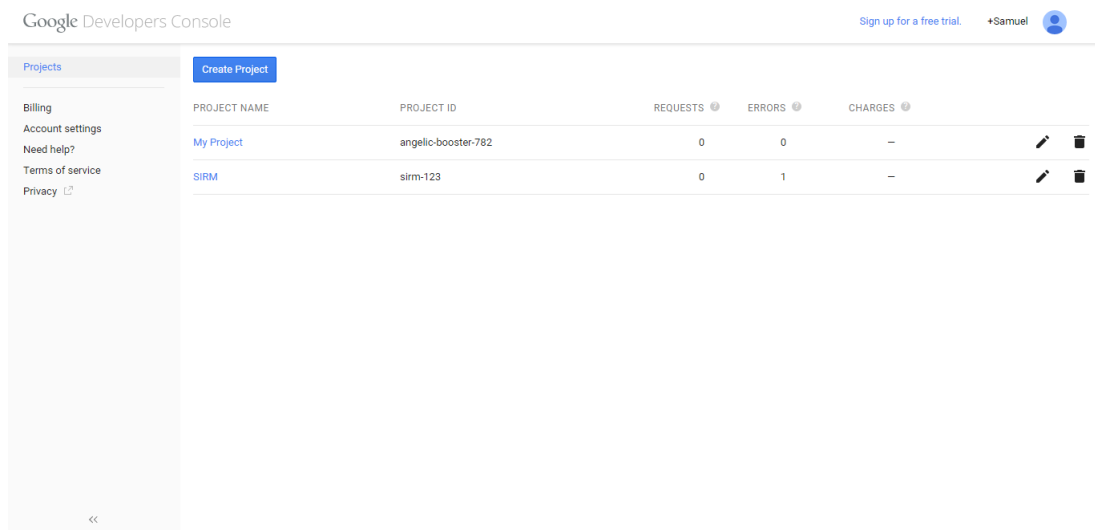
Pada penelitian ini tidak akan menggunakan tahap memperbaharui token akses karena token akses hanya digunakan selama penelitian ini berlangsung.

3.2.2 Skenario Aplikasi

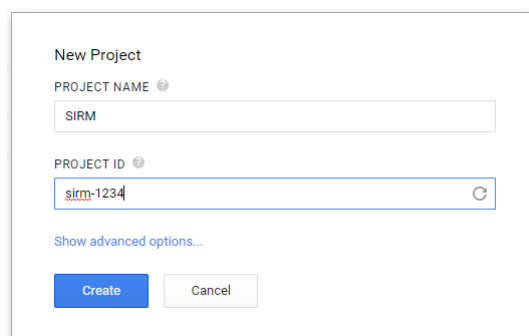
Berdasarkan landasan teori skenario yang ada pada Sub Bab 2.1 dan berdasarkan perangkat lunak yang akan dibangun, maka skenario yang akan digunakan pada penelitian ini adalah skenario aplikasi web server. Aplikasi SIRM akan melakukan permintaan token ke Server Google. Dosen sebagai pengguna akan melakukan 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.

3.3 Analisis Markdown

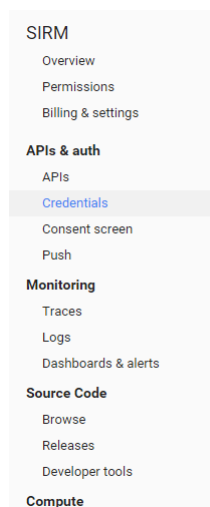
Sintaks Markdown yang akan digunakan sesuai dengan landasan teori pada Sub Bab 2.2. Sintaks Markdown akan digunakan pada bagian keterangan mahasiswa agar seragam



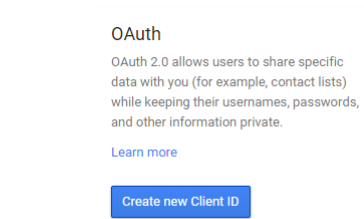
Gambar 3.1: Google Developers Console



Gambar 3.2: Membuat Proyek Baru



Gambar 3.3: Menu Credentials



Gambar 3.4: Membuat Client ID yang Baru

This is a screenshot of the 'Create Client ID' dialog box. It has a title 'Create Client ID'. Under 'APPLICATION TYPE', there are three radio buttons: 'Web application' (selected), 'Service account', and 'Installed application'. Each has a brief description. Below this is the 'AUTHORIZED JAVASCRIPT ORIGINS' section with a text box containing 'https://www.example.com'. The 'AUTHORIZED REDIRECT URIS' section has a text box containing 'https://www.example.com/oauth2callback'. At the bottom are 'Create Client ID' and 'Cancel' buttons.

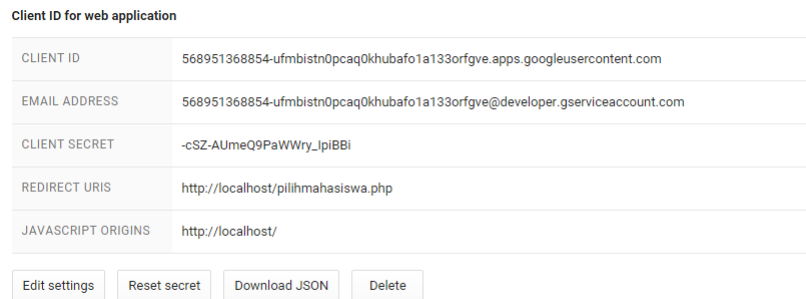
Gambar 3.5: Tipe Aplikasi

This screenshot shows the 'Create Client ID' dialog box with the 'Web application' type selected. The 'Authorized JavaScript origins' text box now contains 'http://localhost/'. The 'Authorized redirect URIs' text box now contains 'http://localhost/oauth.php'. The 'Create Client ID' and 'Cancel' buttons remain at the bottom.

Gambar 3.6: Pengisian Tipe Aplikasi

Tabel 3.8: Jawaban Pertanyaan Kedelapan Survei Kebutuhan Pengguna

Dosen	Sangat Setuju	Setuju	Netral	Tidak Setuju	Sangat Tidak Setuju
PAS		✓			
RDL				✓	
VSM		✓			
CAN				✓	
VAN		✓			
LNV	✓				



Gambar 3.7: Client ID

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.

```
1 | **NPM** — *2010730013*
```

* Sintaks Judul Bab

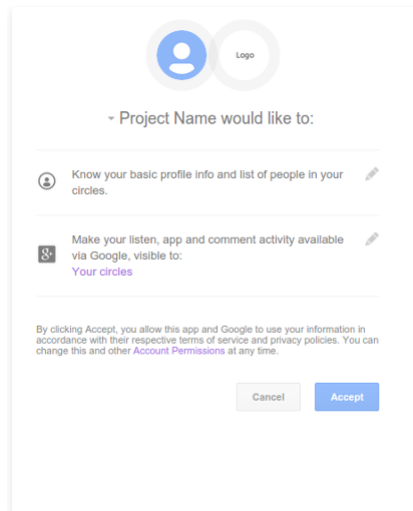
Sintaks ini akan digunakan untuk menampilkan judul setiap bagian (NPM, nama, umum, dan catatan). Berikut penggunaan sintaks dan hasilnya dapat dilihat pada Gambar 3.11.

```
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.

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



Gambar 3.8: Izin Pihak Pengguna

```

3 |
4 | Baris ini tanpa
5 | batas baris

```

* Sintaks Paragraf

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

```

1 | Samuel adalah seorang mahasiswa yang periang namun terkadang sulit
   |      diatur . Dia aktif di himpunan sebagai ketua divisi pelayanan
   |      masyarakat .
2 |
3 | Grady adalah seorang mahasiswa yang memiliki jiwa pemimpin . Dia aktif
   |      di UKM sebagai 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.

```

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

```

* Daftar

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

```

1 | * 9 Oktober 2014, bimbingan skripsi
2 | * 3 Oktober 2014, bimbingan skripsi
3 | * 1 September 2014, perwalian
4 | * 1 September 2014, pertama kali dibuat

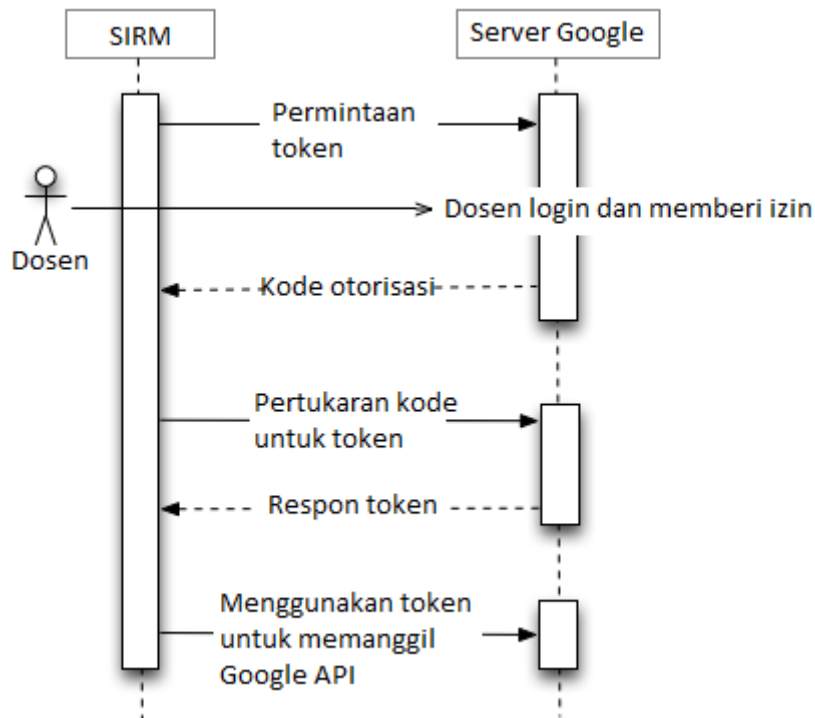
```

Berikut penggunaan sintaks Markdown secara keseluruhan untuk bagian keterangan mahasiswa. Berikut penggunaan sintaks dan hasilnya dapat dilihat pada Gambar

```

1 | #### NPM

```



Gambar 3.9: Skenario Aplikasi SIRM

```

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

3.4 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.

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

NPM

2010730013

Nama

Samuel

Umum

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

Catatan

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

Gambar 3.16: Output Keterangan Mahasiswa

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

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

```
1 <!doctype html>
2 <html class="no-js" lang="en">
3   <head>
4     <meta charset="utf-8" />
5     <meta name="viewport" content="width=device-width, initial-scale
      =1.0" />
6     <title>SIRM | Welcome</title>
7     <link rel="stylesheet" href="css/foundation.css" />
8     <script src="js/vendor/modernizr.js"></script>
9   </head>
10  <body>
11    <div class="row">
12      <h5>Anda melihat catatan mahasiswa ini sebagai test@unpar.ac.id
        .</h5>
13    </div>
14    <div class="row">
15      <ul class="button-group">
16        <li><a href="editmahasiswa.php" class="button">Edit</a></li>
17        <li><a href="lihathistori.php" class="button">Lihat Histori
          </a></li>
18      </ul>
19    </div>
20    <hr/>
21    <xmp style="display:none;">
22    #### NPM
23
24    2010730013
25
26    #### Nama
27
28    Samuel
```

```

29 |
30 | ### Umm
31 |
32 | 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://bletrack.blogspot.com/](http://
   |     bletrack.blogspot.com/).
33 |
34 | ### Catatan
35 |
36 | * 9 Oktober 2014, bimbingan skripsi
37 | * 3 Oktober 2014, bimbingan skripsi
38 | * 1 September 2014, perwalian
39 | * 1 September 2014, pertama kali dibuat
40 |
41 | </xmp>
42 |     <script src="js/0.2/strapdown.js"></script>
43 | </body>
44 | </html>

```

Untuk baris 22 sampai baris 40 pada skrip view.php akan diambil dari database.

3.5 Analisis Zurb Foundation

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

```

1 | <!doctype html>
2 | <html class="no-js" lang="en">
3 |     <head>
4 |         <meta charset="utf-8" />
5 |         <meta name="viewport" content="width=device-width, initial-scale
   |             =1.0" />
6 |         <title>SIRM | List</title>
7 |         <link rel="stylesheet" href="css/foundation.css" />
8 |         <script src="js/vendor/modernizr.js"></script>
9 |     </head>
10 |     <body>
11 |         <?php
12 |             session_start();
13 |         ?>
14 |         <div class="row">
15 |             <h3>Pilih NPM yang ingin dicari / tambah baru.</h3>
16 |             <ul class="button-group">
17 |                 <li><a href="new.php" class="button secondary">Add</a></li>
18 |                 <li><a href="index.php?logout" class="button secondary">
   |                     Logout</a></li>

```

```

19         </ul>
20         <hr/>
21     </div>
22
23     <div class="row">
24         <?php
25             include_once "configDatabase.php";
26
27             if (! $id_mysql)
28             {
29                 die("Database tidak bisa dibuka");
30             }
31
32             if (! mysql_select_db("sirm", $id_mysql))
33             {
34                 die("Database tidak bisa dipilih");
35             }
36
37             $hasil = mysql_query("SELECT * FROM info_mahasiswa",
38                                 $id_mysql);
39
40             if (! $hasil)
41             {
42                 die("Permintaan gagal");
43             }
44
45             echo "<table>
46             <thead>
47             <tr>
48             <th width='250'>NPM</th>
49             <th width='500'>Nama</th>
50             <th width='250'>Last Update</th>
51             </tr>
52             </thead>";
53
54             while($row = mysql_fetch_array($hasil))
55             {
56                 echo "<tr>";
57                 echo "<td><a href='view.php?npm=". $row['npm'] ."'>" . $row
58                     ['npm'] . "</a></td>";
59                 echo "<td>" . $row['nama'] . "</td>";
60                 echo "<td>" . $row['pembaruan_terakhir'] . "</td>";
61                 echo "</tr>";
62             }
63             echo "</table>";
64         ?>
65     </div>
66 </body>
67 </html>

```

Pilih NPM yang ingin dicari / tambah baru.

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

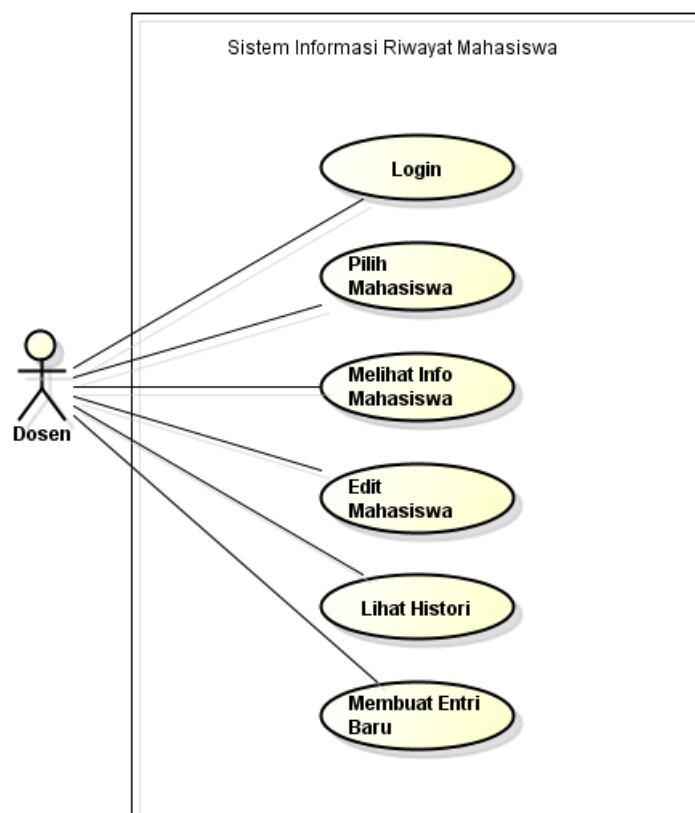
NPM	Nama	Last Update
2003730013	Pascal	2015-05-05 09:26:15
2010120031	Kenneth Natanael	2015-04-30 14:29:51
2010730001	Andri Agustian	2015-04-29 15:09:33
2010730005	Grady Ireneus	2015-04-30 14:31:19
2010730012	Kevin PL	2015-04-24 00:55:57
2010730013	Samuel Herman	2015-05-08 00:37:01

Gambar 3.17: Tampilan pilihmahasiswa.php dengan Zurb Foundation

3.6 Analisis Kebutuhan Perangkat Lunak

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

3.6.1 Use Case Diagram



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

Use case diagram merupakan pemodelan yang menunjukkan kegiatan apa saja yang dapat dilakukan pengguna dan kegiatan yang dilakukan sistem. Berikut adalah deskripsi dari *use 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 menekan tombol "*Add*" untuk menambah entri baru.

* Melihat Info Mahasiswa

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 mengubah info mahasiswa yang sudah ada.

* Lihat Histori

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

* Membuat Entri Baru

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

3.6.2 Skenario

Skenario untuk menjelaskan *use case* yang terpadat pada Gambar 3.18 dapat dilihat pada daftar di bawah ini.

- * Untuk *use case* Login, skenarionya dapat dilihat pada Tabel 3.9.

Tabel 3.9: 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		

- * Untuk *use case* Pilih Mahasiswa, skenarionya dapat dilihat pada Tabel 3.10.

- * Untuk *use case* Melihat Info Mahasiswa, skenarionya dapat dilihat pada Tabel 3.11.

Tabel 3.10: 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 3.11: Skenario Melihat Info Mahasiswa

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

- * Untuk *use case* Edit Mahasiswa, skenarionya dapat dilihat pada Tabel 3.12.
- * Untuk *use case* Lihat Histori, skenarionya dapat dilihat pada Tabel 3.13.
- * Untuk *use case* Membuat Entri Baru, skenarionya dapat dilihat pada Tabel 3.14.

3.6.3 Entity Relationship Diagram (ERD)

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

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

Tabel 3.12: 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 3.13: 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	-		

3.6.4 Data Flow Diagram

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

(1) Data Context Diagram

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

(2) Data Flow Diagram Level 1

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

Untuk spesifikasi proses dapat dilihat di bawah ini.

* No dan Nama Proses : P1 Login

Deskripsi : Proses akses ke perangkat lunak

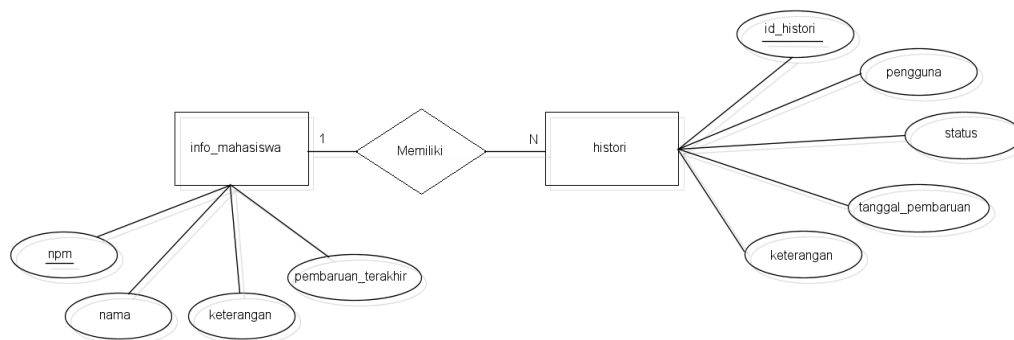
Data Input : Email, password

Data Output : Login valid

Proses : Mengecek email yang digunakan diakhiri @unpar.ac.id atau tidak. Jika

Tabel 3.14: 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 merekam inputan pengguna ke dalam database
Eksepsi	-		

Gambar 3.19: *Entity Relationship* Diagram

ya maka sistem akan mengarahkan ke halaman utama akan tetapi jika tidak maka sistem akan menampilkan pesan login gagal.

* No dan Nama Proses : P2 Pilih Mahasiswa

Deskripsi : Proses memilih mahasiswa

Data Input : Data mahasiswa

Data Output : Tabel daftar mahasiswa

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

* No dan Nama Proses : P3 Melihat Info Mahasiswa

Deskripsi : Proses melihat info mahasiswa

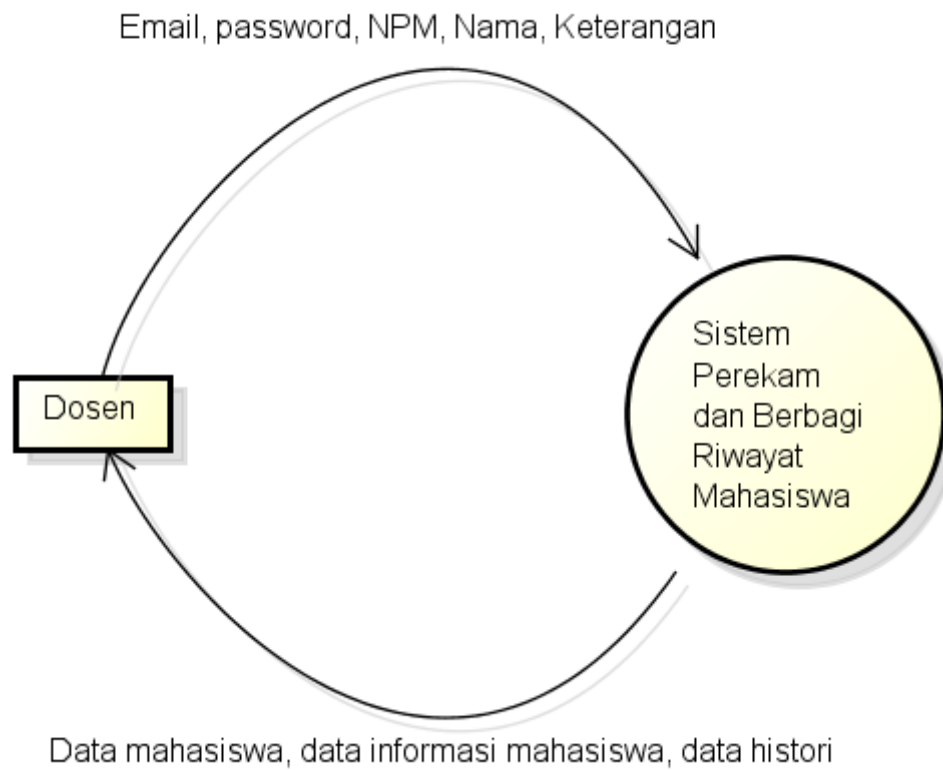
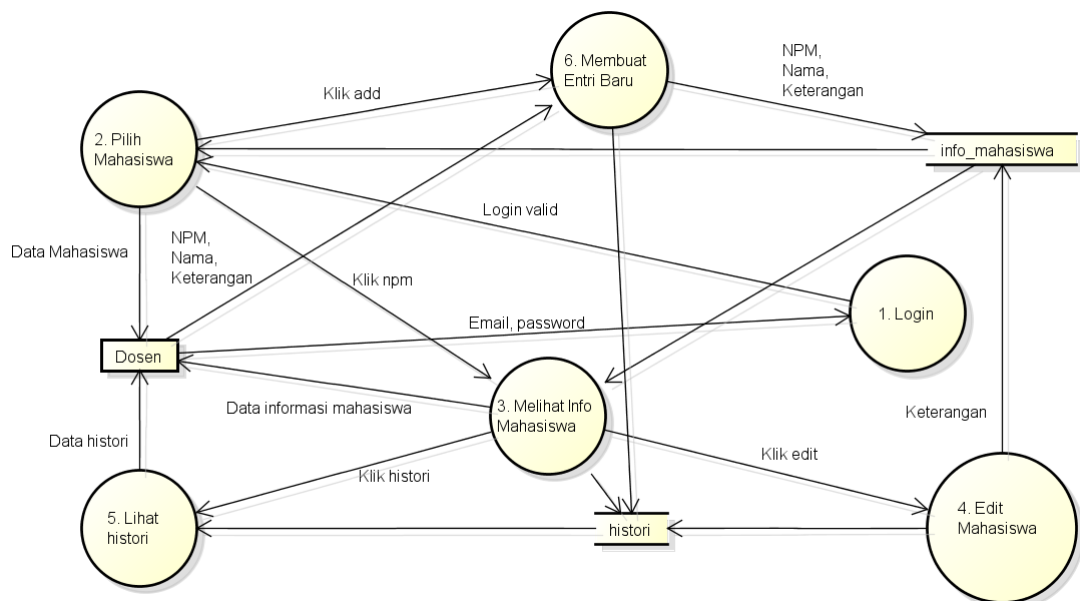
Data Input : Data informasi mahasiswa

Data Output : Data histori

Proses : pengguna mendapatkan info mahasiswa dari mahasiswa yang telah pengguna pilih pada proses 2. Proses ini dicatat pada histori sebagai aksi melihat.

* No dan Nama Proses : P4 Edit Mahasiswa

Deskripsi : Proses mengedit info mahasiswa

Gambar 3.20: Data *Context* Diagram Sistem Informasi Riwayat MahasiswaGambar 3.21: Data *Flow* Diagram Sistem Informasi Riwayat Mahasiswa

Data Input : Keterangan mahasiswa terbaru

Data Output : Data informasi mahasiswa dan data histori

Proses : pengguna memperbaharui keterangan yang dimiliki mahasiswa yang telah pengguna pilih pada proses 2. Proses ini dicatat pada histori sebagai aksi mengedit.

* No dan Nama Proses : P5 Lihat Histori

Deskripsi : Proses melihat histori

Data Input : Aksi pengguna dan keterangan

Data Output : Daftar histori

Proses : pengguna mendapatkan histori dari mahasiswa yang telah pengguna pilih pada proses 2. Pengguna juga dapat melihat informasi mahasiswa versi sebelumnya.

* No dan Nama Proses : P6 Membuat Entri Baru

Deskripsi : Proses membuat entri baru

Data Input : NPM, Nama, Keterangan

Data Output : Data informasi mahasiswa dan data histori

Proses : pengguna memasukan data NPM, Nama, Keterangan untuk entri baru lalu menyimpannya. Proses ini dicatat pada histori sebagai aksi membuat entri baru.

BAB 4

PERANCANGAN

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

4.1 Perancangan Tampilan *Web* Yang Digunakan

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

4.1.1 Tampilan Halaman Awal

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

1. **WELCOME to SIRM**
(Sistem Informasi Riwayat Mahasiswa)
 2. [Login with Google](#)
-

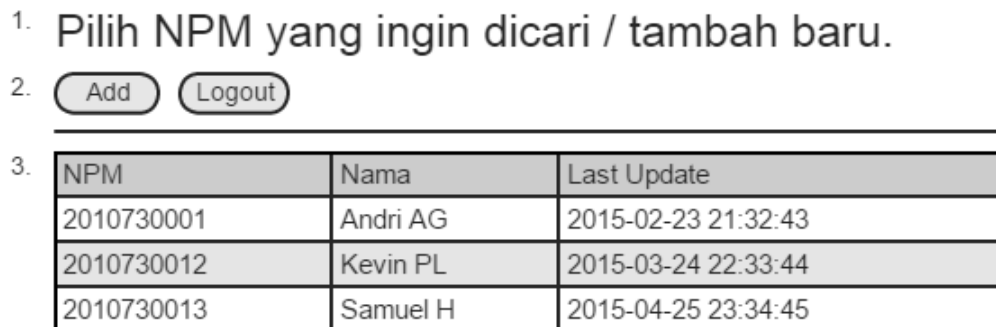
Gambar 4.1: Desain Antarmuka Halaman Awal

Keterangan :

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

4.1.2 Tampilan Web Pilih Mahasiswa

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



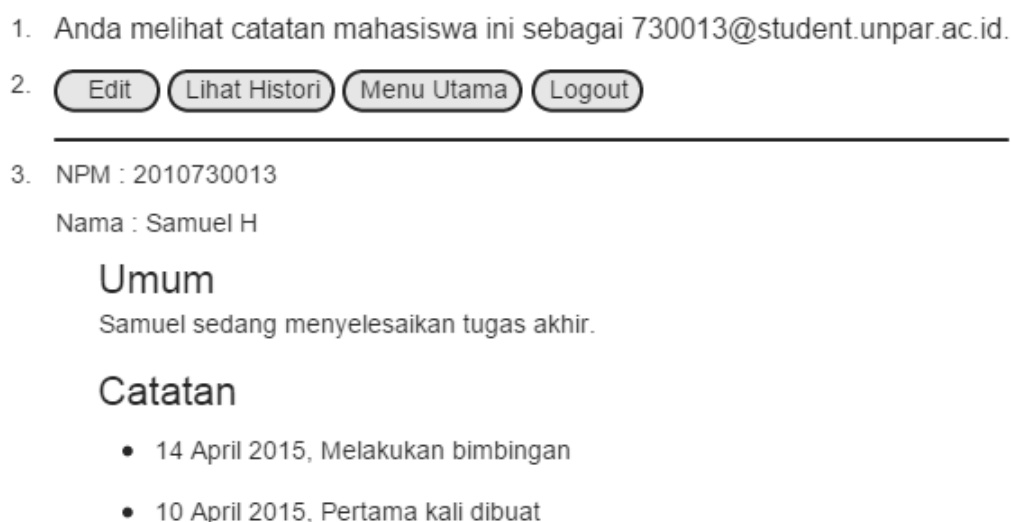
Gambar 4.2: Desain Antarmuka Pilih Mahasiswa

Keterangan :

- (1) Bagian ini merupakan judul dari halaman untuk memilih mahasiswa.
- (2) Bagian ini merupakan tombol untuk melakukan aksi add atau logout.
- (3) Bagian ini merupakan tempat menampilkan data mahasiswa dalam bentuk tabel. NPM dapat diklik untuk memilih mahasiswa.

4.1.3 Tampilan Web Info Mahasiswa

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



Gambar 4.3: Desain Antarmuka Info Mahasiswa


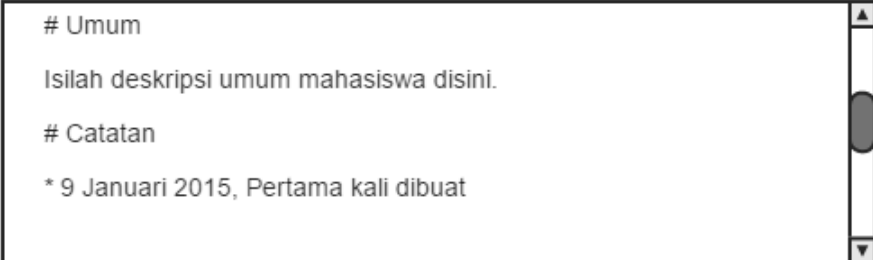
Keterangan :

- (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, lihat histori, pindah ke menu utama, dan logout.
- (3) Bagian ini merupakan tempat menampilkan info mahasiswa yang berasal dari database.

4.1.4 Tampilan *Web* Edit Mahasiswa

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

1. Anda mengedit catatan mahasiswa ini sebagai 730013@student.unpar.ac.id.
2. NPM 2010730013 Nama Samuel H
3. 
4. 

Gambar 4.4: Desain Antarmuka Edit Mahasiswa

Keterangan :

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

4.1.5 Tampilan *Web* Lihat Histori

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

Keterangan :

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

1. Kembali Menu Utama Logout
2. NPM : 2010730013 Nama : Samuel H

3.
 - 2015-05-02 20:21:23 7310013@student.unpar.ac.id melihat 2010730013
 - 2015-04-22 13:31:13 7310013@student.unpar.ac.id mengedit 2010730013
 - 4. [\[lihat versi ini\]](#)
 - 2015-04-17 10:11:12 7310013@student.unpar.ac.id melihat 2010730013
 - 2015-04-10 08:09:10 7310013@student.unpar.ac.id membuat entri 2010730013
 - [\[lihat versi ini\]](#)

Gambar 4.5: Desain Antarmuka Lihat Histori

4.1.6 Tampilan Web Lihat Versi Ini

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

1. Kembali Logout

2. Umum

Samuel sedang menyelesaikan tugas akhir.

Catatan

 - 10 April 2015, Pertama kali dibuat

Gambar 4.6: Desain Antarmuka Lihat Versi Ini

Keterangan :

- (1) Bagian ini merupakan tombol untuk melakukan aksi kembali dan logout.
- (2) Bagian ini merupakan daftar histori dari mahasiswa yang telah dipilih.

4.1.7 Tampilan Web Entri Baru

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

Keterangan :

- (1) Bagian ini merupakan teks yang menampilkan keterangan dan juga pengguna yang sedang menggunakan Sistem Infomasi Riwayat Mahasiswa.
- (2) Bagian ini merupakan tombol untuk melakukan aksi kembali ke pilih mahasiswa, simpan, menu utama, dan logout.
- (3) Bagian ini merupakan form yang terdiri dari area untuk memasukkan NPM mahasiswa, nama mahasiswa, dan keterangan mahasiswa yang akan ditambah dengan format yang telah disediakan (ditulis dengan format markdown).

- Anda membuat catatan mahasiswa ini sebagai 730013@student.unpar.ac.id.
-
- NPM
 Nama

Umum

Isilah deskripsi umum mahasiswa disini.

Catatan

* 9 Januari 2015, Pertama kali dibuat

Gambar 4.7: Desain Antarmuka Pilih Entri Baru

4.2 Perancangan Modul

Perancangan modul untuk sistem informasi riwayat mahasiswa yang akan dibuat dapat dilihat pada sub bab berikut.

4.2.1 Modul Login

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

Tabel 4.1: Modul Login

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

4.2.2 Modul Pilih Mahasiswa

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

4.2.3 Modul Info Mahasiswa

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

Tabel 4.2: Modul Pilih Mahasiswa

Nama Modul	list.php
Input	npm
Output	Tabel mahasiswa
Tabel yang diakses	InfoMahasiswa
Deskripsi	Pengguna memilih npm yang ingin dicari sebagai input yang akan diteruskan ke modul info mahasiswa dan pengguna juga dapat membuat entri baru.

Tabel 4.3: Modul Info Mahasiswa

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

4.2.4 Modul Edit Mahasiswa

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

Tabel 4.4: Modul *Edit* Mahasiswa

Nama Modul	edit.php
Input	teks dalam format markdown
Output	-
Tabel yang diakses	InfoMahasiswa dan Histori
Deskripsi	Pengguna memasukkan atau merubah keterangan mahasiswa pada teks area yang telah disediakan menggunakan teks dengan sintaks Markdown lalu pengguna menyimpan untuk menaruh perubahan yang dilakukan. Pengguna dapat kembali ke modul info mahasiswa tanpa melakukan perubahan.

4.2.5 Modul Lihat Histori

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

4.2.6 Modul Entri Baru

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

Tabel 4.5: Modul Lihat Histori

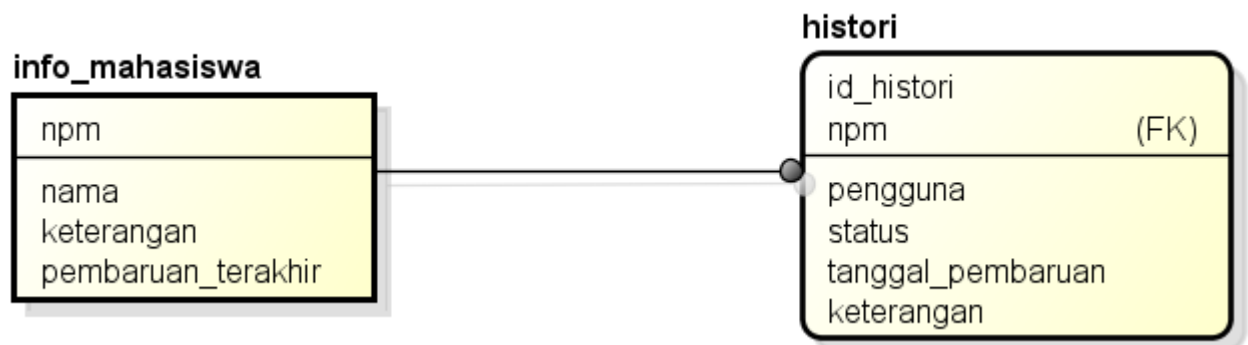
Nama Modul	history.php
Input	-
Output	Daftar histori mahasiswa
Tabel yang diakses	Histori
Deskripsi	Pengguna mendapatkan laporan berupa daftar histori yang dimiliki setiap mahasiswa.

Tabel 4.6: Modul Entri Baru

Nama Modul	new.php
Input	npm, nama, dan teks dalam format markdown
Output	-
Tabel yang diakses	InfoMahasiswa dan Histori
Deskripsi	Pengguna memasukkan npm, nama, dan keterangan mahasiswa pada teks area yang telah disediakan menggunakan teks dengan sintaks Markdown lalu pengguna menyimpan untuk membuat entri baru tersebut. Pengguna dapat kembali ke modul pilih mahasiswa tanpa melakukan perubahan.

4.3 Perancangan Diagram Relasional

Berdasarkan ERD pada sub sub bab 3.5.3, dapat dihasilkan perancangan diagram relasional yang dapat dilihat pada Gambar 4.8.



Gambar 4.8: Diagram Relasional

4.4 Perancangan Tabel Sistem Informasi Riwayat Mahasiswa

4.4.1 Perancangan Tabel Info Mahasiswa

Untuk rancangan tabel info mahasiswa dapat dilihat pada Tabel 4.7.

Tabel 4.7: Rancangan Tabel Info Mahasiswa

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

4.4.2 Perancangan Tabel Histori

Untuk rancangan tabel histori dapat dilihat pada Tabel 4.8.

Tabel 4.8: Rancangan Tabel Histori

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

BAB 5

IMPLEMENTASI DAN PENGUJIAN PERANGKAT LUNAK

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

5.1 Implementasi Perangkat Lunak

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

5.1.1 Lingkungan Implementasi Perangkat Keras

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

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

5.1.2 Lingkungan Implementasi Perangkat Lunak

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

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

- (f) Library: Google APIs Client Library untuk PHP
- (g) Javascript: Strapdown.js
- (h) Framework: Foundation 5

5.1.3 Hasil Implementasi Perangkat Lunak

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

5.2 Implementasi Basis Data

Implementasi basis data dalam sistem informasi riwayat mahasiswa, tahap pertama membuat sebuah basis data baru dan memberi nama sirm untuk basis data tersebut. Untuk kode dapat dilihat di bawah ini.

```
1 | CREATE DATABASE 'sirm' ;
```

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

- * Tabel InfoMahasiswa, digunakan untuk menyimpan semua data mahasiswa yang dapat diakses oleh pengguna. Untuk kode dapat dilihat di bawah ini.

```
1 | CREATE TABLE 'info_mahasiswa' (
2 | 'npm' varchar(10) NOT NULL,
3 | 'nama' varchar(60) NOT NULL,
4 | 'keterangan' text NOT NULL,
5 | 'pembaruan_terakhir' datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
6 | PRIMARY KEY ('npm'),
7 | ) ENGINE=InnoDB DEFAULT CHARSET=latin1 ;
```

- * Tabel Histori, digunakan untuk menyimpan semua data histori baik aksi pengguna dan riwayat mahasiswa. Untuk kode dapat dilihat di bawah ini.

```
1 | CREATE TABLE 'histori' (
2 | 'id_histori' int(5) NOT NULL AUTO_INCREMENT,
3 | 'npm' varchar(10) NOT NULL,
4 | 'pengguna' varchar(60) NOT NULL,
5 | 'status' text NOT NULL,
6 | 'tanggal_pembaruan' datetime NOT NULL,
7 | 'keterangan' text NOT NULL,
8 | PRIMARY KEY ('id_histori'),
9 | KEY 'npm' ('npm'),
10 | ) ENGINE=InnoDB DEFAULT CHARSET=latin1 ;
```

5.3 Pengujian Perangkat Lunak

Pada bagian ini akan dibahas mengenai pengujian yang akan dilakukan terhadap perangkat lunak. Pengujian tersebut terdiri dari dua bagian yaitu pengujian fungsional dan pengujian eksperimental. Pengujian fungsional bertujuan untuk memastikan bahwa seluruh fungsi yang dibangun pada perangkat lunak berjalan sesuai dengan yang direncanakan. Sedangkan pengujian eksperimental bertujuan untuk mengujikan perangkat lunak langsung ke pengguna. Pada bagian pengujian terdapat perubahan program pada

bagian oauth.php, jadi dapat menjalankan pengujian dengan email yang diakhiri @student.unpar.ac.id dikarenakan penulis tidak memiliki email yang diakhiri @unpar.ac.id. Kode program untuk pengujian perangkat lunak dapat dilihat pada Lampiran ??.

5.3.1 Lingkungan Pengujian Perangkat Keras

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

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

5.3.2 Lingkungan Pengujian Perangkat Lunak

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

- (a) Sistem Operasi: Windows 8.1 Pro 64-bit
- (b) Bahasa Pemrograman: PHP Version 5.6.3
- (c) Aplikasi: XAMPP v5.6.3
- (d) DBMS: MySQL
- (e) Aplikasi web browser: Google Chrome
- (f) Library: Google APIs Client Library untuk PHP
- (g) Javascript: Strapdown.js
- (h) Framework: Foundation 5

5.3.3 Pengujian Fungsional

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

(1) Fungsi login

Pengujian fungsi ini dilakukan untuk memastikan perangkat lunak terhubung ke server Google untuk melakukan otentikasi dan otorisasi serta memeriksa apakah email yang digunakan untuk login diakhiri "@unpar.ac.id" atau "@student.unpar.ac.id" dikarenakan penulis tidak mempunyai akun dosen.

Contoh kasus adalah melakukan login sebanyak dua kali, yang pertama menggunakan email yang diakhiri "@unpar.ac.id" atau "@student.unpar.ac.id", dan yang kedua menggunakan email yang diakhiri selain "@unpar.ac.id" dan "@student.unpar.ac.id".

Pengujian pertama pengguna membuka halaman index.php dapat dilihat pada Gambar 5.1. Lalu pengguna melakukan login menggunakan email "7310013@sudent.unpar.ac.id"

dapat dilihat pada Gambar 5.2. Lalu akan ada konfirmasi bahwa akun yang digunakan dikelola oleh student.unpar.ac.id dapat dilihat pada Gambar 5.3. Lalu pengguna akan diarahkan ke CAS (Central Authentication Service) UNPAR dan melakukan login kembali dapat dilihat pada Gambar 5.4. Lalu pengguna akan diminta untuk memberikan izin akses dari pihak pengguna dapat dilihat pada Gambar 5.5. Setelah pengguna memberikan izin akses maka akan dilakukan dengan fungsi memilih mahasiswa yang akan dibahas pada poin berikutnya. Sedangkan pengujian kedua pengguna melakukan login menggunakan email "bletack@gmail.com" dapat dilihat pada Gambar 5.6. Lalu pengguna akan mendapat *alert* karena email yang digunakan tidak sesuai dengan ketentuan dapat dilihat pada Gambar 5.7. Setelah pengguna menekan tombol ok pada *alert* maka pengguna akan dikembalikan ke halaman index.php. Hal ini menunjukkan fungsi login sudah berjalan dengan baik.

(2) Fungsi memilih mahasiswa

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

(3) Fungsi melihat info mahasiswa

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

(4) Fungsi mengedit info mahasiswa

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

(5) Fungsi melihat histori

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

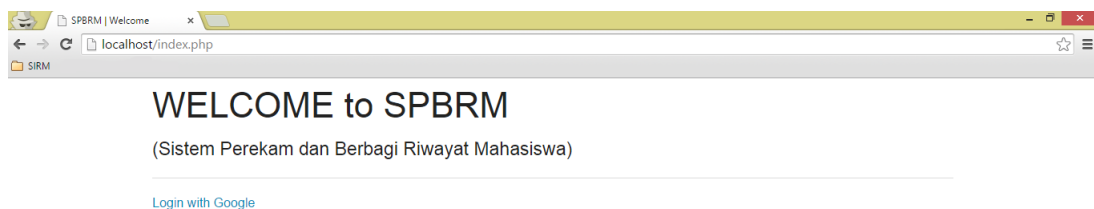
dan Gambar 5.13. Hal ini menunjukkan fungsi melihat histori sudah berjalan dengan baik.

(6) Fungsi membuat entri baru

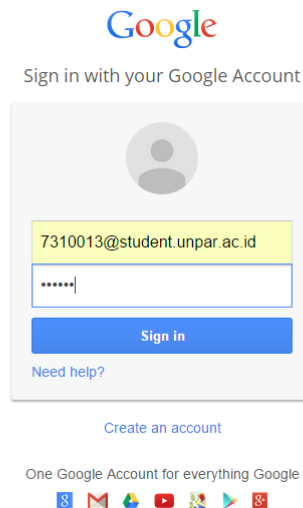
Pengujian fungsi ini dilakukan untuk memastikan pada saat membuat entri baru terdapat template markdown dan berhasil menyimpan entri baru tersebut. Contoh pengujian menambahkan entri baru untuk mahasiswa yang memiliki npm 2010730014, nama Nadia, dan keterangan sesuai template. Terdapat template markdown pada saat membuka halaman new.php dapat dilihat pada Gambar 5.14. Mengisi data npm 2010730014 dan nama Nadia dapat dilihat pada Gambar 5.15. Setelah pengguna menekan tombol "Simpan" maka data yang telah dimasukan akan tersimpan dan pengguna akan dikembalikan ke halaman list.php. Pengguna dapat melihat entri baru dengan npm 2010730014 dan nama Nadia telah masuk kedalam tabel dapat dilihat pada Gambar 5.16. Hal ini menunjukkan fungsi membuat entri baru sudah berjalan dengan baik.

(7) Antarmuka yang responsif

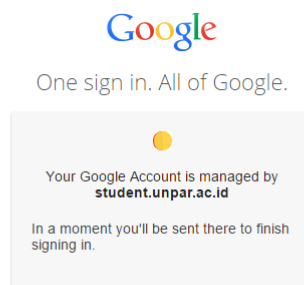
Pengujian antarmuka yang responsif dilakukan untuk memastikan tampilan antarmuka yang dibuat menggunakan Zurb Foundation berhasil. Contoh pengujian dilakukan dengan menggunakan mesin pencari yang telah dikecilkan ukurannya, lalu membuka index.php, list.php, dan new.php pada mesin pencari tersebut. Pengujian untuk index.php dapat dilihat pada Gambar 5.17, untuk list.php dapat dilihat pada Gambar 5.18, dan untuk new.php dapat dilihat pada Gambar 5.19. Hal ini menunjukkan antarmuka yang responsif sudah berjalan dengan baik.



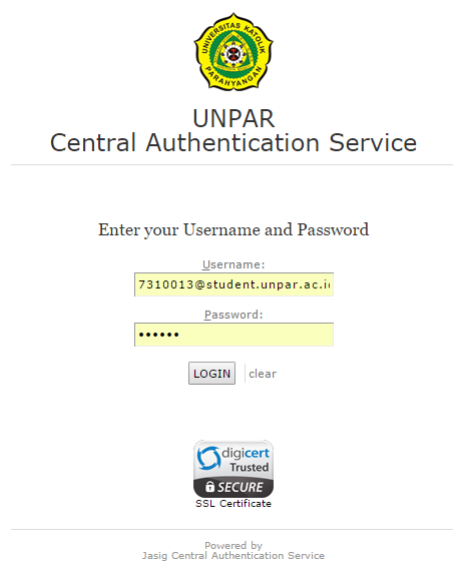
Gambar 5.1: Membuka Halaman index.php



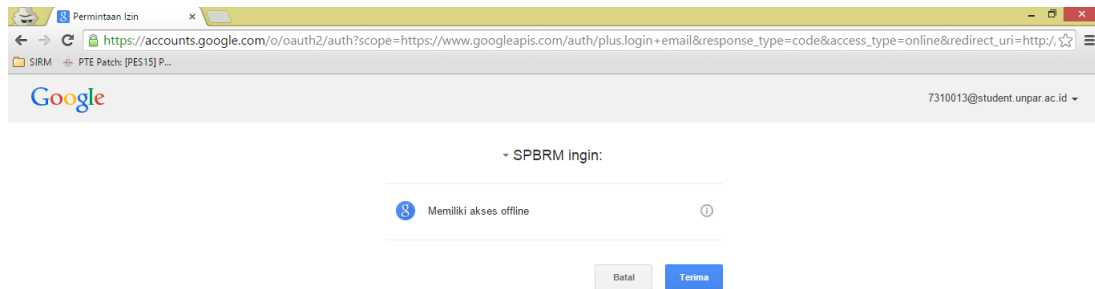
Gambar 5.2: Login Dengan Email yang Diakhiri "@student.unpar.ac.id"



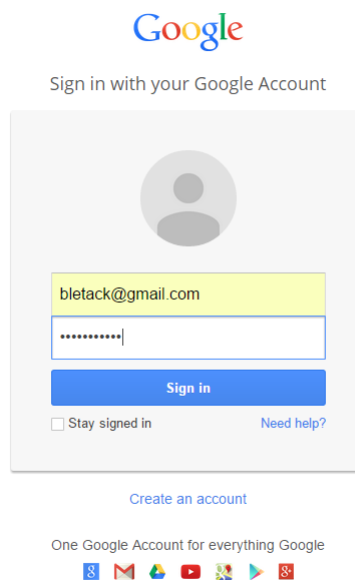
Gambar 5.3: Konfirmasi Email yang Dikelola oleh student.unpar.ac.id



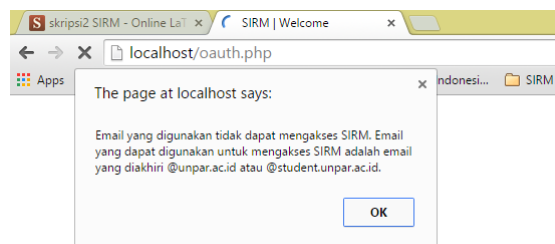
Gambar 5.4: CAS UNPAR



Gambar 5.5: Izin Akses Dari Pihak Pengguna



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



Gambar 5.7: Alert Email yang Digunakan Tidak Dapat Mengakses SIRM

Pilih NPM yang ingin dicari / tambah baru.

Add
Logout

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

Gambar 5.8: Memilih Mahasiswa

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

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

NPM : 2010730013
Nama : Samuel Herman

Umum

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

Catatan

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

Gambar 5.9: Melihat Info Mahasiswa

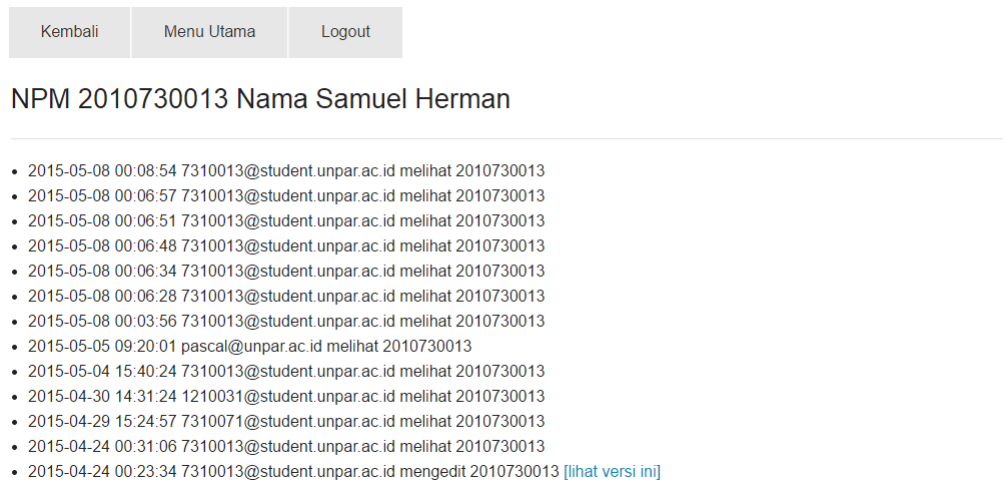
Anda mengedit catatan mahasiswa ini sebagai 7310013@student.unpar.ac.id.
NPM 2010730013 Nama Samuel Herman

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

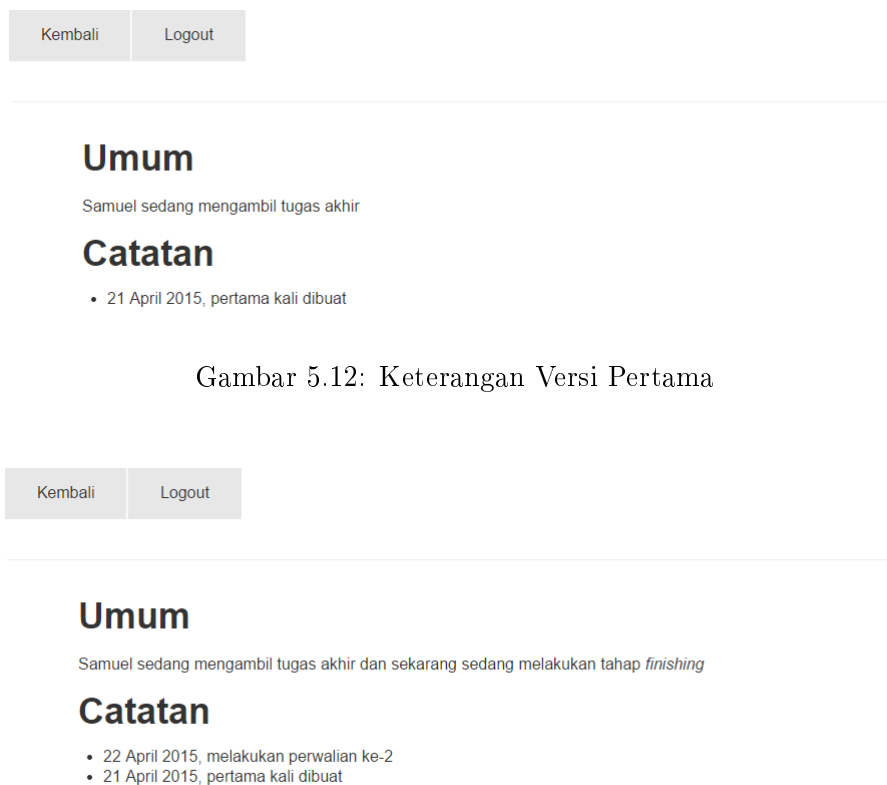
Umum
Samuel sedang mengambil tugas akhir dan sekarang sedang melakukan tahap *_finishing_*

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

Gambar 5.10: Mengedit Info Mahasiswa



Gambar 5.11: Melihat Histori



Gambar 5.12: Keterangan Versi Pertama

Gambar 5.13: Keterangan Versi Kedua

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

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

NPM

Nama

Umum

Isilah deskripsi umum mahasiswa disini.

Catatan

* 9 Oktober 2014, pertama kali dibuat

Gambar 5.14: Template Entri Baru

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

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

NPM

Nama

Umum

Nadia mahasiswa tingkat akhir yang sedang menunggu sidang

Catatan

* 8 April 2015, pertama kali dibuat

Gambar 5.15: Membuat Entri Baru

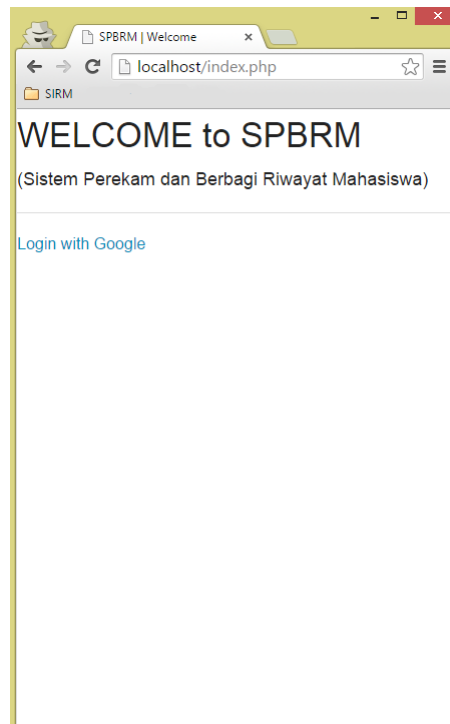
Pilih NPM yang ingin dicari / tambah baru.

Add

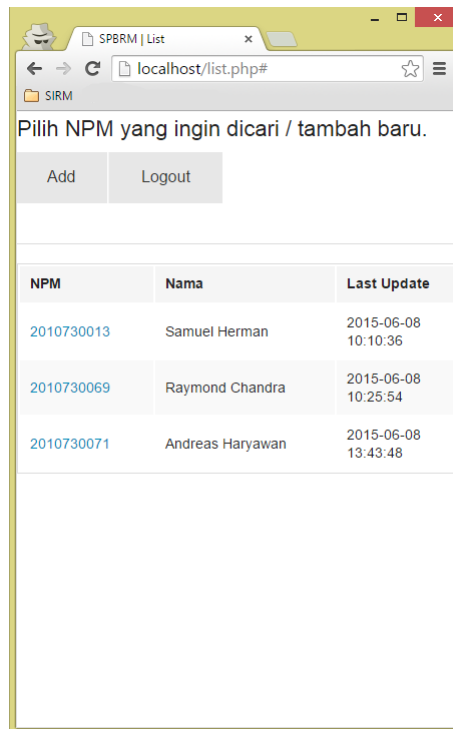
Logout

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

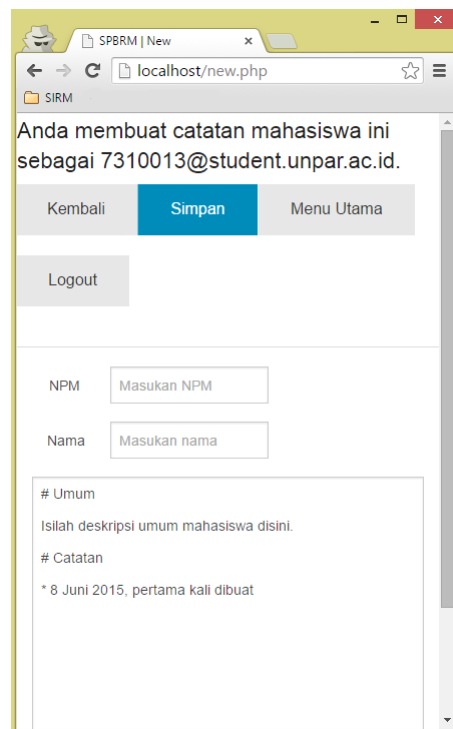
Gambar 5.16: Entri Baru Berhasil Dibuat



Gambar 5.17: Antarmuka Responsif index.php



Gambar 5.18: Antarmuka Responsif list.php



Gambar 5.19: Antarmuka Responsif new.php

5.3.4 Pengujian Eksperimental

Pengujian eksperimental dilakukan langsung ke lima orang dosen. Kelima dosen menggunakan

BAB 6

KESIMPULAN DAN SARAN

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

6.1 Kesimpulan

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

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

6.2 Saran

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

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

DAFTAR REFERENSI

- [1] C. E. Nugraheni, L. Owen, F. E. Yosafat, L. Abednego, P. T. Imanto, and T. A. Basuki, *Petunjuk Pelaksanaan Kegiatan Akademik Tahun 2014/2015*. Falkutas Teknologi Informasi dan Sains (FTIS), 2014.
- [2] Google, Inc, *Google Identity Platform*. <https://developers.google.com/accounts/docs/OAuth2>, 2013.
- [3] Google, Inc, *Google+ Platform*. <https://developers.google.com/+api/oauth>, 2013.
- [4] Google, Inc, *API Client Library for PHP*. <https://developers.google.com/api-client-library/php/start/getstarted>, 2015.
- [5] A. Kumar, *Markdown Guide*. Self-published, 2015.
- [6] GitHub, Inc, *GitHub Flavored Markdown*. <https://help.github.com/articles/github-flavored-markdown/>, 2015.
- [7] A. Adib, *Strapdown.js - Instant and elegant Markdown documents*. <http://strapdownjs.com/>, 2014.
- [8] ZURB, Inc, *Foundation Documentation*. <http://foundation.zurb.com/docs/>, 2015.

LAMPIRAN A

KODE PROGRAM

Listing A.1: index.php

```

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

```

Listing A.2: oauth.php

```

1 <!doctype html>
2 <html class="no-js" lang="en">

```

```

3 | <head>
4 |     <meta charset="utf-8" />
5 |     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6 |     <title>SIRM | OAuth</title>
7 |     <link rel="stylesheet" href="css/foundation.css" />
8 |     <script src="js/vendor/modernizr.js"></script>
9 | </head>
10 | <body>
11 |     <?php
12 |         include_once "google-api-php-client-master/src/Google/Client.php";
13 |         include_once "google-api-php-client-master/src/Google/Service/Oauth2.php";
14 |
15 |         session_start();
16 |
17 |         include_once "client.php";
18 |         $client->setScopes(array('https://www.googleapis.com/auth/plus.login','email'));
19 |         $plus = new Google_Service_Oauth2($client);
20 |
21 |         if (isset($_GET['code']))
22 |         {
23 |             $client->authenticate($_GET['code']);
24 |             $_SESSION['access_token'] = $client->getAccessToken();
25 |             header('Location: http://'. $_SERVER['HTTP_HOST'] . $_SERVER['PHP_SELF']);
26 |         }
27 |
28 |         if (isset($_SESSION['access_token']))
29 |         {
30 |             $client->setAccessToken($_SESSION['access_token']);
31 |         }
32 |
33 |         if ($client->getAccessToken())
34 |         {
35 |             $info = $plus->userinfo;
36 |             $userinfo = $info->get();
37 |             $email = ($userinfo['email']);
38 |             $_SESSION['email'] = $email;
39 |         }
40 |
41 |         $status="";
42 |         function is_valid_email($email)
43 |         {
44 |             $result = 'valid_email';
45 |             if (!preg_match("[a-zA-Z0-9_+]+@unpar.ac.id+$", $email))
46 |             {
47 |                 $result = 'invalid_email';
48 |             }
49 |             return $result;
50 |         }
51 |         $status = is_valid_email($email);
52 |
53 |         if ($status == "valid_email")
54 |         {
55 |             header("Location: list.php");
56 |             exit;
57 |         }
58 |         else
59 |         {
60 |             echo "<script>alert('Email yang digunakan tidak dapat mengakses SIRM. Email yang dapat digunakan untuk mengakses SIRM adalah email yang diakhiri @unpar.ac.id atau @student.unpar.ac.id.');

```

Listing A.3: list.php

```

1 | <!doctype html>
2 | <html class="no-js" lang="en">
3 |     <head>
4 |         <meta charset="utf-8" />
5 |         <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6 |         <title>SIRM | List</title>
7 |         <link rel="stylesheet" href="css/foundation.css" />
8 |         <script src="js/vendor/modernizr.js"></script>
9 |     </head>
10 |     <body>
11 |         <?php
12 |             session_start();
13 |
14 |             <?>
15 |             <div class="row">
16 |                 <h3>Pilih NPM yang ingin dicari / tambah baru.</h3>
17 |                 <ul class="button-group">
18 |                     <li><a href="new.php" class="button_secondary">Add</a></li>
19 |                     <li><a href="index.php?logout" class="button_secondary">Logout</a></li>
20 |                 </ul>
21 |                 <hr/>
22 |             </div>
23 |
24 |             <div class="row">
25 |                 <?php
26 |                     include_once "configDatabase.php";
27 |
28 |                     if (! $id_mysql)

```



```

29         die("Database_tidak_bisa_dibuka");
30     }
31
32     if(! mysql_select_db("sirm", $id_mysql))
33     {
34         die("Database_tidak_bisa_dipilih");
35     }
36
37     $hasil = mysql_query("SELECT_*_FROM_info_mahasiswa", $id_mysql);
38
39     if(! $hasil)
40     {
41         die("Permintaan_gagal");
42     }
43
44     echo "<table>
45     <thead>
46     <tr>
47     <th width='250'>NPM</th>
48     <th width='500'>Nama</th>
49     <th width='250'>Last_Update</th>
50     </tr>
51     </thead>";
52
53     while($row = mysql_fetch_array($hasil))
54     {
55         echo "<tr>";
56         echo "<td x:href='view.php?npm=" . $row['npm'] . ">" . $row['npm'] . "</a></td>";
57         echo "<td>" . $row['nama'] . "</td>";
58         echo "<td>" . $row['pembaruan_terakhir'] . "</td>";
59         echo "</tr>";
60     }
61     echo "</table>";
62
63     ?>
64 </div>
65 </body>
66 </html>

```

Listing A.4: view.php

```

1 <!doctype html>
2 <html class="no-js" lang="en">
3     <head>
4         <meta charset="utf-8" />
5         <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6         <title>SIRM | View</title>
7         <link rel="stylesheet" href="css/foundation.css" />
8         <script src="js/vendor/modernizr.js"></script>
9     </head>
10    <body>
11        <?php
12            session_start();
13            $npm = $_GET["npm"];
14
15            ?>
16            <div class="row">
17                <div class="small-11 small-centered columns">
18                    <h3>Anda melihat catatan mahasiswa ini sebagai <?php echo $_SESSION['email']
19                    </h3>
20                    <ul class="button-group">
21                        <li><a href="edit.php?npm=<?php echo $npm?>" class="button secondary">Edit
22                        </a></li>
23                        <li><a href="history.php?npm=<?php echo $npm?>" class="button secondary">
24                        Lihat Histori</a></li>
25                        <li><a href="list.php" class="button secondary">Menu Utama</a></li>
26                        <li><a href="index.php?logout" class="button secondary">Logout</a></li>
27                    </ul>
28                </div>
29            </div>
30
31            <?php
32                include_once "configDatabase.php";
33
34                if(! $id_mysql)
35                {
36                    die("Database_tidak_bisa_dibuka");
37                }
38
39                if(! mysql_select_db("sirm", $id_mysql))
40                {
41                    die("Database_tidak_bisa_dipilih");
42                }
43
44                $lihat = "INSERT INTO_histori_(npm, pengguna, status, tanggal_pembaruan, keterangan)
45                VALUES('". mysql_real_escape_string($npm) . "', '$_SESSION['email']', 'melihat', now(), '')";
46
47                if (mysql_query($lihat) === TRUE)
48                {
49                }
50                else
51                {
52                    echo "Error: " . $lihat . "<br>" . $id_mysql->error;
53                }
54
55                $cari = mysql_query("SELECT_*_FROM_info_mahasiswa WHERE npm='$_npm'", $id_mysql);
56
57                while($row = mysql_fetch_array($cari))

```

```

52         {
53             echo "NPM:_" . $row['npm']; echo "<br>";
54             echo "Nama:_" . $row['nama']; echo "<br>";
55         }
56     }
57 }
58 <?php
59 echo $row['keterangan'];
60 ?>
61 </xmp>
62 <?php
63     }
64 }
65 ?>
66 <script src="js/0.2/strapdown.js"></script>
67 </body>
68 </html>

```

Listing A.5: edit.php

```

1 <!doctype html>
2 <html class="no-js" lang="en">
3     <head>
4         <meta charset="utf-8" />
5         <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6         <title>SIRM | Edit</title>
7         <link rel="stylesheet" href="css/foundation.css" />
8         <script src="js/vendor/modernizr.js"></script>
9     </head>
10    <body>
11        <?php
12            session_start();
13            $npm = $_GET['npm'];
14
15            include_once "configDatabase.php";
16
17            if(! $id_mysql)
18            {
19                die("Database tidak bisa dibuka");
20            }
21
22            if(! mysql_select_db("sirm", $id_mysql))
23            {
24                die("Database tidak bisa dipilih");
25            }
26
27            $hasil = mysql_query("SELECT * FROM info_mahasiswa WHERE npm='$npm'", $id_mysql);
28
29            if(! $hasil)
30            {
31                die("Permintaan gagal");
32            }
33
34            while($row = mysql_fetch_array($hasil))
35            {
36                $carinama = $row['nama'];
37                $cariketerangan = $row['keterangan'];
38            }
39
40            if(isset($_POST['submit']))
41            {
42                $keteranganbaru = "";
43                $keteranganbaru = $_POST['keteranganbaru'];
44
45                $sql1 = "UPDATE info_mahasiswa SET keterangan='$keteranganbaru', pembaruan_terakhir
46                    =now() WHERE npm='$npm'";
47                $sql2 = "INSERT INTO histori(npm, pengguna, status, tanggal_pembaruan, keterangan)
48                    VALUES('". mysql_real_escape_string($npm) . "', '$_SESSION[ 'email' ].', '
49                    mendedit', now(), '". mysql_real_escape_string($keteranganbaru) . "')";
50
51                if (mysql_query($sql1) & mysql_query($sql2) == TRUE)
52                {
53                    echo '<META-HTTP-EQUIV="Refresh" _CONTENT="1; _URL=list.php">';
54                }
55                else
56                {
57                    echo "Error: " . $sql1 . "<br>" . $id_mysql->error;
58                    echo "Error: " . $sql2 . "<br>" . $id_mysql->error;
59                }
60            }
61            else
62            {
63                ?>
64                <div class="row">
65                    <h3>Anda mengedit catatan mahasiswa ini sebagai <?php echo $_SESSION['email'];>.<br>
66                    NPM <?php echo $npm; ?> Nama <?php echo $carinama; ?>
67                </h3>
68            </div>
69            <form method="post" action="edit.php?npm=<?php echo $npm?>">
70                <div class="row">
71                    <ul class="button-group">
72                        <li><a href="view.php?npm=<?php echo $npm?>" class="button_secondary">
73                            Kembali</a></li>
74                        <li><input class="button" type="submit" name="submit" value="Simpan"></li>
75                        <li><a href="list.php" class="button_secondary">Menu Utama</a></li>
76                        <li><a href="index.php?logout" class="button_secondary">Logout</a></li>

```

```

73         </ul>
74     </div>
75     <div class="row">
76         <div class="small-12 columns">
77             <textarea style="height:_300px;" placeholder="<?php echo $cariketerangan;_
78                 ?>" name="keteranganbaru"><?php echo $cariketerangan; ?></textarea>
79         </div>
80     </div>
81 </form>
82 <?php
83 }
84 ?>
85 </body>
86 </html>

```

Listing A.6: history.php

```

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

```

Listing A.7: past.php

```

1 <!doctype html>

```

```

2 | <html class="no-js" lang="en">
3 |   <head>
4 |     <meta charset="utf-8" />
5 |     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6 |     <title>SIRM | Past</title>
7 |     <link rel="stylesheet" href="css/foundation.css" />
8 |     <script src="js/vendor/modernizr.js"></script>
9 |   </head>
10 |   <body>
11 |     <?php
12 |       session_start();
13 |       $id = $_GET["id"];
14 |       ?>
15 |       <div class="row">
16 |         <div class="small-11 small-centered columns">
17 |           <ul class="button-group">
18 |             <li <a href="javascript:history.back(1)" class="button_secondary">Kembali</a></li>
19 |             <li <a href="index.php?logout" class="button_secondary">Logout</a></li>
20 |           </ul>
21 |           <hr/>
22 |           <?php
23 |
24 |             include_once "configDatabase.php";
25 |
26 |             if(! $id_mysql)
27 |             {
28 |               die("Database_tidak_bisa_dibuka");
29 |             }
30 |
31 |             if(! mysql_select_db("sirm", $id_mysql))
32 |             {
33 |               die("Database_tidak_bisa_dipilih");
34 |             }
35 |
36 |             $cari = mysql_query("SELECT_keterangan_FROM_histori_WHERE_id_histori='$id '",
37 |                                   $id_mysql);
38 |             while($row = mysql_fetch_array($cari))
39 |             {
40 |               ?>
41 |             </div>
42 |           <xmp style="display:none;">
43 |           <?php
44 |             echo $row['keterangan'];
45 |           ?>
46 |         </xmp>
47 |         <?php
48 |         }
49 |       ?>
50 |       <script src="js/0.2/strapdown.js"></script>
51 |     </body>
52 | </html>

```

Listing A.8: new.php

```

1 | <!doctype html>
2 | <html class="no-js" lang="en">
3 |   <head>
4 |     <meta charset="utf-8" />
5 |     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6 |     <title>SIRM | New</title>
7 |     <link rel="stylesheet" href="css/foundation.css" />
8 |     <script src="js/vendor/modernizr.js"></script>
9 |   </head>
10 |   <body>
11 |     <?php
12 |       session_start();
13 |
14 |       if(isset($_POST['submit']))
15 |       {
16 |         include_once "configDatabase.php";
17 |
18 |         if(! $id_mysql)
19 |         {
20 |           die("Database_tidak_bisa_dibuka");
21 |         }
22 |
23 |         if(! mysql_select_db("sirm", $id_mysql))
24 |         {
25 |           die("Database_tidak_bisa_dipilih");
26 |         }
27 |
28 |         $npm = $nama = $keterangan = "";
29 |
30 |         $npm = $_POST['npm'];
31 |         $nama = $_POST['nama'];
32 |         $keterangan = $_POST['keterangan'];
33 |
34 |         $cek = "SELECT_npm_from_info_mahasiswa_where_npm='". mysql_real_escape_string($npm)
35 |               . "'";
36 |         $found = mysql_query($cek) or die(mysql_error());
37 |         if(mysql_num_rows($found)>0)
38 |         {
39 |           echo "<script>alert('Data_telah_terdaftar..Silahkan_diulangi_dengan_data_yang_
40 |             lain..');window.location.href='new.php';</script>";
41 |         }

```

```

40         else
41         {
42             $sql1 = "INSERT INTO info_mahasiswa (npm, nama, keterangan) VALUES ('" .
                mysql_real_escape_string($npm) . "','" . mysql_real_escape_string($nama)
                . "','" . mysql_real_escape_string($keterangan) . "')";
43             $sql2 = "INSERT INTO histori (npm, pengguna, status, tanggal_pembaruan,
                keterangan) VALUES ('" . mysql_real_escape_string($npm) . "','" . $SESSION[
                'email'] . "','" . 'membuat entri ' . now() . "','" . mysql_real_escape_string(
                $keterangan) . "')";
44
45             if (mysql_query($sql1) & mysql_query($sql2) === TRUE)
46             {
47                 echo '<META-HTTP-EQUIV="Refresh" _CONTENT="1;_URL=list.php">';
48             }
49             else
50             {
51                 echo "Error: " . $sql1 . "<br>" . $id_mysql->error;
52                 echo "Error: " . $sql2 . "<br>" . $id_mysql->error;
53             }
54         }
55     }
56     else
57     {
58         ?>
59         <div class="row">
60             <h3>Anda membuat catatan mahasiswa ini sebagai <?php echo $SESSION['email']
                ]?>.</h3>
61         </div>
62         <form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>"
63         <div class="row">
64             <ul class="button-group">
65                 <li><a href="list.php" class="button_secondary">Kembali</a></li>
66                 <li><input class="button" type="submit" name="submit" value="Simpan"></li>
67                 <li><a href="list.php" class="button_secondary">Menu Utama</a></li>
68                 <li><a href="index.php?logout" class="button_secondary">Logout</a></li>
69             </ul>
70             <hr/>
71         </div>
72         <div class="small-8_columns">
73             <div class="row">
74                 <div class="small-3_columns">
75                     <label for="right-label" class="right_inline">NPM</label>
76                 </div>
77                 <div class="small-9_columns">
78                     <input type="text" name="npm" id="right-label" placeholder="Masukan
                        _NPM">
79                 </div>
80             </div>
81             <div class="row">
82                 <div class="small-3_columns">
83                     <label for="right-label" class="right_inline">Nama</label>
84                 </div>
85                 <div class="small-9_columns">
86                     <input type="text" name="nama" id="right-label" placeholder="
                        Masukan_nama">
87                 </div>
88             </div>
89         </div>
90         <div class="row">
91             <div class="small-12_columns">
92                 <textarea style="height: 300px;" name="keterangan">
93             # Umum
94             Isilah deskripsi umum mahasiswa disini.
95             # Catatan
96             * 9 Oktober 2014, pertama kali dibuat
97             </div>
98         </div>
99         </div>
100     </form>
101 </div>
102 </div>
103 </div>
104 <?php
105 }
106 ?>
107 </body>
108 </html>

```

Listing A.9: client.php

```

1 <?php
2 $client = new Google_Client();
3 $client->setClientId('568951368854-ufmbistn0pcaq0khubaf01a133orfgve.apps.googleusercontent.
    com');
4 $client->setClientSecret('-cSZ-AUmeQ9PaWWry_IpiBbi');
5 $client->setRedirectUri('http://localhost/oauth.php');
6 $client->setDeveloperKey('AlZaSyDRoDJAzUR_TsNUNRUeTYsBb7dFBQKZy7M');
7 ?>

```

Listing A.10: configDatabase.php

```

1 <?php
2 $pemain="admin";
3 $pass="admin";
4 $id_mysql=mysql_connect("localhost", $pemain, $pass);
5 ?>

```


LAMPIRAN B

KODE PROGRAM UNTUK PENGUJIAN

Listing B.1: index.php

```

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

```

Listing B.2: oauth.php

```

1 <!doctype html>
2 <html class="no-js" lang="en">
3   <head>
4     <meta charset="utf-8" />

```

```

5 | <meta name="viewport" content="width=device-width,initial-scale=1.0" />
6 | <title>SIRM | OAuth</title>
7 | <link rel="stylesheet" href="css/foundation.css" />
8 | <script src="js/vendor/modernizr.js"></script>
9 | </head>
10 | <body>
11 | <?php
12 |     include_once "google-api-php-client-master/src/Google/Client.php";
13 |     include_once "google-api-php-client-master/src/Google/Service/Oauth2.php";
14 |
15 |     session_start();
16 |
17 |     include_once "client.php";
18 |     $client->setScopes(array('https://www.googleapis.com/auth/plus.login','email'));
19 |     $plus = new Google_Service_Oauth2($client);
20 |
21 |     if (isset($_GET['code']))
22 |     {
23 |         $client->authenticate($_GET['code']);
24 |         $_SESSION['access_token'] = $client->getAccessToken();
25 |         header('Location: http://'. $_SERVER['HTTP_HOST'] . $_SERVER['PHP_SELF']);
26 |     }
27 |
28 |     if (isset($_SESSION['access_token']))
29 |     {
30 |         $client->setAccessToken($_SESSION['access_token']);
31 |     }
32 |
33 |     if ($client->getAccessToken())
34 |     {
35 |         $info = $plus->userinfo;
36 |         $userinfo = $info->get();
37 |         $email = ($userinfo['email']);
38 |         $_SESSION['email'] = $email;
39 |     }
40 |
41 |     $status="";
42 |     function is_valid_email($email)
43 |     {
44 |         $result = 'valid_email';
45 |         if(!preg_match("[a-zA-Z0-9_+]+@unpar.ac.id+$", $email) && !preg_match("[a-zA-Z0-9_+]+@student.unpar.ac.id+$", $email))
46 |         {
47 |             $result = 'invalid_email';
48 |         }
49 |         return $result;
50 |     }
51 |     $status = is_valid_email($email);
52 |
53 |     if($status == "valid_email")
54 |     {
55 |         header("Location: list.php");
56 |         exit;
57 |     }
58 |     else
59 |     {
60 |         echo "<script>alert('Email_yang_digunakan_tidak_dapat_mengakses_SIRM_Email_yang_dapat_digunakan_untuk_mengakses_SIRM_adalah_email_yang_diakhiri_@unpar.ac.id_atau_@student.unpar.ac.id. ');window.location.href='index.php?logout';</script>";
61 |         exit;
62 |     }
63 |     ?>
64 |     <?= is_valid_email($email); ?>
65 | </body>
66 | </html>

```

Listing B.3: list.php

```

1 | <!doctype html>
2 | <html class="no-js" lang="en">
3 | <head>
4 |     <meta charset="utf-8" />
5 |     <meta name="viewport" content="width=device-width,initial-scale=1.0" />
6 |     <title>SIRM | List</title>
7 |     <link rel="stylesheet" href="css/foundation.css" />
8 |     <script src="js/vendor/modernizr.js"></script>
9 | </head>
10 | <body>
11 | <?php
12 |     session_start();
13 |     ?>
14 |     <div class="row">
15 |         <h3>Pilih NPM yang ingin dicari / tambah baru.</h3>
16 |         <ul class="button-group">
17 |             <li><a href="new.php" class="button_secondary">Add</a></li>
18 |             <li><a href="index.php?logout" class="button_secondary">Logout</a></li>
19 |         </ul>
20 |         <hr/>
21 |     </div>
22 |
23 |     <div class="row">
24 |         <?php
25 |             include_once "configDatabase.php";
26 |
27 |             if(! $id_mysql)
28 |             {
29 |                 die("Database_tidak_bisa_dibuka");

```



```

30     }
31
32     if (! mysql_select_db("sirm", $id_mysql))
33     {
34         die("Database_tidak_bisa_dipilih");
35     }
36
37     $hasil = mysql_query("SELECT_*_FROM_info_mahasiswa", $id_mysql);
38
39     if (! $hasil)
40     {
41         die("Permintaan_gagal");
42     }
43
44     echo "<table>
45     <thead>
46     <tr>
47     <th width='250'>NPM</th>
48     <th width='500'>Nama</th>
49     <th width='250'>Last_Update</th>
50     </tr>
51     </thead>";
52
53     while($row = mysql_fetch_array($hasil))
54     {
55         echo "<tr>";
56         echo "<td><a href='view.php?npm=" . $row['npm'] . "'> . $row['npm'] . "</a></td>";
57         echo "<td> . $row['nama'] . "</td>";
58         echo "<td> . $row['pembaruan_terakhir'] . "</td>";
59         echo "</tr>";
60     }
61     echo "</table>";
62
63     ?>
64     </div>
65     </body>
66     </html>

```

Listing B.4: view.php

```

1  <!doctype html>
2  <html class="no-js" lang="en">
3  <head>
4      <meta charset="utf-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6      <title>SIRM | View</title>
7      <link rel="stylesheet" href="css/foundation.css" />
8      <script src="js/vendor/modernizr.js"></script>
9  </head>
10 <body>
11     <?php
12         session_start();
13         $npm = $_GET["npm"];
14     ?>
15     <div class="row">
16         <div class="small-11 small-centered columns">
17             <h3>Anda melihat catatan mahasiswa ini sebagai <?php echo $_SESSION['email']
18             ?>.</h3>
19             <ul class="button-group">
20                 <li><a href="edit.php?npm=<?php echo $npm?>" class="button secondary">Edit
21                 </a></li>
22                 <li><a href="history.php?npm=<?php echo $npm?>" class="button secondary">
23                     Lihat Histori</a></li>
24                 <li><a href="list.php" class="button secondary">Menu Utama</a></li>
25                 <li><a href="index.php?logout" class="button secondary">Logout</a></li>
26             </ul>
27             <hr/>
28         <?php
29             include_once "configDatabase.php";
30
31             if (! $id_mysql)
32             {
33                 die("Database_tidak_bisa_dibuka");
34             }
35
36             if (! mysql_select_db("sirm", $id_mysql))
37             {
38                 die("Database_tidak_bisa_dipilih");
39             }
40
41             $lihat = "INSERT INTO histori (npm, pengguna, status, tanggal_pembaruan, keterangan)
42             VALUES ('" . mysql_real_escape_string($npm) . "', '" . $_SESSION['email'] . "', '" .
43             melihat', now())";
44
45             if (mysql_query($lihat) == TRUE)
46             {
47                 echo "Error: " . $lihat . "<br>" . $id_mysql->error;
48             }
49
50             $cari = mysql_query("SELECT_*_FROM_info_mahasiswa WHERE npm=' $npm '", $id_mysql);
51
52             while($row = mysql_fetch_array($cari))
53             {

```

```

53         echo "NPM_:" ; echo $row['npm']; echo "<br>";
54         echo "Nama_:" ; echo $row['nama']; echo "<br>";
55     ?>
56     </div>
57 </div>
58 <xmp style="display:none;">
59 <?php
60 echo $row['keterangan'];
61 ?>
62 </xmp>
63 <?php
64     }
65 ?>
66 <script src="js/0.2/strapdown.js"></script>
67 </body>
68 </html>

```

Listing B.5: edit.php

```

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

```

```

74         <hr/>
75     </div>
76     <div class="row">
77         <div class="small-12_columns">
78             <textarea style="height:_300px;" placeholder="<?php_echo_$cariketerangan;_
79                 ?>" name="keteranganbaru"><?php_echo_$cariketerangan; ?></textarea>
80         </div>
81     </form>
82 <?php
83 }
84 ?>
85 </body>
86 </html>

```

Listing B.6: history.php

```

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

```

Listing B.7: past.php

```

1 <!doctype html>
2 <html class="no-js" lang="en">

```

```

3 | <head>
4 | <meta charset="utf-8" />
5 | <meta name="viewport" content="width=device-width,initial-scale=1.0" />
6 | <title>SIRM | Past</title>
7 | <link rel="stylesheet" href="css/foundation.css" />
8 | <script src="js/vendor/modernizr.js"></script>
9 | </head>
10 | <body>
11 | <?php
12 |     session_start();
13 |     $id = $_GET["id"];
14 |     ?>
15 |     <div class="row">
16 |         <div class="small-11_small-centered_columns">
17 |             <ul class="button-group">
18 |                 <li><a href="javascript:history.back(1)" class="button_secondary">Kembali</a></li>
19 |                 <li><a href="index.php?logout" class="button_secondary">Logout</a></li>
20 |             </ul>
21 |             <hr/>
22 |             <?php
23 |
24 |                 include_once "configDatabase.php";
25 |
26 |                 if(! $id_mysql)
27 |                 {
28 |                     die("Database_tidak_bisa_dibuka");
29 |                 }
30 |
31 |                 if(! mysql_select_db("sirm", $id_mysql))
32 |                 {
33 |                     die("Database_tidak_bisa_dipilih");
34 |                 }
35 |
36 |                 $cari = mysql_query("SELECT_keterangan_FROM_histori_WHERE_id_histori='$id '",
37 |                                     $id_mysql);
38 |                 while($row = mysql_fetch_array($cari))
39 |                 {
40 |                     ?>
41 |                 </div>
42 |             </div>
43 | <xmp style="display:none;">
44 | <?php
45 | echo $row['keterangan'];
46 | ?>
47 | </xmp>
48 | <?php
49 | }
50 | ?>
51 | <script src="js/0.2/strapdown.js"></script>
52 | </body>
53 | </html>

```

Listing B.8: new.php

```

1 | <!doctype html>
2 | <html class="no-js" lang="en">
3 | <head>
4 | <meta charset="utf-8" />
5 | <meta name="viewport" content="width=device-width,initial-scale=1.0" />
6 | <title>SIRM | New</title>
7 | <link rel="stylesheet" href="css/foundation.css" />
8 | <script src="js/vendor/modernizr.js"></script>
9 | </head>
10 | <body>
11 | <?php
12 |     session_start();
13 |
14 |     if(isset($_POST['submit']))
15 |     {
16 |         include_once "configDatabase.php";
17 |
18 |         if(! $id_mysql)
19 |         {
20 |             die("Database_tidak_bisa_dibuka");
21 |         }
22 |
23 |         if(! mysql_select_db("sirm", $id_mysql))
24 |         {
25 |             die("Database_tidak_bisa_dipilih");
26 |         }
27 |
28 |         $npm = $nama = $keterangan = "";
29 |
30 |         $npm = $_POST['npm'];
31 |         $nama = $_POST['nama'];
32 |         $keterangan = $_POST['keterangan'];
33 |
34 |         $cek = "SELECT_npm_from_info_mahasiswa_where_npm='". mysql_real_escape_string($npm)
35 |               "'";
36 |         $found = mysql_query($cek) or die(mysql_error());
37 |         if(mysql_num_rows($found)>0)
38 |         {
39 |             echo "<script>alert('Data_telah_terdaftar..Silahkan_diulangi_dengan_data_yang_
40 |                 lain.');

```

```

41     {
42         $sql1 = "INSERT INTO info_mahasiswa (npm, nama, keterangan) VALUES ('" .
            mysql_real_escape_string($npm) . "','" . mysql_real_escape_string($nama)
            . "','" . mysql_real_escape_string($keterangan) . "')";
43         $sql2 = "INSERT INTO histori (npm, pengguna, status, tanggal_pembaruan,
            keterangan) VALUES ('" . mysql_real_escape_string($npm) . "','" . $SESSION[
            'email'] . "','" . 'membuat entri ' . now() . "','" . mysql_real_escape_string(
            $keterangan) . "')";
44
45         if (mysql_query($sql1) & mysql_query($sql2) === TRUE)
46         {
47             echo '<META_HTTP-EQUIV="Refresh" _CONTENT="1;_URL=list.php">';
48         }
49         else
50         {
51             echo "Error: " . $sql1 . "<br>" . $id_mysql->error;
52             echo "Error: " . $sql2 . "<br>" . $id_mysql->error;
53         }
54     }
55 }
56 else
57 {
58     ?>
59     <div class="row">
60         <h3>Anda membuat catatan mahasiswa ini sebagai <?php echo $_SESSION['email']
            ]?>. </h3>
61     </div>
62     <form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
63         <div class="row">
64             <ul class="button-group">
65                 <li><a href="list.php" class="button_secondary">Kembali</a></li>
66                 <li><input class="button" type="submit" name="submit" value="Simpan"></li>
67                 <li><a href="list.php" class="button_secondary">Menu Utama</a></li>
68                 <li><a href="index.php?logout" class="button_secondary">Logout</a></li>
69             </ul>
70             <hr>
71         </div>
72         <div class="small-8_columns">
73             <div class="row">
74                 <div class="small-3_columns">
75                     <label for="right-label" class="right_inline">NPM</label>
76                 </div>
77                 <div class="small-9_columns">
78                     <input type="text" name="npm" id="right-label" placeholder="Masukan
                        _NPM">
79                 </div>
80             </div>
81             <div class="row">
82                 <div class="small-3_columns">
83                     <label for="right-label" class="right_inline">Nama</label>
84                 </div>
85                 <div class="small-9_columns">
86                     <input type="text" name="nama" id="right-label" placeholder="
                        Masukan_nama">
87                 </div>
88             </div>
89             <div class="row">
90                 <div class="small-12_columns">
91                     <div class="small-12_columns">
92                         <textarea style="height:_300px;" name="keterangan">
93                     # Umum
94
95                     Isilah deskripsi umum mahasiswa disini.
96
97                     # Catatan
98
99                     * 9 Oktober 2014, pertama kali dibuat
100                         </textarea>
101                     </div>
102                 </div>
103             </form>
104         <?php
105         }
106     ?>
107 </body>
108 </html>

```

Listing B.9: client.php

```

1 <?php
2 $client = new Google_Client();
3 $client->setClientId('568951368854-ufmbistn0pcaq0khubaf01a133orfgeve.apps.googleusercontent.
    com');
4 $client->setClientSecret('-cSZ-AUmeQ9PaWWry_IpiBBi');
5 $client->setRedirectUri('http://localhost/oauth.php');
6 $client->setDeveloperKey('AlzaSyDRoDJAzUR_TsNUNRUeTYsBb7dFBQKZy7M');
7 ?>

```

Listing B.10: configDatabase.php

```

1 <?php
2 $pemakai="admin";
3 $pass="admin";
4 $id_mysql=mysql_connect("localhost", $pemakai, $pass);
5 ?>

```


LAMPIRAN C

SURVEI KEBUTUHAN PENGGUNA

Untuk setiap pertanyaan di bawah, mohon mengisi jawaban dengan SS (Sangat Setuju), S (Setuju), N (Netral), TS (Tidak Setuju), STS (Sangat Tidak Setuju).

No	Pertanyaan	Jawaban
1	Dalam memutuskan sesuatu terhadap seorang mahasiswa, saya mempertimbangkan sisi humanis di luar prestasi akademis mahasiswa tersebut.	
2	Dalam memutuskan sesuatu terhadap seorang mahasiswa, saya berkonsultasi dengan rekan dosen.	
3	Dalam memutuskan sesuatu terhadap seorang mahasiswa, saya berkonsultasi dengan mahasiswa lain.	
4	Dalam memutuskan sesuatu terhadap seorang mahasiswa, saya disiplin menerapkan aturan yang berlaku.	
5	Dalam memutuskan sesuatu terhadap seorang mahasiswa, saya percaya penilaian sayalah yang paling baik.	
6	Dalam menggunakan aplikasi yang berhubungan dengan mahasiswa, saya cenderung menggunakan akun dosen yang diberikan oleh UNPAR.	
7	Dalam membuat catatan terhadap seorang mahasiswa, saya cenderung membuat format yang mempermudah untuk menulis dan membaca catatan tersebut.	
8	Dalam menggunakan aplikasi yang berhubungan dengan mahasiswa, saya cenderung menggunakan perangkat yang saya miliki (Laptop, tablet, atau handphone pribadi).	

LAMPIRAN D

KUESIONER PENGUJIAN EKSPERIMENTAL

Setelah melakukan pengujian SPBRM (Sistem Perekam dan Berbagi Riwayat Mahasiswa), mohon jawab kuesioner berikut.

No	Pertanyaan	Jawaban				
		SS	S	N	TS	STS
1	SPBRM membantu saya dalam mengingat setiap riwayat mahasiswa.					
2	Saya mengoperasikan SPBRM dengan mudah.					
3	Apakah SPBRM membantu pemahaman saya dalam mengelola riwayat mahasiswa?					
4	Saya merasa lebih efektif bekerja dengan SPBRM dalam mengelola riwayat mahasiswa.					
5	Saya merasa lebih efisien bekerja dengan SPBRM dalam mengelola riwayat mahasiswa.					