

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

Fakultas Teknologi Informasi dan Sains memiliki 14 dosen Teknik Informatika yang memiliki tugas untuk berinteraksi dengan banyak mahasiswa. Hal tersebut menimbulkan masalah bagi dosen untuk mengingat setiap riwayat mahasiswa yang berinteraksi dengan dosen. Untuk masalah tersebut, dibuatkan sebuah sistem yang mencatat setiap riwayat mahasiswa dan membagikan setiap riwayat mahasiswa ke semua dosen. Pada skripsi ini, dibahas pembangunan Sistem Perekam dan Berbagi Riwayat Mahasiswa (SPBRM) yang menggunakan 4 teknologi; Google OAuth, Markdown, StrapdownJS, Zurb Foundation.

SPBRM menggunakan Google OAuth untuk mengautentikasi dan mengotorisasi pengguna yang menggunakan akun dosen UNPAR. SPBRM menggunakan sintaksis Markdown untuk membuat format penulisan teks menjadi seragam. SPBRM menggunakan StrapdownJS untuk mengkonversi teks dengan format penulisan yang telah dibuat menggunakan Markdown menjadi tampilan HTML. SPBRM menggunakan Zurb Foundation untuk membuat tampilan antarmuka yang responsif. Berdasarkan pengujian SPBRM sudah berjalan dengan baik dan memberikan keluaran sesuai yang diharapkan pengguna.

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

ABSTRACT

The Faculty of Information Technology and Science has 14 lecturers of Information Technology whose duty is to interact with many students. This raises a problem for the lecturers to remember every student's history who has interaction with them. To solve the problem, there is a system made to record every student's history and share it to all the lecturers. This thesis will discuss about the establishment of the Student History Recording and Sharing System (SPBRM) which uses four technologies: Google OAuth, Markdown, StrapdownJS, Zurb Foundation.

Google OAuth used to authenticate and authorize users who use the account of UNPAR lecturers. Markdown syntax is to make text entry into a similar format. StrapdownJS used to convert the writing format text that has been made using Markdown into the HTML view. While the function of Zurb Foundation technology is to create a responsive user interface. Based SPBRM testing has been going well and it has given the user output as expected.

Keywords: Google OAuth, Markdown, StrapdownJS, Zurb Foundation

*Dipersembahkan untuk diri sendiri, keluarga, teman-teman, dan
semua orang yang berperan dalam pembuatan skripsi ini.*

KATA PENGANTAR

Terima kasih kepada Tuhan Yesus Kristus yang telah memberikan kesempatan kepada penulis untuk dapat menyelesaikan skripsi dengan judul "Sistem Perekan dan Berbagi Riwayat Mahasiswa untuk Dosen". Banyak kesulitan yang dihadapi dan berbagai masalah yang menjadikan beban pikiran selama proses pengerjaan skripsi ini. Skripsi ini dibuat sebagai salah satu syarat kelulusan di Universitas Katolik Parahyangan. Penulis juga ingin mengucapkan terima kasih kepada beberapa pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses pengerjaan skripsi ini, karena tanpa pihak-pihak tersebut skripsi ini mungkin tidak akan pernah selesai. Beberapa pihak tersebut adalah:

1. Ayah (Herman Limandjaja), Ibu (Emmi Muljati), Adik (Andreas Herman), dan seluruh keluarga yang telah memberikan doa dan dukungan.
2. Bapak Pascal Alfadian, M.Com. sebagai dosen pembimbing yang selalu sabar dan setia dalam memberikan arahan, kritik, dan dukungan selama proses penyusunan skripsi.
3. Bapak Lionov, M.Sc. dan Ibu Dr. Veronica Sri Moertini, Ir., MT. sebagai dosen penguji yang telah memberikan koreksi, kritik, dan saran agar skripsi ini menjadi lebih baik.
4. Seluruh dosen di Jurusan Teknik Informatika Fakultas Teknologi Informasi dan Sains, Universitas Katolik Parahyangan.
5. Seluruh anak Archer Youth Community yang selalu memberikan doa dan dukungan.
6. Billy Sebastian, Yohanes Subrata, Kenneth Natanael, Liviana Devi, dan Nerisa Arviana sebagai sahabat SBYjK.
7. Andreas Haryawan, Andri Agustian, Dominikus, Edwin Herawanputra, Grady Ireneus, Hans Wirya, Henry Setiadi, Kevin PL, Reyner Subrata, Rico Fransisco, Stephanus Jeffry, dan Steven Tjiardy sebagai teman bermain dan rekan seperjuangan di Unpar.
8. Seluruh teman-teman IT Unpar 2010.
9. Pihak-pihak yang membantu dan tidak dapat penulis sebutkan satu persatu.

Akhir kata penulis berharap agar skripsi ini dapat memberikan manfaat kepada orang-orang yang membaca. Penulis juga memohon maaf apabila terdapat kesalahan penulisan nama atau kata-kata yang kurang berkenan. Tuhan Yesus memberkati. Amin.

Bandung, Mei 2015

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xx
DAFTAR TABEL	xxiii
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi Penelitian	2
1.6 Sistematika Pembahasan	3
2 LANDASAN TEORI	5
2.1 Google OAuth 2.0 [1]	5
2.1.1 Langkah Dasar	5
2.1.2 Skenario Google OAuth	6
2.1.3 Masa Berlaku Token	11
2.1.4 Lingkup Otorisasi [2]	11
2.1.5 API <i>Client Library</i> untuk PHP [3]	14
2.2 Markdown	16
2.2.1 Apa itu Markdown? [4]	16
2.2.2 Sintaks yang Berguna [4]	16
2.2.3 GitHub Flavored Markdown [5]	23
2.3 StrapdownJS [6]	27
2.4 Zurb Foundation [7]	29
2.4.1 Kompatibilitas	29
2.4.2 Apa Saja yang Dapat Dibuat dengan Foundation?	29
3 ANALISIS	41
3.1 Analisis Kebutuhan Pengguna	41
3.2 Analisis Google OAuth	46
3.2.1 Langkah Dasar Penggunaan OAuth 2.0	47
3.2.2 Skenario Aplikasi	48
3.2.3 Penggunaan API <i>Client Library</i> untuk PHP	48
3.3 Analisis Markdown	52
3.4 Analisis StrapdownJS	55
3.5 Analisis Zurb Foundation	56
3.6 Analisis Kebutuhan Perangkat Lunak	58
3.6.1 Use Case Diagram	59

3.6.2	Skenario	61
3.6.3	<i>Entity Relationship</i> Diagram (ERD)	61
3.6.4	Data <i>Flow</i> Diagram	62
4	PERANCANGAN	69
4.1	Perancangan Tampilan <i>Web</i> Yang Digunakan	69
4.1.1	Tampilan Halaman Awal	69
4.1.2	Tampilan <i>Web</i> Pilih Mahasiswa	70
4.1.3	Tampilan <i>Web</i> Cari Mahasiswa	70
4.1.4	Tampilan <i>Web</i> Info Mahasiswa	71
4.1.5	Tampilan <i>Web</i> Edit Info Mahasiswa	71
4.1.6	Tampilan <i>Web</i> Tambah Masalah Baru	72
4.1.7	Tampilan <i>Web</i> Lihat Daftar Masalah	73
4.1.8	Tampilan <i>Web</i> Lihat Histori	74
4.1.9	Tampilan <i>Web</i> Lihat Versi Ini	74
4.1.10	Tampilan <i>Web</i> Entri Baru	74
4.2	Perancangan Modul	76
4.2.1	Modul Login	76
4.2.2	Modul Pilih Mahasiswa	76
4.2.3	Modul Cari Mahasiswa	76
4.2.4	Modul Info Mahasiswa	76
4.2.5	Modul Edit Mahasiswa	76
4.2.6	Modul Menambah Masalah Baru	77
4.2.7	Modul Lihat Daftar Masalah	77
4.2.8	Modul Lihat Histori	77
4.2.9	Modul Entri Baru	77
4.3	Perancangan Diagram Relasional	78
4.4	Perancangan Tabel Sistem Perekam dan Berbagi Riwayat Mahasiswa	78
4.4.1	Perancangan Tabel Info Mahasiswa	78
4.4.2	Perancangan Tabel Histori	78
4.4.3	Perancangan Tabel Masalah	78
5	IMPLEMENTASI DAN PENGUJIAN PERANGKAT LUNAK	81
5.1	Implementasi Perangkat Lunak	81
5.1.1	Lingkungan Implementasi Perangkat Keras	81
5.1.2	Lingkungan Implementasi Perangkat Lunak	81
5.1.3	Hasil Implementasi Perangkat Lunak	82
5.2	Implementasi Basis Data	82
5.3	Pengujian Perangkat Lunak	83
5.3.1	Lingkungan Pengujian Perangkat Keras	83
5.3.2	Lingkungan Pengujian Perangkat Lunak	83
5.3.3	Pengujian Fungsional	84
5.3.4	Pengujian Eksperimental	95
6	KESIMPULAN DAN SARAN	107
6.1	Kesimpulan	107
6.2	Saran	107
	DAFTAR REFERENSI	109
	A KODE PROGRAM	111
	B KODE PROGRAM UNTUK PENGUJIAN FUNGSIONAL	119

C WAWANCARA	127
D SURVEI KEBUTUHAN PENGGUNA	131
E KUESIONER PENGUJIAN EKSPERIMENTAL	133

DAFTAR GAMBAR

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

2.43	Tabel Foundation	39
2.44	Contoh Navigasi Zurb Foundation	39
2.45	Contoh Plugins Zurb Foundation	40
3.1	Google Developers Console	49
3.2	Membuat Proyek Baru	49
3.3	Menu Credentials	49
3.4	Membuat Client ID yang Baru	49
3.5	Tipe Aplikasi	50
3.6	Pengisian Tipe Aplikasi	50
3.7	Client ID	51
3.8	Izin Pihak Pengguna	51
3.9	Skenario Aplikasi SPBRM	52
3.10	Output Sintaks Cetak Tebal dan Cetak Miring	54
3.11	Output Sintaks Judul Bab	54
3.12	Output Sintaks Batas Baris	54
3.13	Output Sintaks Paragraf	54
3.14	Output Sintaks Link	54
3.15	Output Sintaks Daftar	54
3.16	Output Keterangan Mahasiswa	55
3.17	Tampilan list.php Dengan Ukuran Layar Penuh	58
3.18	Tampilan list.php Dengan Ukuran Layar Diperkecil	59
3.19	<i>Use Case</i> Diagram Sistem perekam dan Berbagi Riwayat Mahasiswa	60
3.20	<i>Entity Relationship</i> Diagram	67
3.21	<i>Data Context</i> Diagram Sistem perekam dan Berbagi Riwayat Mahasiswa	67
3.22	<i>Data Flow</i> Diagram Sistem perekam dan Berbagi Riwayat Mahasiswa	68
4.1	Desain Antarmuka Halaman Awal	69
4.2	Desain Antarmuka Pilih Mahasiswa	70
4.3	Desain Antarmuka Cari Mahasiswa	70
4.4	Desain Antarmuka Info Mahasiswa	71
4.5	Desain Antarmuka Edit Info Mahasiswa	72
4.6	Desain Antarmuka Tambah Masalah Baru	73
4.7	Desain Antarmuka Lihat Daftar Masalah	73
4.8	Desain Antarmuka Lihat Histori	74
4.9	Desain Antarmuka Lihat Versi Ini	75
4.10	Desain Antarmuka Entri Baru	75
4.11	Diagram Relasional	80
5.1	Membuka Halaman index.php	86
5.2	Login Dengan Email yang Diakhiri "@student.unpar.ac.id"	87
5.3	Konfirmasi Email yang Dikelola oleh student.unpar.ac.id	87
5.4	CAS UNPAR	87
5.5	Izin Akses Dari Pihak Pengguna	88
5.6	Login Dengan Email yang Diakhiri "@gmail.com"	88
5.7	Alert Email yang Digunakan Tidak Dapat Mengakses SPBRM	88
5.8	Memilih Mahasiswa	89
5.9	Mencari Mahasiswa Dengan NPM Seutuhnya	89
5.10	Hasil Mencari Mahasiswa Dengan NPM Seutuhnya	89
5.11	Mencari Mahasiswa Dengan Sebagian NPM	90
5.12	Hasil Mencari Mahasiswa Dengan Sebagian NPM	90
5.13	Melihat Info Mahasiswa	90

5.14 Mengedit Info Mahasiswa	91
5.15 Menambah Catatan Masalah Baru	91
5.16 Melihat Daftar Masalah	91
5.17 Melihat Histori	92
5.18 Keterangan Versi Pertama	92
5.19 Keterangan Versi Kedua	92
5.20 Membuat Entri Baru	93
5.21 Entri Baru Berhasil Dibuat	93
5.22 Antarmuka Responsif index.php	94
5.23 Antarmuka Responsif list.php	94
5.24 Antarmuka Responsif new.php	95
5.25 Penguji Pertama Mengakses Halaman Utama	96
5.26 Penguji Pertama Mengakses Info Mahasiswa Dengan NPM 2010730013	96
5.27 Penguji Pertama Mengakses <i>History</i> Mahasiswa Dengan NPM 2010730013	97
5.28 Penguji Pertama Melihat Versi Ini Untuk Mahasiswa Dengan NPM 2010730013	97
5.29 Penguji Kedua Mengakses Halaman Utama	98
5.30 Penguji Kedua Mengakses Info Mahasiswa Dengan NPM 2010730069	98
5.31 Penguji Kedua Mengakses <i>History</i> Mahasiswa Dengan NPM 2010730069	99
5.32 Penguji Pertama Mengakses Halaman Utama Untuk Kedua Kali	99
5.33 Penguji Pertama Mengakses Info Mahasiswa Dengan NPM 2010730071	100
5.34 Penguji Pertama Mengakses <i>History</i> Mahasiswa Dengan NPM 2010730071	100
5.35 Penguji Pertama Melihat Versi Ini Untuk Mahasiswa Dengan NPM 2010730071	101
5.36 Penguji Pertama Mengakses Info Mahasiswa Dengan NPM 2010730069	101
5.37 Penguji Pertama Mengakses <i>History</i> Mahasiswa Dengan NPM 2010730069	102
5.38 Penguji Ketiga Mengakses Info Mahasiswa Dengan NPM 2010730013	102
5.39 Penguji Ketiga Mengakses Info Mahasiswa Dengan NPM 2010730071	103
5.40 Penguji Ketiga Mengakses <i>History</i> Mahasiswa Dengan NPM 2010730071	103

DAFTAR TABEL

2.1	Tabel Daftar Pengujian Zurb Foundation	29
3.1	Tabel Jawaban Pertanyaan Pertama, Sisi Humanis	43
3.2	Tabel Jawaban Pertanyaan Kedua, Konsultasi Dengan Dosen Lain	44
3.3	Tabel Jawaban Pertanyaan Ketiga, Konsultasi Dengan Mahasiswa Lain	44
3.4	Tabel Jawaban Pertanyaan Keempat, Disiplin Menerapkan Aturan	45
3.5	Tabel Jawaban Pertanyaan Kelima, Penilaian Sayalah yang Paling Baik	45
3.6	Tabel Jawaban Pertanyaan Keenam, Menggunakan Akun Dosen	46
3.7	Tabel Jawaban Pertanyaan Ketujuh, Membuat Sebuah Format	46
3.8	Tabel Jawaban Pertanyaan Kedelapan, Penggunaan Perangkat Pribadi	47
3.9	Tabel Skenario <i>Login</i>	61
3.10	Tabel Skenario Pilih Mahasiswa	62
3.11	Tabel Skenario Cari Mahasiswa	63
3.12	Tabel Skenario Melihat Info Mahasiswa	64
3.13	Tabel Skenario Edit Mahasiswa	65
3.14	Tabel Skenario Menambah Masalah Baru	65
3.15	Tabel Skenario Lihat Daftar Masalah	66
3.16	Tabel Skenario Lihat Histori	66
3.17	Tabel Skenario Membuat Entri Baru	66
4.1	Tabel Modul Login	76
4.2	Tabel Modul Pilih Mahasiswa	76
4.3	Tabel Modul Cari Mahasiswa	77
4.4	Tabel Modul Info Mahasiswa	77
4.5	Tabel Modul <i>Edit</i> Mahasiswa	78
4.6	Tabel Modul Menambah Masalah Baru	78
4.7	Tabel Modul Lihat Daftar Masalah	79
4.8	Tabel Modul Lihat Histori	79
4.9	Tabel Modul Entri Baru	79
4.10	Tabel Rancangan Tabel Info Mahasiswa	79
4.11	Tabel Rancangan Tabel Histori	80
4.12	Tabel Rancangan Tabel Masalah	80
5.1	Tabel Jawaban Pertanyaan Pertama, SPBRM Membantu Mengingat	95
5.2	Tabel Jawaban Pertanyaan Kedua, Kemudahan SPBRM	104
5.3	Tabel Jawaban Pertanyaan Ketiga, SPBRM Membantu Pemahaman	104
5.4	Tabel Jawaban Pertanyaan Keempat, SPBRM Efektif	104
5.5	Tabel Jawaban Pertanyaan Kelima, SPBRM Efisien	104

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Di tahun 2014 terdapat 14 dosen Teknik Informatika di Falkutas Teknologi Informasi dan Sains [8] 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 melakukan login pada sistem dengan menggunakan akun dosen UNPAR. Kedua, dosen dapat melihat data mahasiswa yang telah terdaftar di basis data. Ketiga, dosen dapat mencari mahasiswa yang telat terdaftar di basis data berdasarkan npm. Keempat, dosen dapat melihat data informasi mahasiswa yang ingin dilihat. Kelima, dosen dapat mengubah data informasi mahasiswa. Keenam, dosen dapat membuat catatan masalah baru seorang mahasiswa. Ketujuh, dosen dapat melihat daftar masalah yang dimiliki seorang mahasiswa. Kedelapan, 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. Kesembilan, dosen dapat membuat entri baru untuk mahasiswa yang belum tercatat pada basis data.

Berdasarkan penjabaran solusi untuk menangani masalah, maka dari itu untuk membangun sistem usulan 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. Pengguna yang menggunakan akun selain akun dosen UNPAR tidak dapat mengakses sistem. Markdown Syntax akan digunakan untuk membuat informasi riwayat 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?
- Bagaimana mengukur tingkat keberhasilan sistem yang dibuat?

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.
- Mengukur tingkat keberhasilan sistem yang dibuat.

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.
- Melakukan pengambilan kesimpulan berdasarkan analisis dan pengujian yang telah dilakukan.

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 referensi utama dalam pelaksanaan penelitian.
- Bab III Analisis
Bab ini berisi analisis mengenai kebutuhan pengguna, Google OAuth, Markdown Syntax, StrapdownJS, Zurb Foundation dan kebutuhan perangkat lunak 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 2.0 [1]

API Google menggunakan protokol OAuth 2.0 untuk autentikasi 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 autentikasi (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 hanya 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 membutuhkan data yang rahasia, tapi layanan aplikasi *web* tentu membutuhkannya.
- (2) Memperoleh token akses dari Google Authorization Server
Sebelum data pribadi dapat diakses aplikasi yang menggunakan Google API, terlebih dahulu diperlukan token akses untuk mengakses API tersebut. Berbagai tingkat akses untuk mengakses beberapa API, diberikan dengan melakukan pertukaran suatu token akses. Izin token

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

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

akses merupakan parameter untuk variabel ruang lingkup yang mengontrol sumber daya dan operasi yang akan digunakan. 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 dan harus disesuaikan aplikasi yang dibangun. Contohnya aplikasi JavaScript, mesin pencari digunakan untuk mendapatkan token akses yang akan diarahkan 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 autentikasi dimana pengguna diharuskan login dengan menggunakan akun Google mereka. Setelah login pengguna akan ditanya apakah pengguna akan memberi izin untuk mengakses aplikasi yang telah melakukan permintaan tersebut. Proses ini disebut izin dari pihak pengguna. Jika pengguna memberi izin, maka Google Authorization Server akan mengirimkan aplikasi tersebut sebuah token akses. Jika pengguna tidak memberi izin, maka server akan menunjukkan respon yang menyatakan eror.

(3) Kirim token akses ke API

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

(4) Memperbaharui token akses jika diperlukan

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

2.1.2 Skenario Google OAuth

Terdapat lima skenario yang dapat digunakan untuk Google OAuth yaitu Skenario Aplikasi *Web* Server, Skenario Aplikasi yang Terinstal, Skenario Aplikasi Sisi Klien (JavaScript), Skenario Aplikasi Pada Perangkat Dengan Masukan yang Terbatas, dan Skenario Layanan Akun. Untuk penjelasan lebih lanjut dapat dilihat pada 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 autentikasi 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 token akses 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 autentikasi pengguna, pemilihan sesi, dan izin pengguna. Hasilnya adalah sebuah kode otorisasi yang dapat ditukar untuk mendapatkan 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.

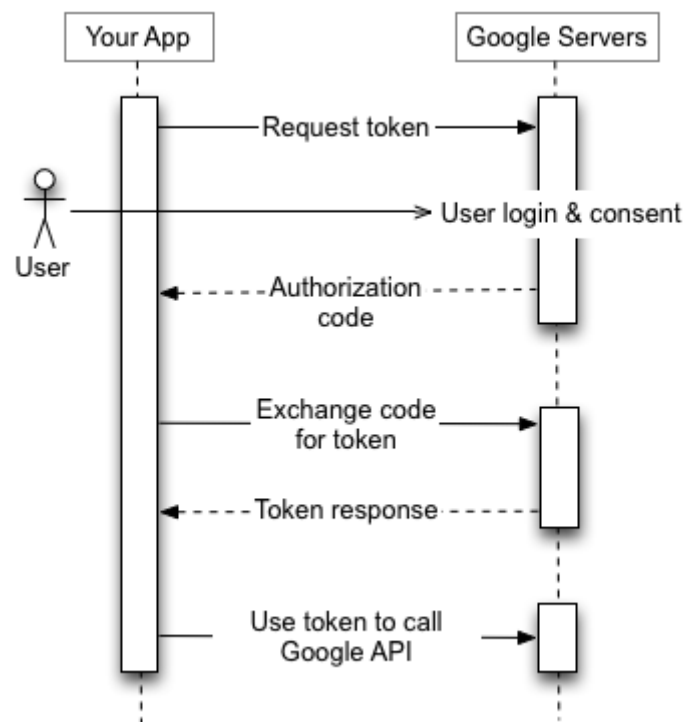
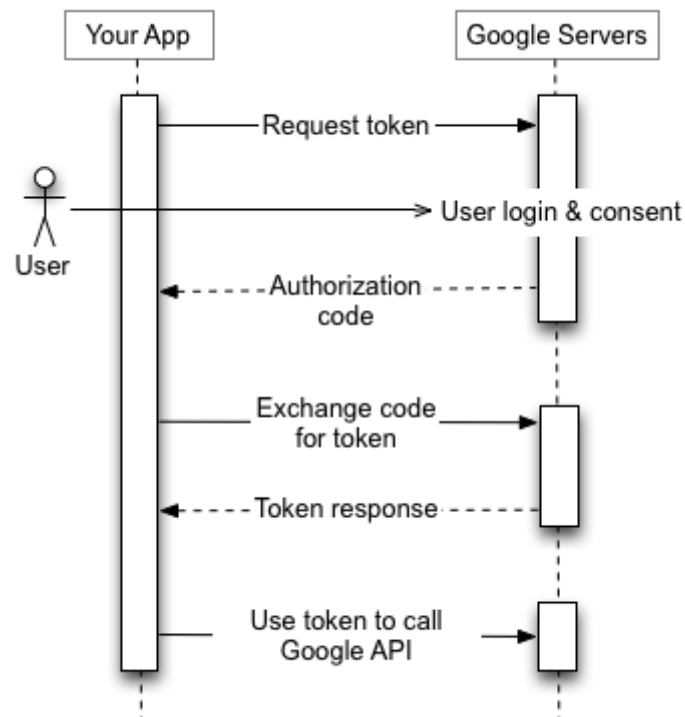
- Skenario Aplikasi Sisi Klien (JavaScript)

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 autentikasi pengguna, pemilihan sesi, dan izin pengguna. Hasilnya adalah token akses dimana klien harus melakukan validasi sebelum memasukkannya ke dalam permintaan Google API. Ketika masa token berakhir, aplikasi mengulangi proses. Untuk gambar skenario dapat dilihat pada Gambar 2.3.

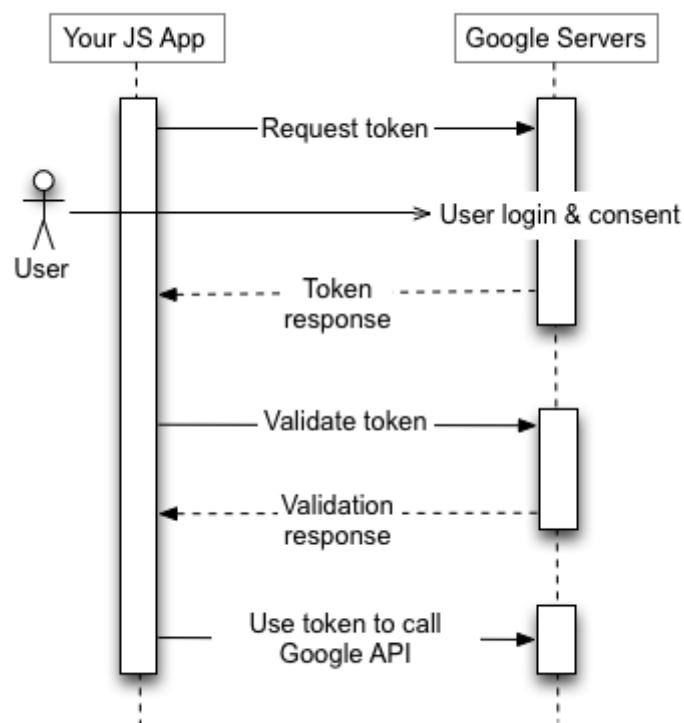
- Skenario Aplikasi Pada Perangkat Dengan Masukan Yang Terbatas

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, mengarahkan ke URL tertentu, melakukan *log in*, dan memasukan kode.

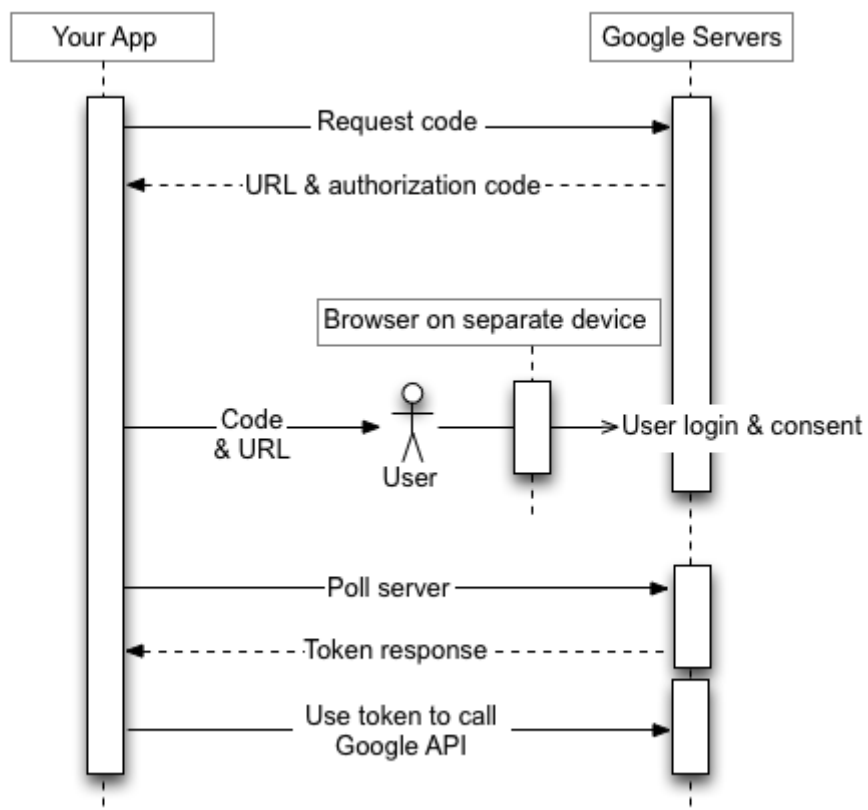
Sementara itu, aplikasi jajak pendapat dari URL Google pada interval tertentu. Setelah pengguna menyetujui akses, respon dari server Google berisi token akses dan memperbaharui token. Aplikasi harus menyimpan token yang baru untuk penggunaan masa depan dan menggunakan token akses untuk mengakses Google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

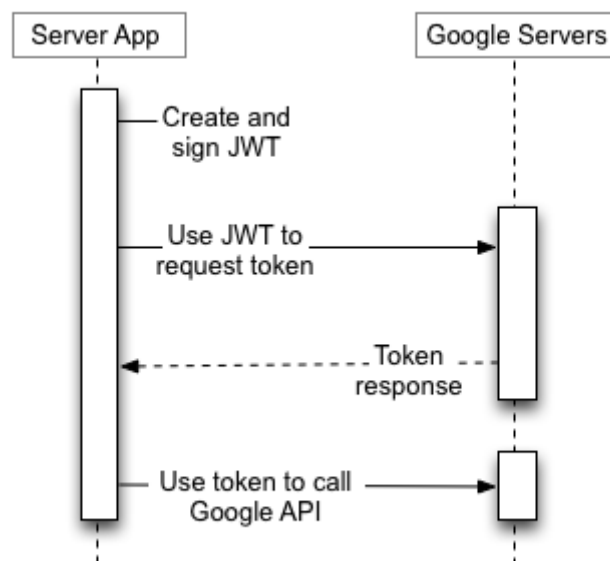
- Skenario Layanan Akun

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 yang 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 untuk tanda penciptaan dan penandatanganan secara abstrak tanpa menggunakan perpustakaan, mungkin akan membuat kesalahan yang memiliki dampak yang parah pada keamanan aplikasi yang dibangun.

Mandat untuk 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 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 autentikasi 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 melakukan otorisasi beberapa program, mesin, atau perangkat, salah satu solusi yang ditawarkan pihak Google adalah dengan membatasi jumlah klien, dimana satu pengguna harus melakukan otorisasi antara 15 hingga 20 akun. Jika anda adalah admin Google Apps, anda dapat membuat admin tambahan untuk mengizinkan beberapa klien.

2.1.4 Lingkup Otorisasi [2]

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 autentikasi (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 sebagai berikut:

- Meminta agar aplikasi diberikan akses ke informasi profil dasar bagi pengguna yang terautentikasi.

- Memungkinkan aplikasi untuk mengetahui siapa pengguna yang dikonfirmasi dengan mengganti id pengguna dengan "*me*" yang mewakili pengguna yang telah terautentikasi 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 terautentikasi.
- 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 terautentikasi 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 terautentikasi disetiap permintaan yang dilakukan.
- Lingkup yang tidak dipakai lagi

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

Lingkup diatas sudah tidak dipakau lagi diganti 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
```

Lingkup diatas sudah tidak dipakau lagi diganti 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 [3]

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 untuk 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 berbagai kondisi, tergantung pada akses yang digunakan merupakan akses sederhana atau akses yang memerlukan izin. Berikut dua metode untuk menentukan jenis akses.

- Akses sederhana API (kunci API)

Metode ini tidak dapat digunakan untuk mengakses data pribadi pengguna. Aplikasi harus mengautentikasi 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 mengautentikasi 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 atau mungkin telah membaca dan menulis 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 rahasia klien anda pribadi. Jika seseorang memperoleh rahasia 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 klien adalah wadah utama untuk kelas dan konfigurasi di *library*.

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

- Membangun layanan objek

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

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

- Memanggil API

Setiap API menyediakan sumber dan metode, biasanya terdapat di dalam rantai. Hal tersebut 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.

```

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

```

- Penanganan hasil

Terdapat dua jenis respon utama 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.

```

1| foreach ($results as $item) {
2|     echo $item['volumeInfo']['title'], "<br /> \n";
3| }

```

2.2 Markdown

2.2.1 Apa itu Markdown? [4]

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

2.2.2 Sintaks yang Berguna [4]

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>Alice melaporkan bahwa dirinya mengidap penyakit <em>Amyotrophic Lateral
   | Sclerosis </em> sehingga mengganggu proses belajar.</p>

```



```

2 |
3 | <p>Bob datang terlambat di kelas algoritma , dan <strong>menolak untuk keluar</
   | strong> saat diusir dosen kelas.</p>

```

Alice melaporkan bahwa dirinya mengidap penyakit *Amyotrophic Lateral Sclerosis* sehingga mengganggu proses belajar.

Bob datang terlambat di kelas algoritma, dan **menolak untuk keluar** saat diusir dosen kelas.

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 | # Bab 1
2 | ## Bab 1.1
3 | ### Bab 1.1.1
4 | #### Bab 1.1.1.1
5 | ##### Bab 1.1.1.1.1
6 | ##### Bab 1.1.1.1.1.1

```

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

```

1 | <h1>Bab 1</h1>
2 |
3 | <h2>Bab 1.1</h2>
4 |
5 | <h3>Bab 1.1.1</h3>
6 |
7 | <h4>Bab 1.1.1.1</h4>
8 |
9 | <h5>Bab 1.1.1.1.1</h5>
10 |
11 | <h6>Bab 1.1.1.1.1.1</h6>

```

Bab 1

Bab 1.1

Bab 1.1.1

Bab 1.1.1.1

Bab 1.1.1.1.1

Bab 1.1.1.1.1.1

Gambar 2.7: Markdown Judul Bab

- Batas Baris

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

```
1| Bob melihat pengumuman
2| kelas PBO ditiadakan
```

Contoh di atas, setelah kata 'pengumuman' 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>Bob melihat pengumuman <br />
2| kelas PBO ditiadakan </p>
```

Bob melihat pengumuman
kelas PBO ditiadakan

Gambar 2.8: Markdown Batas Baris Dengan Tiga Spasi

```
1| Alice menunggu
2| kelas PBO
```

Contoh di atas, setelah kata 'menunggu' 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>Alice menunggu
2| kelas PBO</p>
```

Alice menunggu kelas PBO

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 | Alice memberikan tugas yang telah dititipkan Bob. Bob merasa Alice baik
   | terhadap dirinya. Maka Bob mengirim pesan ke Alice.
2 |
3 | Ternyata Alice mendapat hadiah dari tugas yang dititipkan oleh Bob.

```

Contoh diatas baris pertama adalah paragraf 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>Alice memberikan tugas yang telah dititipkan Bob. Bob merasa Alice baik
   | terhadap dirinya. Maka Bob mengirim pesan ke Alice.</p>
2 |
3 | <p>Ternyata Alice mendapat hadiah dari tugas yang dititipkan oleh Bob.</p>

```

Alice memberikan tugas yang telah dititipkan Bob. Bob merasa Alice baik terhadap dirinya. Maka Bob mengirim pesan ke Alice.
Ternyata Alice mendapat hadiah dari tugas yang dititipkan oleh Bob.

Gambar 2.10: Markdown Paragraf

- Gambar

Untuk menyisipkan gambar pada dokumen Markdown, menggunakan 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 | Bob mendefinisikan fungsi penjumlahan()
2 |
3 | '''
4 |     fungsi penjumlahan() {
5 |         A + 1 = C;
6 |         2 + B = D
7 |     }
8 | '''
9 | 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>Bob mendefinisikan fungsi penjumlahan()</p>
2 |
3 | <p>'''
4 |     fungsi penjumlahan() {
5 |         A + 1 = C;
6 |         2 + B = D
7 |     }
8 | '''</p>
```

```
Bob mendefinisikan fungsi penjumlahan()
''' fungsi penjumlahan() { A + 1 = C; 2 + B = D } '''
```

Gambar 2.13: Markdown Kode

- Kutipan

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

```
1 | Alice mempunyai *quote* favorit :
2 |
3 | > Kalau kamu mau, kamu pasti bisa
```

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

```
1 | <p>Alice mempunyai <em>quote</em> favorit:</p>
2 |
3 | <blockquote>
4 |   <p>Kalau kamu mau, kamu pasti bisa</p>
5 | </blockquote>
```

Alice mempunyai quote favorit:
Kalau kamu mau, kamu pasti bisa

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 | # Pertemuan pertama kelas ASD
2 |
3 | Bob dapat mengerti materi Tree dan Hash dengan cepat
4 |
5 | —
6 |
7 | Diubah pada tanggal *13 Januari 2015*
```

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>Pertemuan pertama kelas ASD</h1>
2 |
3 | <p>Bob dapat mengerti materi <strong>Tree</strong> dan <strong>Hash</strong>
   |   dengan cepat</p>
4 |
5 | <hr />
6 |
7 | <p>Diubah pada tanggal <em>13 Januari 2015</em></p>
```

Pertemuan pertama kelas ASD

Bob dapat mengerti materi **Tree** dan **Hash** dengan cepat

Diubah pada tanggal 13 Januari 2015

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 * 13 Januari, Bob terlambat masuk kelas PBW
2 * 14 Januari, Bob terlambat masuk kelas PBO
3 * 15 Januari, Bob terlambat masuk kelas IMK
4
5 + 16 Januari, Bob terlambat masuk kelas PBD
6 + 17 Januari, Bob terlambat masuk kelas ASD
7 + 18 Januari, Bob terlambat masuk kelas PBW ke 2
8
9 - 19 Januari, Bob terlambat masuk kelas IMK ke 2
10 - 20 Januari, Bob terlambat masuk kelas ASD ke 2
11 - 21 Januari, Bob terlambat masuk kelas PBO ke 2
```

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

```
1 <ul>
2 <li>13 Januari, Bob terlambat masuk kelas PBW</li>
3 <li>14 Januari, Bob terlambat masuk kelas PBO</li>
4 <li><p>15 Januari, Bob terlambat masuk kelas IMK</p></li>
5 <li><p>16 Januari, Bob terlambat masuk kelas PBD</p></li>
6 <li>17 Januari, Bob terlambat masuk kelas ASD</li>
7 <li><p>18 Januari, Bob terlambat masuk kelas PBW ke 2</p></li>
8 <li><p>19 Januari, Bob terlambat masuk kelas IMK ke 2</p></li>
9 <li>20 Januari, Bob terlambat masuk kelas ASD ke 2</li>
10 <li>21 Januari, Bob terlambat masuk kelas PBO ke 2</li>
11 </ul>
```

- 13 Januari, Bob terlambat masuk kelas PBW
- 14 Januari, Bob terlambat masuk kelas PBO
- 15 Januari, Bob terlambat masuk kelas IMK
- 16 Januari, Bob terlambat masuk kelas PBD
- 17 Januari, Bob terlambat masuk kelas ASD
- 18 Januari, Bob terlambat masuk kelas PBW ke 2
- 19 Januari, Bob terlambat masuk kelas IMK ke 2
- 20 Januari, Bob terlambat masuk kelas ASD ke 2
- 21 Januari, Bob terlambat masuk kelas PBO ke 2

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. Bob aktif di dalam kelas PBWL
2 | 2. Alice datang terlambat pada kelas PBWL
3 | 3. Carol mendapat nilai tertinggi untuk tugas akhir PBWL
```

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

```
1 |
2 | Bob aktif di dalam kelas PBWL
3 | Alice datang terlambat pada kelas PBWL
4 | Carol mendapat nilai tertinggi untuk tugas akhir PBWL
5 |
```

1. Bob aktif di dalam kelas PBWL
2. Alice datang terlambat pada kelas PBWL
3. Carol mendapat nilai tertinggi untuk tugas akhir PBWL

Gambar 2.17: Markdown Daftar Berurutan

### 2.2.3 GitHub Flavored Markdown [5]

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 diantara garis bawah akan Diubah 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| Bob_mempunyai_perangkat_canggih
2| terlihat_dari_apa_yang_dia_gunakan.
```

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.

**Bob\_mempunyai\_perangkat\_canggih**  
**terlihat\_dari\_apa\_yang\_dia\_gunakan.**

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://www.unpar.ac.id
```

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

<http://www.unpar.ac.id>

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| ~~Dibuka kelas ASD pada masa SP.~~
```

Untuk hasil pada HTML dapat dilihat pada Gambar 2.20.

~~Dibuka kelas ASD pada masa SP.~~

Gambar 2.20: GFM Tanda Coret

- Blok Kode

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

```
1| Berikut kode **Java**
2|
3| ```
4| public class HelloWorld
```



```

5 | {
6 | public static void main(String [] args) {
7 | System.out.println(" Hello World!");
8 | }
9 | }
10| '''

```

Untuk hasil pada HTML dapat dilihat pada Gambar 2.21.

Berikut kode **Java**

```

public class HelloWorld
{
 public static void main(String[] args) {
 System.out.println("Hello World!");
 }
}

```

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 Java.

```

1 | '''java
2 | public class HelloWorld
3 | {
4 | public static void main(String [] args) {
5 | System.out.println(" Hello World!");
6 | }
7 | }
8 | '''

```

Untuk hasil pada HTML dapat dilihat pada Gambar 2.22.

```

public class HelloWorld
{
 public static void main(String[] args) {
 System.out.println("Hello World!");
 }
}

```

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 | No | Nama
2 | — | —
3 | 1 | Bob
4 | 2 | Alice

```

Untuk hasil pada HTML dapat dilihat pada Gambar 2.23.

| No | Nama  |
|----|-------|
| 1  | Bob   |
| 2  | Alice |

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 |  | No |  | Nama  |  |
| 2 |  | —  |  | —     |  |
| 3 |  | 1  |  | Bob   |  |
| 4 |  | 2  |  | Alice |  |

Untuk hasil pada HTML dapat dilihat pada Gambar 2.24.

| No | Nama  |
|----|-------|
| 1  | Bob   |
| 2  | Alice |

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 |  | Nama  |  | Mata Kuliah yang Diambil |  |
| 2 |  | —     |  | —                        |  |
| 3 |  | Alice |  | RPL                      |  |
| 4 |  | Bob   |  | PBWL                     |  |

Untuk hasil pada HTML dapat dilihat pada Gambar 2.25.

| Nama  | Mata Kuliah yang Diambil |
|-------|--------------------------|
| Alice | RPL                      |
| Bob   | PBWL                     |

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 | Nama | Deskripsi |
2 | ----- | ----- |
3 | Alice | ~~Terlambat~~ di setiap kelas. |
4 | Bob | Seharusnya Bob _menjemput_ Alice |

```

Untuk hasil pada HTML dapat dilihat pada Gambar 2.26.

| Nama  | Deskripsi                             |
|-------|---------------------------------------|
| Alice | <del>Terlambat</del> di setiap kelas. |
| Bob   | Seharusnya Bob <i>menjemput</i> Alice |

Gambar 2.26: GFM Tabel 4

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

```

1 | Rata Kiri | Rata Tengah | Rata Kanan |
2 | :-----: | :-----: | :-----: |
3 | Google | Bob | PBO |
4 | Mozilla | Carol | PBW |
5 | Opera | Alice | PBD |

```

Untuk hasil pada HTML dapat dilihat pada Gambar 2.27.

| Rata Kiri | Rata Tengah | Rata Kanan |
|-----------|-------------|------------|
| Google    | Bob         | PBO        |
| Mozilla   | Carol       | PBW        |
| Opera     | Alice       | PBD        |

Gambar 2.27: GFM Tabel 5

## 2.3 StrapdownJS [6]

Strapdown.js merupakan sebuah teknologi untuk membuat dokumen Markdown menjadi lebih sederhana namun tetap elegan. Penggunaannya 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 tempor
 incididunt ut labore
11|et dolore magna aliqua.
12|
13|## Chapter 2
14|
15|Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut
16|aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate
 velit esse
17|cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non
 proident, sunt in
18|culpa qui officia deserunt mollit anim id est laborum.
19|</xmp>
20|
21|<script src="http://strapdownjs.com/v/0.2/strapdown.js"></script>
22|</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)

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

| <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               | ✓               | ✓                | ✓         |

(3) Github menggunakan Markdown (Tabel, Sintaks, Judul Bab)

(4) Dapat menggunakan tema

## 2.4 Zurb Foundation [7]

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?

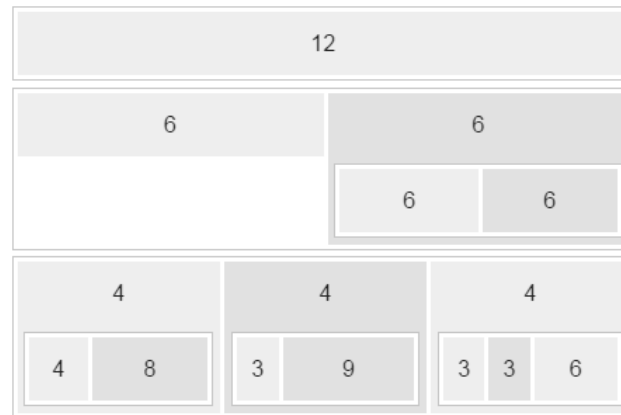
Foundation memiliki banyak komponen dan struktur untuk membantu membangun sebuah situs responsif. Untuk komponen Foundation dapat melihat beberapa gambar dibawah ini:

(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

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

waktu yang singkat dapat menciptakan tata letak yang kompleks seperti ini. Untuk contoh *grid* dapat dilihat pada Gambar 2.28.



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

```

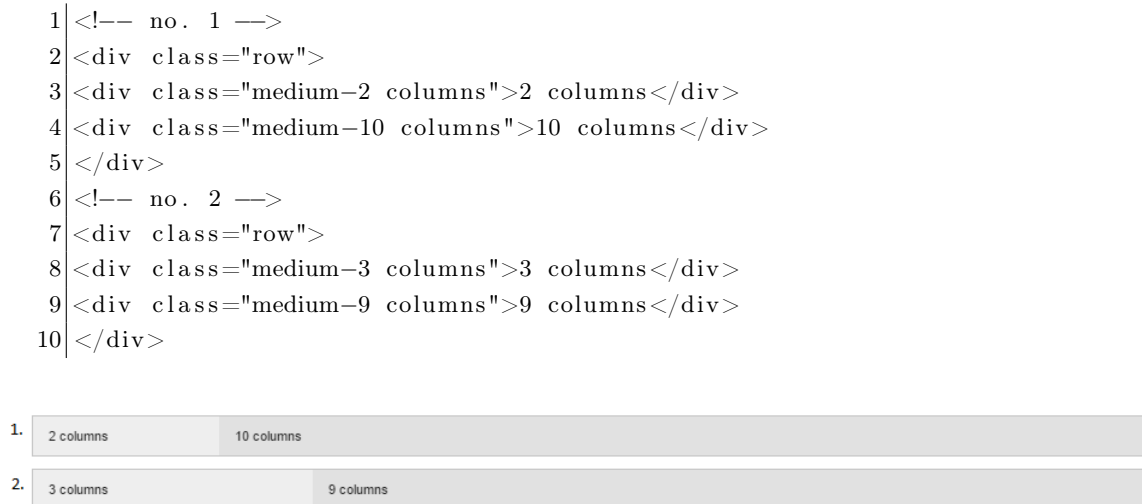
1 |<!-- no. 1 -->
2 |<div class="row">
3 |<div class="small-2 columns">2 columns</div>
4 |<div class="small-10 columns">10 columns</div>
5 |</div>
6 |<!-- no. 2 -->
7 |<div class="row">
8 |<div class="small-3 columns">3 columns</div>
9 |<div class="small-9 columns">9 columns</div>
10|</div>

```

Gambar 2.30: *Grid Kecil*

- *Grid* Menengah

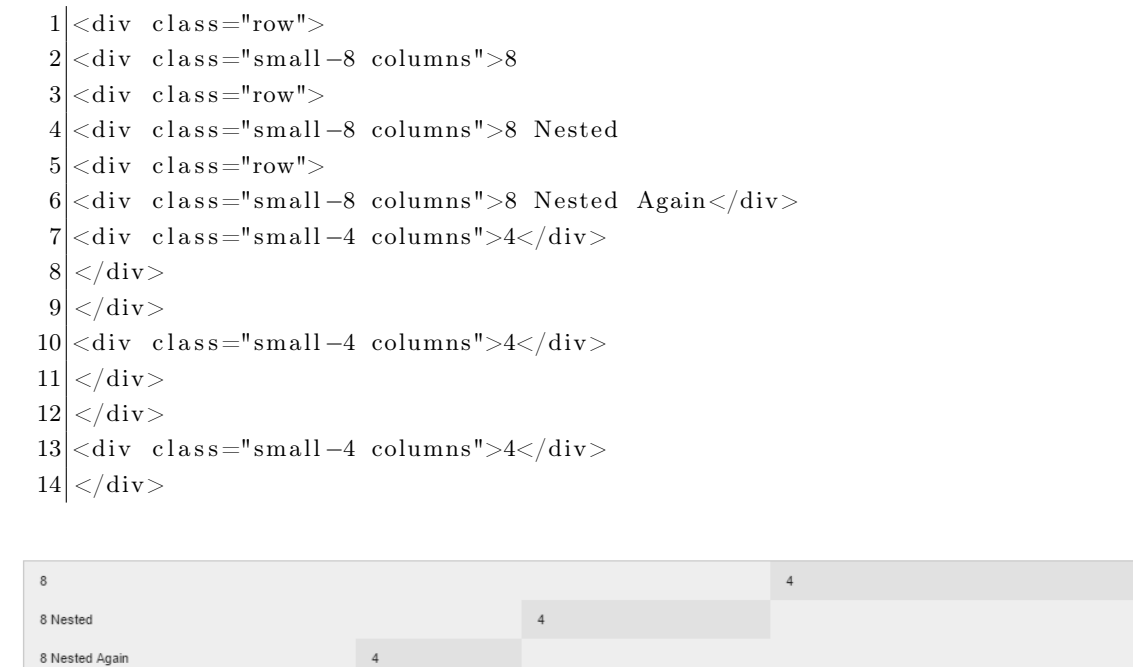
Layar berukuran sedang akan mendapatkan bagian dari gaya yang kecil, kecuali ditentukan tata letak yang berbeda dengan menggunakan *grid* menengah. Berikut kode HTML untuk membuat *grid* menengah dan dapat dilihat hasilnya pada Gambar 2.31.



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.



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 kondisi dimana 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.

```

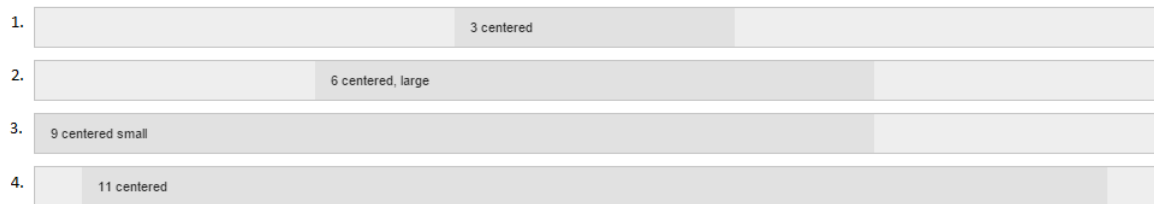
1 <!-- no. 1 -->
2 <div class="row">
3 <div class="small-3 small-centered columns">3 centered</div>
4 </div>
5 <!-- no. 2 -->
6 <div class="row">
7 <div class="small-6 large-centered columns">6 centered</div>
8 </div>
9 <!-- no. 3 -->
10 <div class="row">
11 <div class="small-9 small-centered large-uncentered 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, last</div>
20 </div>
21 <!-- no. 5 -->
22 <div class="row">
23 <div class="medium-6 medium-push-6 columns">6</div>
24 <div class="medium-6 medium-pull-6 columns">6, last</div>
25 </div>

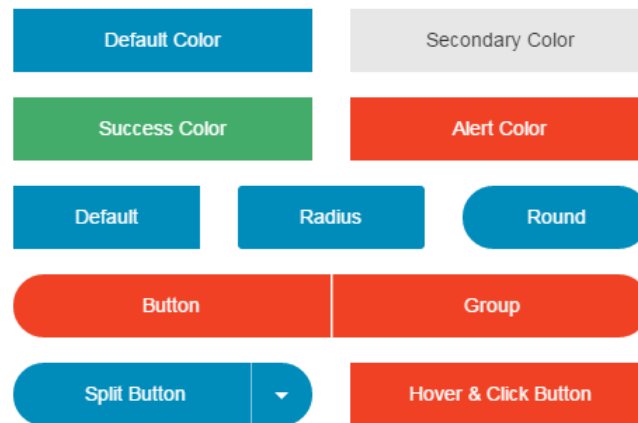
```

|    |         |    |
|----|---------|----|
| 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 | Default Button
```



Gambar 2.39: Tombol Dasar

- Mengubah Ukuran Tombol

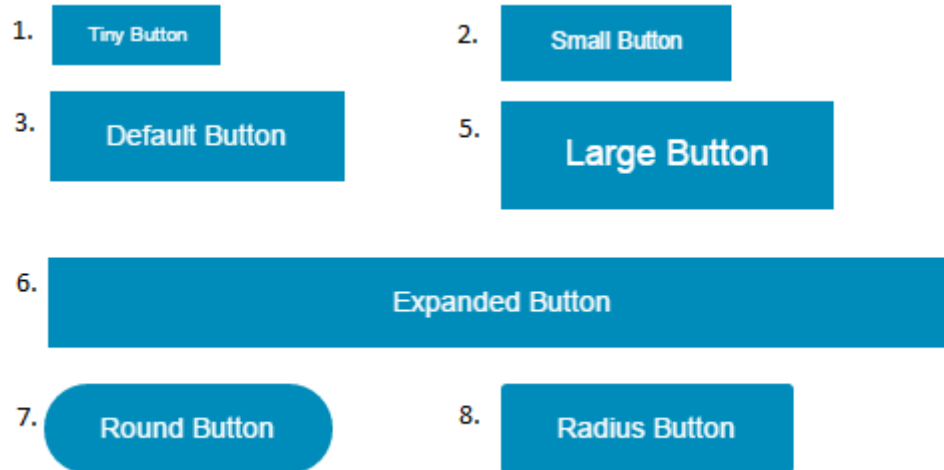
Terdapat kelas tambahan untuk merubah bentuk dan ukuran tombol, berikut kode HTML yang dapat digunakan dan dapat dilihat hasilnya pada Gambar 2.40.

```
1 | <!-- Kelas untuk merubah ukuran -->
2 | 1. Tiny Button
3 | 2. Small Button
```

```

4 | 3. Default Button
5 | 4. Disabled Button
6 | 5. Large Button
7 | 6. Expanded Button
8 | <!-- Kelas untuk merubah bentuk -->
9 | 7. Round Button
10| 8. Radius Button

```



Gambar 2.40: Ukuran dan Bentuk Tombol

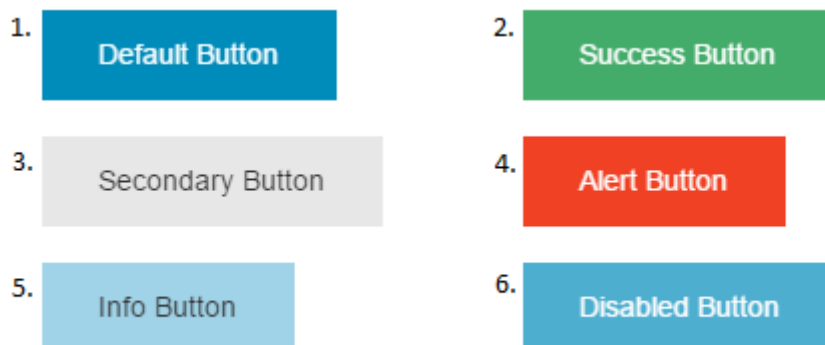
- Warna Tombol

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

```

1 | <!-- Kelas untuk merubah warna -->
2 | 1. Default Button
3 | 2. Success Button
4 | 3. Secondary Button
5 | 4. Alert Button
6 | 5. Info Button
7 | 6. Disabled Button

```



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 `<a href="">` maka cukup menambahkan `tabindex="0"` pada `div` atau `span` untuk membuat tombol tersebut dapat difokuskan.

```

1 | 1. Default Button
2 | 2. Submit
 |
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>

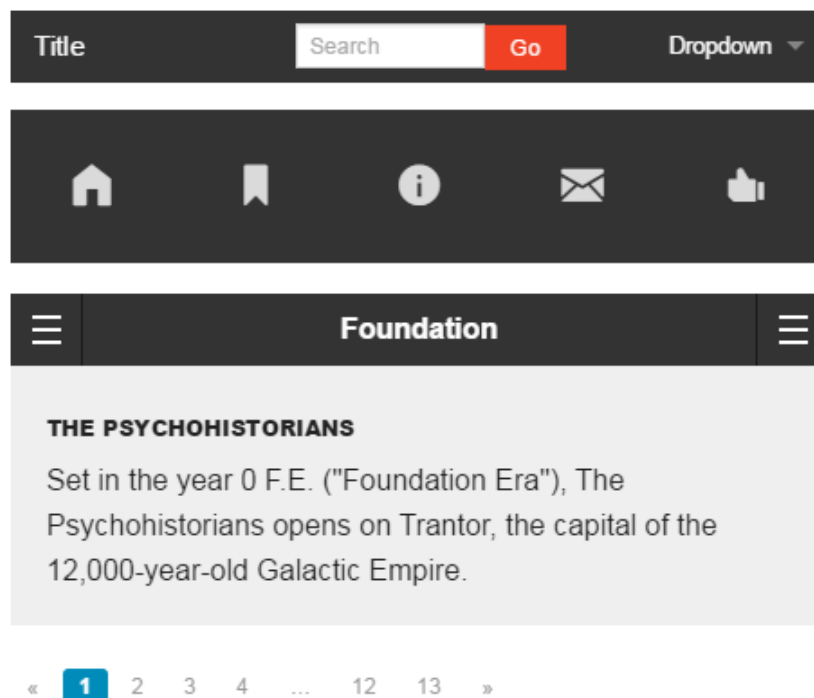
```

| 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

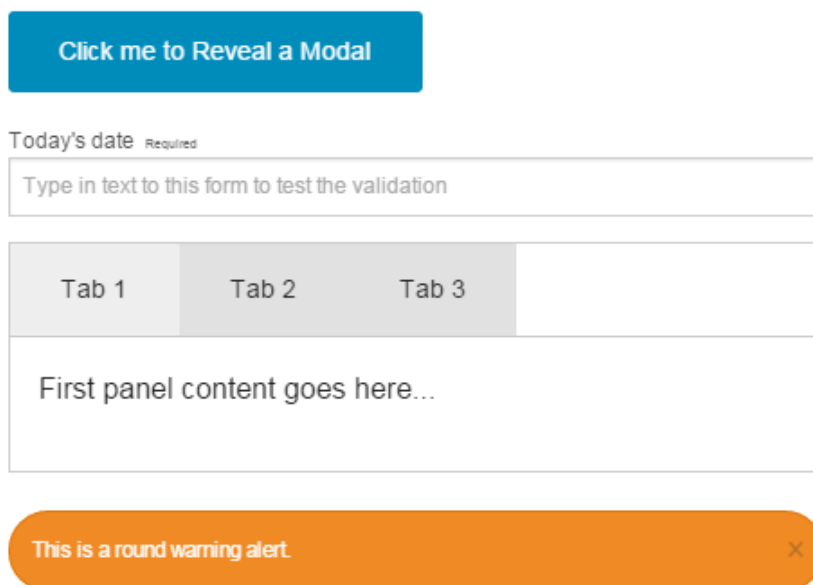
Orang yang mengakses harus bisa berkeliling melihat menu-menu yang ada. Gaya navigasi pada Foundation meliputi : bar bagian atas yang kuat dengan menu dropdown; tombol; bar pencari; ikon bar yang keren; implementasi kanvas yang lepas dari keluhan; dan sekelompok navigasi lainnya. Untuk contoh macam-macam 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 Kebutuhan Perangkat Lunak. 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 wawancara dan kuesioner. Wawancara dilakukan secara langsung ke 6 dosen Teknik Informatika. Untuk data dari hasil wawancara yang terdapat pada Lampiran C dapat dilihat di bawah ini.

- Rata-rata dosen Teknik Informatika UNPAR memiliki 25-50 anak wali.
- Masalah yang dialami pada saat menjadi dosen wali antara lain :
  - Administrasi mengenai syarat tanda tangan pada saat perwalian.
  - Mahasiswa tidak memiliki sanksi jika tidak datang perwalian.
  - Tidak mengingat semua riwayat mahasiswa.
  - Ada persoalan tapi tidak cerita.
  - Ada mahasiswa yang tidak hadir pada waktu perwalian.
  - Harus mengingat *track record* masalah.
  - Sulit menghubungi mahasiswa.
  - Ada mahasiswa yang menitipkan perwalian ke teman atau kontak via email atau whatsapp sehingga kurang intensif.
- Perwalian yang dilakukan sudah optimal

- Satu orang dosen menjawab sudah tercapai.
  - Sisanya masih merasa belum.
- Dibuatkan sebuah perangkat lunak yang membantu dalam mencatat dan membagikan riwayat mahasiswa dan semua dosen menjawab setuju.
- Fitur yang diharapkan :
  - Mendukung faktor efisien.
  - Kebebasan dalam mencatat apa pun.
  - Komentar positif dan komentar negatif dipisahkan.
  - Catatan khusus dosen wali.
  - Mudah diakses.
  - Form mengakses masalah mahasiswa.
  - Form daftar kehadiran kelas.

Berikut analisis yang dihasilkan dari wawancara kebutuhan pengguna :

- Penelitian ini akan membantu dosen dalam menghadapi masalah pada saat menjadi dosen wali. Masalah yang akan dibantu antara lain; tidak mengingat semua riwayat mahasiswa, ada persoalan tetapi tidak cerita, harus mengingat *track record* masalah. Karena masalah-masalah tersebut sesuai dengan ide yang diusulkan untuk membuat sebuah perangkat lunak yang membantu dalam mencatat dan membagikan riwayat mahasiswa. Sedangkan untuk masalah seperti administrasi mengenai syarat tanda tangan pada saat perwalian, mahasiswa tidak memiliki sanksi jika tidak datang perwalian, ada mahasiswa yang tidak hadir pada waktu perwalian, sulit menghubungi mahasiswa, dan ada mahasiswa yang perwalian menitipkan ke teman atau kontak via email atau whatsapp sehingga kurang intensif. Alangkah lebih baik untuk mengatasi masalah tersebut, langsung diselesaikan oleh pihak jurusan, apakah masih diperlukan tanda tangan pada saat perwalian, apakah akan diberikan sanksi untuk mahasiswa yang tidak melakukan perwalian, apakah mahasiswa wajib menghadap dosen wali pada saat perwalian sehingga intensif.
- Penelitian ini juga bertujuan untuk membuat proses perwalian yang dilakukan oleh seorang dosen wali menjadi optimal karena sebagian besar dosen wali masih merasa proses perwalian yang dilakukannya belum optimal. Karena ide usulan untuk membuat sebuah perangkat lunak yang membantu dalam mencatat dan membagikan riwayat mahasiswa, akan mempermudah untuk mengetahui pendapat dosen lain mengenai seorang mahasiswa tanpa perlu mendatangi dosen lain untuk membicarakan hal tersebut. Kemudian dosen wali yang memiliki keterbatasan waktu dapat mengakses informasi anak walinya dengan cepat. Dosen wali juga akan mengetahui masalah yang dimiliki anak walinya tanpa perlu menunggu anak walinya menceritakan kepadanya.
- Para dosen setuju untuk dibuatkan sebuah perangkat lunak yang membantu dalam mencatat dan membagikan riwayat mahasiswa karena hal tersebut merupakan ide yang baik, menarik,

merasa akan tertolong dan terbantu sekali apalagi dijaga oleh setiap dosen wali agar selalu *uptodate*.

- Fitur yang akan diimplementasi pada perangkat lunak yang akan dibangun sesuai dengan fitur yang diharapkan dosen adalah mendukung faktor efisien, kebebasan dalam mencatat apa pun, komentar positif dan komentar negatif dipisah, mudah diakses, dan form mengakses masalah mahasiswa. Mendukung faktor efisien akan diimplementasikan dengan menampilkan deskripsi umum, catatan, daftar masalah dan daftar histori. Kebebasan dalam mencatat apa pun akan diimplementasikan dengan menyediakan form untuk memasukkan data informasi mahasiswa, kemudian pengguna dapat mengisikan data-data deskripsi umum, catatan dan masalah pada tempatnya. Komentar positif dan komentar negatif akan diimplementasikan dengan membuat deskripsi umum dan catatan sebagai komentar positif yang terpisah dari masalah sebagai komentar negatif. Mudah diakses diimplementasikan dengan semua dosen mendapatkan semua informasi yang dimiliki setiap mahasiswa. Form mengakses masalah diimplementasikan dengan membuatkan sebuah form masalah untuk membuat laporan sebuah masalah yang dimiliki seorang mahasiswa. Untuk fitur catatan khusus dosen wali tidak akan diimplementasikan pada perangkat lunak karena perangkat lunak yang akan dibangun akan mencatat dan membagikan setiap riwayat mahasiswa ke setiap dosen. Untuk fitur form daftar kehadiran kelas tidak akan diimplementasikan pada perangkat lunak karena perangkat lunak yang akan dibangun bertujuan untuk membantu seorang dosen wali untuk mengetahui deskripsi umum, catatan dan masalah yang dimiliki anak walinya.

Kuesioner yang disebarkan sebanyak 10 kuesioner dan yang menjadi responden adalah dosen Teknik Informatika dari Fakultas Teknik Informasi dan Sains. Untuk kuesioner kebutuhan pengguna dapat dilihat pada Lampiran D. Hasil kuesioner dari 10 orang dosen tersebut menghasilkan data yang dapat dilihat pada Tabel 3.1-3.8.

Tabel 3.1: Tabel Jawaban Pertanyaan Pertama, Sisi Humanis

| Responden | Sangat Setuju | Setuju | Netral | Tidak Setuju | Sangat Tidak Setuju |
|-----------|---------------|--------|--------|--------------|---------------------|
| 1         |               | ✓      |        |              |                     |
| 2         |               | ✓      |        |              |                     |
| 3         | ✓             |        |        |              |                     |
| 4         | ✓             |        |        |              |                     |
| 5         |               | ✓      |        |              |                     |
| 6         |               | ✓      |        |              |                     |
| 7         | ✓             |        |        |              |                     |
| 8         |               |        | ✓      |              |                     |
| 9         |               | ✓      |        |              |                     |
| 10        |               | ✓      |        |              |                     |

Berikut analisis data yang dihasilkan dari kuesioner kebutuhan pengguna :

- Sebagian besar responden setuju mempertimbangkan sisi humanis seorang mahasiswa diluar prestasi akademis dalam memutuskan sesuatu terhadap seorang mahasiswa, hal tersebut menunjukkan adanya kebutuhan akan sebuah pendukung (baik perangkat lunak maupun bukan). Misalnya bagi dosen untuk menghitung nilai "belas kasihan" pada kasus seperti nilai 79 berharap jadi A yang mempertimbangkan aspek sering terlambat masuk kelas dan sering terlambat

Tabel 3.2: Tabel Jawaban Pertanyaan Kedua, Konsultasi Dengan Dosen Lain

| Responden | Sangat Setuju | Setuju | Netral | Tidak Setuju | Sangat Tidak Setuju |
|-----------|---------------|--------|--------|--------------|---------------------|
| 1         |               | ✓      |        |              |                     |
| 2         |               | ✓      |        |              |                     |
| 3         |               | ✓      |        |              |                     |
| 4         |               |        | ✓      |              |                     |
| 5         |               | ✓      |        |              |                     |
| 6         | ✓             |        |        |              |                     |
| 7         |               | ✓      |        |              |                     |
| 8         |               |        | ✓      |              |                     |
| 9         |               |        | ✓      |              |                     |
| 10        |               |        | ✓      |              |                     |

Tabel 3.3: Tabel Jawaban Pertanyaan Ketiga, Konsultasi Dengan Mahasiswa Lain

| Responden | Sangat Setuju | Setuju | Netral | Tidak Setuju | Sangat Tidak Setuju |
|-----------|---------------|--------|--------|--------------|---------------------|
| 1         |               |        | ✓      |              |                     |
| 2         |               |        | ✓      |              |                     |
| 3         |               |        |        | ✓            |                     |
| 4         |               |        | ✓      |              |                     |
| 5         |               | ✓      |        |              |                     |
| 6         |               |        |        | ✓            |                     |
| 7         |               | ✓      |        |              |                     |
| 8         |               |        | ✓      |              |                     |
| 9         |               |        |        | ✓            |                     |
| 10        |               |        | ✓      |              |                     |

mengumpulkan tugas. Maka dari itu pada perangkat lunak yang akan dibangun, dosen dapat memasukkan sisi humanis seorang mahasiswa yang diketahui oleh dosen tersebut.

- Setengah dari responden setuju untuk berkonsultasi dengan rekan dosen dalam memutuskan sesuatu terhadap seorang mahasiswa. Hal tersebut menunjukkan adanya kebutuhan dosen untuk berkomunikasi dengan dosen lain sebelum memutuskan sesuatu. Misalnya seorang mahasiswa sering tidak masuk pada suatu kelas karena alasan sakit, maka seorang dosen dapat menanyakan ke dosen lain apakah mahasiswa tersebut sering izin sakit tidak di kelas lain. Maka dari itu pada perangkat lunak yang akan dibangun, semua riwayat mahasiswa akan dibagikan ke setiap dosen.
- Sebagian besar responden merasa netral untuk mencari informasi dari mahasiswa lain dalam memutuskan sesuatu terhadap seorang mahasiswa, tetapi responden yang menyatakan tidak setuju lebih banyak dari responden yang menyatakan setuju. Maka dari itu pada penelitian ini tidak menangani aspek dalam mencari informasi dari mahasiswa lain.
- Sebagian besar setuju untuk disiplin menerapkan aturan yang berlaku dalam memutuskan sesuatu terhadap seorang mahasiswa. Hal tersebut menunjukkan bahwa dosen teknik infomatika disiplin dalam menerapkan aturan. Jadi setelah melihat riwayat mahasiswa yang terdapat pada perangkat lunak yang akan dibangun. Dosen dapat mengambil keputusan sesuai dengan aturan yang berlaku, misalnya dalam memberikan toleransi terhadap seorang mahasiswa yang

Tabel 3.4: Tabel Jawaban Pertanyaan Keempat, Disiplin Menerapkan Aturan

| Responden | Sangat Setuju | Setuju | Netral | Tidak Setuju | Sangat Tidak Setuju |
|-----------|---------------|--------|--------|--------------|---------------------|
| 1         |               |        | ✓      |              |                     |
| 2         |               | ✓      |        |              |                     |
| 3         |               | ✓      |        |              |                     |
| 4         | ✓             |        |        |              |                     |
| 5         |               |        | ✓      |              |                     |
| 6         |               | ✓      |        |              |                     |
| 7         |               | ✓      |        |              |                     |
| 8         |               | ✓      |        |              |                     |
| 9         | ✓             |        |        |              |                     |
| 10        |               | ✓      |        |              |                     |

Tabel 3.5: Tabel Jawaban Pertanyaan Kelima, Penilaian Sayalah yang Paling Baik

| Responden | Sangat Setuju | Setuju | Netral | Tidak Setuju | Sangat Tidak Setuju |
|-----------|---------------|--------|--------|--------------|---------------------|
| 1         |               |        |        | ✓            |                     |
| 2         |               |        |        |              | ✓                   |
| 3         |               |        | ✓      |              |                     |
| 4         |               |        | ✓      |              |                     |
| 5         |               |        |        | ✓            |                     |
| 6         |               |        | ✓      |              |                     |
| 7         |               |        | ✓      |              |                     |
| 8         |               |        | ✓      |              |                     |
| 9         |               | ✓      |        |              |                     |
| 10        |               |        | ✓      |              |                     |

terlambat mengumpulkan tugas.

- Sebagian besar responden merasa netral dalam aspek penilaian sayalah yang paling baik. Namun responden yang menyatakan tidak setuju lebih banyak dari responden yang menyatakan setuju. Hal tersebut menunjukkan adanya kebutuhan dosen untuk membantu dalam memberikan penilaian yang baik bagi seorang mahasiswa. Misalnya dengan mengetahui penilaian seorang mahasiswa dari dosen lain jadi seorang dosen dapat menyimpulkan sebuah penilaian yang terbaik bagi mahasiswa tersebut. Maka dari itu perangkat lunak yang akan dibangun menjadi sebuah wadah dalam bertukar pendapat mengenai seorang mahasiswa, sehingga setiap dosen dapat memberikan penilaian yang terbaik bagi seorang mahasiswa.
- Semua responden setuju untuk menggunakan akun dosen (@unpar.ac.id) dalam menggunakan sistem yang berhubungan dengan mahasiswa. Hal tersebut menunjukkan dosen terbiasa dengan menggunakan akun dosen untuk memfasilitasi mahasiswa. Maka dari itu pengguna akan menggunakan akun dosen untuk mengakses perangkat lunak yang akan dibangun.
- Sebagian besar responden setuju untuk membuat format penulisan yang mempermudah untuk menulis dan membaca catatan seorang mahasiswa. Selain mempermudah juga hal ini akan berguna agar penulisan di dalam perangkat lunak menjadi seragam kedepannya. Hal tersebut menunjukkan adanya kebutuhan untuk membuat sebuah format penulisan dalam membuat sebuah catatan seorang mahasiswa. Maka dari itu pada perangkat lunak yang akan dibangun,

Tabel 3.6: Tabel Jawaban Pertanyaan Keenam, Menggunakan Akun Dosen

| Responden | Sangat Setuju | Setuju | Netral | Tidak Setuju | Sangat Tidak Setuju |
|-----------|---------------|--------|--------|--------------|---------------------|
| 1         | ✓             |        |        |              |                     |
| 2         |               | ✓      |        |              |                     |
| 3         | ✓             |        |        |              |                     |
| 4         | ✓             |        |        |              |                     |
| 5         |               | ✓      |        |              |                     |
| 6         |               | ✓      |        |              |                     |
| 7         | ✓             |        |        |              |                     |
| 8         |               | ✓      |        |              |                     |
| 9         | ✓             |        |        |              |                     |
| 10        | ✓             |        |        |              |                     |

Tabel 3.7: Tabel Jawaban Pertanyaan Ketujuh, Membuat Sebuah Format

| Responden | Sangat Setuju | Setuju | Netral | Tidak Setuju | Sangat Tidak Setuju |
|-----------|---------------|--------|--------|--------------|---------------------|
| 1         |               | ✓      |        |              |                     |
| 2         |               | ✓      |        |              |                     |
| 3         |               | ✓      |        |              |                     |
| 4         | ✓             |        |        |              |                     |
| 5         |               | ✓      |        |              |                     |
| 6         |               | ✓      |        |              |                     |
| 7         | ✓             |        |        |              |                     |
| 8         |               |        | ✓      |              |                     |
| 9         |               |        | ✓      |              |                     |
| 10        | ✓             |        |        |              |                     |

akan dibuatkan sebuah format penulisan yang mudah ditulis dan mudah dibaca sehingga menghasilkan penulisan yang seragam untuk kedepannya.

- Sebagian responden menggunakan perangkat yang disediakan UNPAR dalam menggunakan sistem yang berhubungan dengan mahasiswa. Sebagian responden yang lain menggunakan perangkat pribadi dalam menggunakan sistem yang berhubungan dengan mahasiswa. Hal tersebut menunjukkan adanya kebutuhan pengguna, agar tampilan antarmuka perangkat lunak dapat mengikuti lebar layar perangkat yang digunakan dosen. Maka dari itu perangkat lunak yang akan dibangun, akan menjamin tampilan antarmuka yang responsif. Agar setiap pengguna tetap merasa nyaman dengan tampilan antarmuka perangkat lunak yang akan dibangun, baik menggunakan perangkat yang disediakan UNPAR maupun menggunakan perangkat pribadi.

## 3.2 Analisis Google OAuth

Pada penelitian ini untuk autentikasi dan otentikasi pada fitur *login* akan menggunakan teknologi Google OAuth 2.0. Penggunaan teknologi ini dikarenakan pengguna akan menggunakan akun dosen yang diberikan oleh UNPAR (@unpar.ac.id) untuk melakukan *login*. Dimana akun yang dimiliki dosen tersebut berada dalam naungan Google. Maka dari itu teknologi ini cocok digunakan untuk mengautentikasi dan mengotorisasi akun dosen. Untuk langkah-langkah penggunaan OAuth 2.0

Tabel 3.8: Tabel Jawaban Pertanyaan Kedelapan, Penggunaan Perangkat Pribadi

| Responden | Sangat Setuju | Setuju | Netral | Tidak Setuju | Sangat Tidak Setuju |
|-----------|---------------|--------|--------|--------------|---------------------|
| 1         |               | ✓      |        |              |                     |
| 2         |               | ✓      |        |              |                     |
| 3         | ✓             |        |        |              |                     |
| 4         |               |        | ✓      |              |                     |
| 5         |               |        | ✓      |              |                     |
| 6         | ✓             |        |        |              |                     |
| 7         |               |        |        | ✓            |                     |
| 8         |               | ✓      |        |              |                     |
| 9         | ✓             |        |        |              |                     |
| 10        |               |        |        | ✓            |                     |

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 harus terpenuhi:

(1) Mendapatkan kepercayaan OAuth 2.0 dari Google Developers Console

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

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 - ufbmstn0pcaq0khubafola133orfgve . 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

Izin dari pihak pengguna digunakan untuk memperoleh token akses. Jadi pada saat pengguna melakukan *login*, pengguna diharuskan *login* menggunakan akun Google miliknya sendiri. Setelah *login* pengguna akan ditanya dan diminta 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 mengirimkan permintaan ke API maka diperlukan ruang lingkup. Berdasarkan sub bab 2.1.4, 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
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.

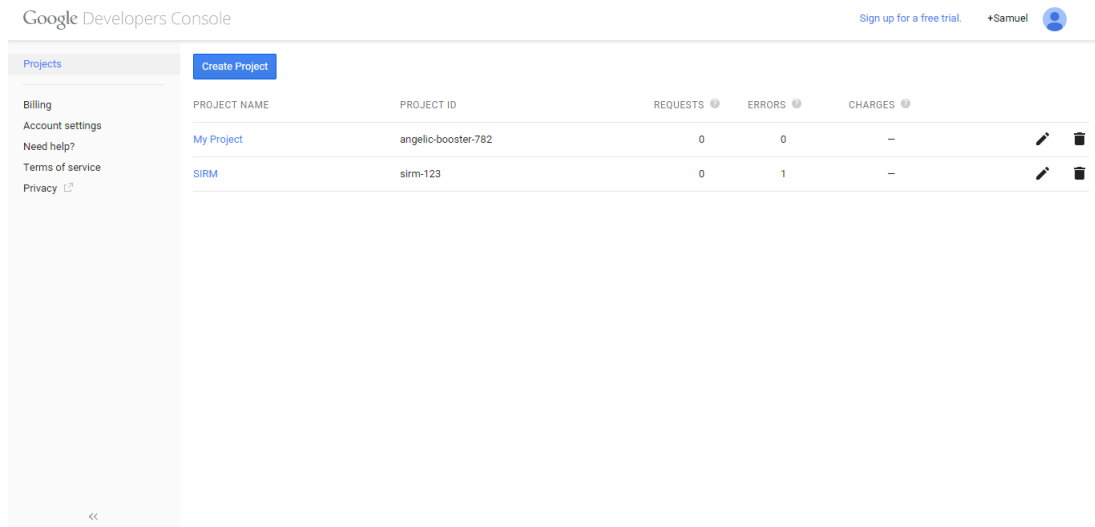
### 3.2.2 Skenario Aplikasi

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 SPBRM 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 SPBRM dapat dilihat pada Gambar 3.9.

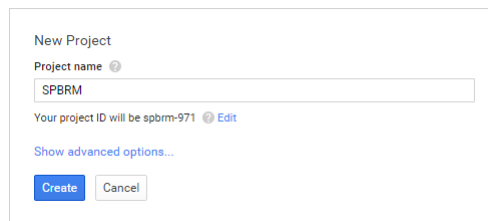
### 3.2.3 Penggunaan API *Client Library* untuk PHP

Pada penelitian ini digunakan API *Client Library* untuk PHP. *Library* terlebih dahulu diunduh dan disimpan sesuai tempat perangkat lunak disimpan. Karena dibuat untuk klien dan menggunakan

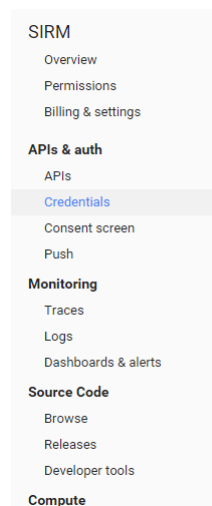




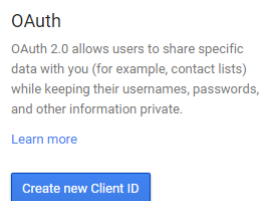
Gambar 3.1: Google Developers Console



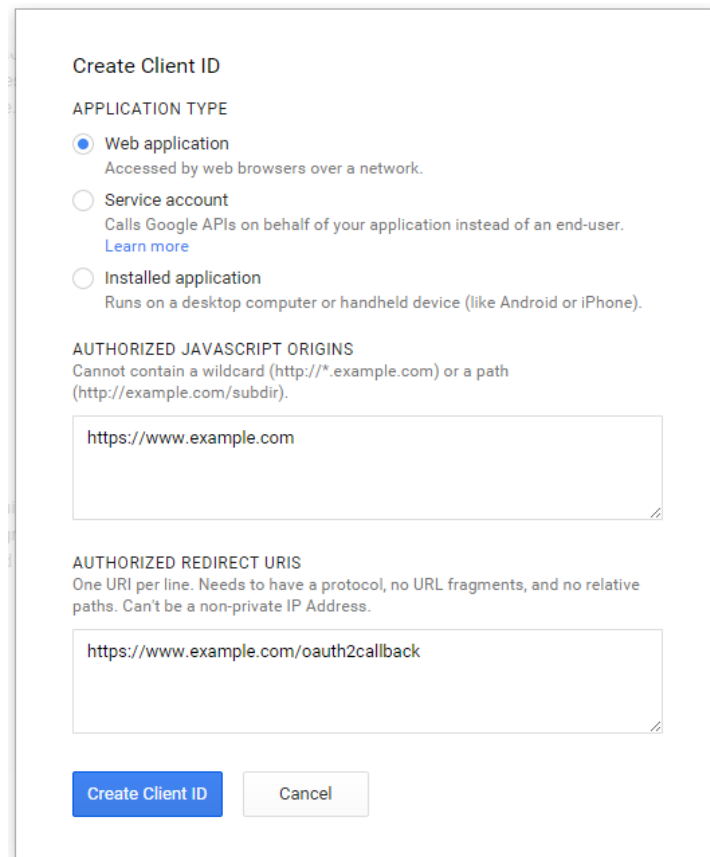
Gambar 3.2: Membuat Proyek Baru



Gambar 3.3: Menu Credentials



Gambar 3.4: Membuat Client ID yang Baru



**Create Client ID**

**APPLICATION TYPE**

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

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

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

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

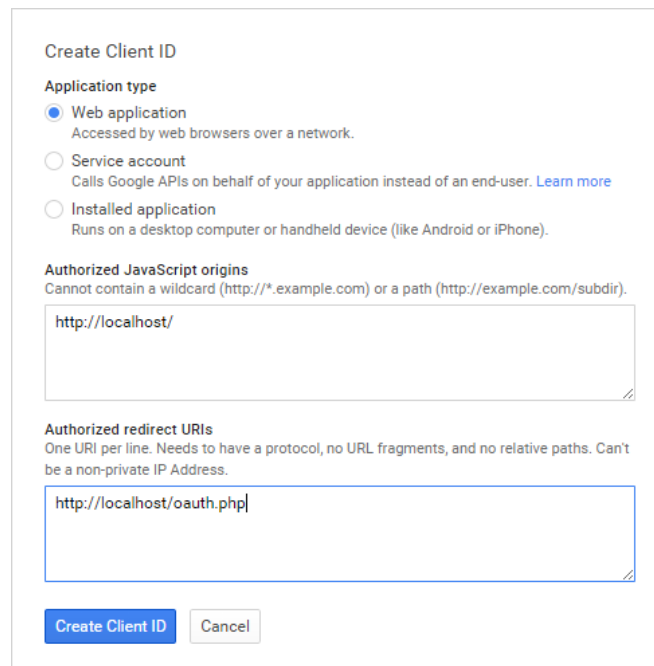
https://www.example.com

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

https://www.example.com/oauth2callback

**Create Client ID** Cancel

Gambar 3.5: Tipe Aplikasi



**Create Client ID**

**Application type**

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

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

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

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

http://localhost/

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

http://localhost/oauth.php

**Create Client ID** Cancel

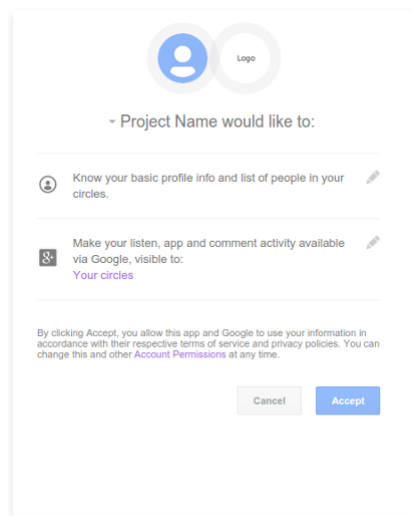
Gambar 3.6: Pengisian Tipe Aplikasi

Client ID for web application

|                    |                                                                             |
|--------------------|-----------------------------------------------------------------------------|
| CLIENT ID          | 568951368854-ufmbistn0pcaq0khubafo1a133orfgve.apps.googleusercontent.com    |
| EMAIL ADDRESS      | 568951368854-ufmbistn0pcaq0khubafo1a133orfgve@developer.gserviceaccount.com |
| CLIENT SECRET      | -cSZ-AUmeQ9PaWWry_IpiBBi                                                    |
| REDIRECT URIS      | http://localhost/pilihmahasiswa.php                                         |
| JAVASCRIPT ORIGINS | http://localhost/                                                           |

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

Gambar 3.7: Client ID



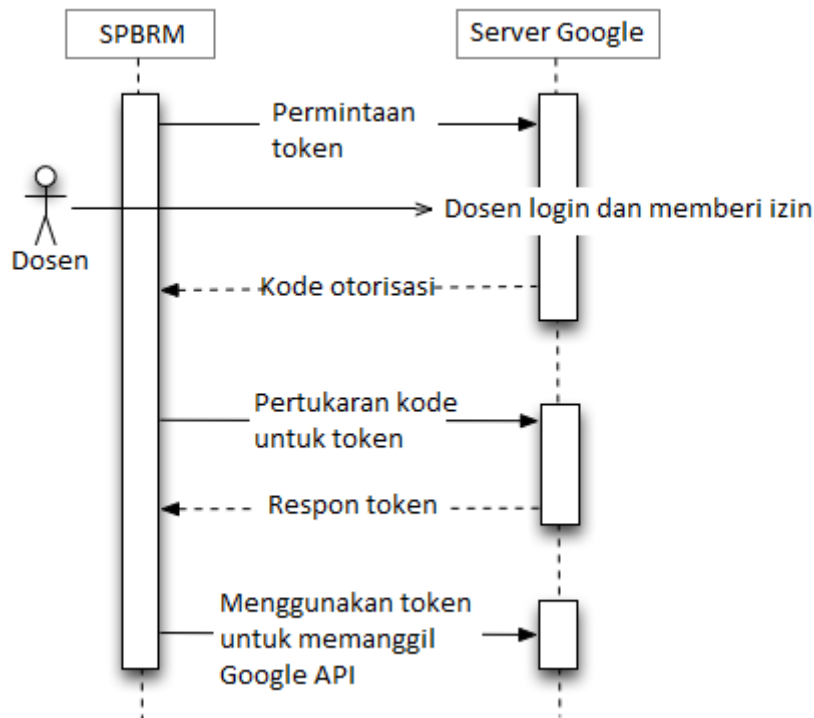
Gambar 3.8: Izin Pihak Pengguna

OAuth maka setiap modul yang menggunakan teknologi tersebut harus memanggil Client.php dan Oauth2.php. Untuk kode dapat dilihat di bawah ini.

```
1 include_once "google-api-php-client-master/src/Google/Client.php";
2 include_once "google-api-php-client-master/src/Google/Service/Oauth2.php";
```

Setelah memanggil Client.php dan Oauth2.php, selanjutnya memanggil klien dan mengisi konfigurasi yang telah dilakukan pada sub sub bab 3.2.1. Konfigurasi yang harus diisi antara lain; id klien, kerahasiaan klien, Redirect URI, Developer Key, dan ruang lingkup. Untuk kode dapat dilihat di bawah ini.

```
1 $client = new Google_Client();
2 $client->setClientId('568951368854-ufmbistn0pcaq0khubafo1a133orfgve.apps.
 googleusercontent.com');
3 $client->setClientSecret('-cSZ-AUmeQ9PaWWry_IpiBBi');
4 $client->setRedirectUri('http://localhost/oauth.php');
5 $client->setDeveloperKey('AIzaSyDRoDJAZUR_TsNUNRUeTYsBb7dFBQKZy7M');
6 $client->setScopes(array('https://www.googleapis.com/auth/plus.login', 'email'));
```



Gambar 3.9: Skenario Aplikasi SPBRM

### 3.3 Analisis Markdown

Sintaks Markdown yang akan digunakan pada perangkat lunak yang akan dibangun. Sesuai dengan landasan teori pada Sub Bab 2.2, Markdown yang akan digunakan adalah GFM. GFM digunakan pada penelitian ini karena memiliki beberapa perbedaan yang signifikan dengan SM dan juga GFM memiliki beberapa fungsi tambahan. Selain itu Markdown cukup mudah untuk dipahami dan digunakan oleh pengguna, karena sintaks Markdown menggunakan teks biasa. Sintaks Markdown akan digunakan pada bagian keterangan mahasiswa agar seragam dalam format penulisannya karena pengguna setuju menggunakan format penulisan yang mempermudah untuk menulis dan membaca. Format penulisan akan disiapkan dengan menggunakan sintaks Markdown untuk pengguna pada bagian membuat entri baru. 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
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
2|
3| 2010730013
4|

```

**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

```

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://bletack.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 pada penelitian ini digunakan untuk menampilkan keterangan mahasiswa yang telah dibuat dengan sintaks Markdown ke halaman HTML. StrapdownJS pada penelitian ini berfungsi sebagai konversi teks yang telah dibuat menggunakan sintaks Markdown menjadi tampilan HTML. Pada penelitian ini strapdown.js terlebih dahulu diunduh, kemudian disimpan pada lokasi yang sama dengan lokasi perangkat lunak dan menggunakan *path* seperti di bawah ini.

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

Skrip tersebut disisipkan pada skrip view.php dan past.php yang berfungsi untuk menampilkan keterangan 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>SPBRM | 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 Edit
17 Lihat Histori
18
19 </div>
20 <hr/>
21 <xmp style="display:none;">
22 ### NPM
23
24 2010730013
25
26 ### Nama
27
28 Samuel
29
30 ### Umum
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://bletaack.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 menjadi responsif. Karena sebagian pengguna menggunakan perangkat yang disediakan UNPAR dan sebagian pengguna menggunakan perangkat sendiri, maka dari itu untuk menjamin tampilan yang responsif untuk pengguna sehingga pada penelitian ini tampilan antarmuka dibuat agar antarmuka sesuai dengan setiap kemungkinan lebar mesin pencari dan/atau layar komputer yang digunakan



pengguna. 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 sehingga terlihat rapih baik lebar mesin pencari dan/atau layar komputer besar maupun kecil. Tombol digunakan untuk merubah tombol yang biasa menjadi lebih enak untuk dilihat. Berikut sintaks penggunaan *Grid* dan Tombol pada list.php. Untuk *grid* pada view.php dan past.php menggunakan kelas *small-11 small-centered columns* agar pada untuk menampilkan keterangan mahasiswa tetap responsif. Dengan menggunakan kelas tersebut, ketika layar mesin pencari dan/atau layar komputer ukuran maksimal, tampilan antarmuka mengikuti ukuran layar tersebut. Namun ketika pengguna mengecilkan lebar layar mesin pencari dan/atau layar komputer, maka antarmuka tetap menyesuaikan dengan lebar layar. Untuk tombol pada penelitian ini menggunakan kelas *button* dan *button secondary*. Pertama, kelas *button* digunakan untuk tombol "Simpan". Kedua, kelas *secondary button* digunakan untuk semua tombol selain tombol "Simpan". Untuk gambar dapat tampilan antarmuka list.php dilihat pada Gambar 3.17 untuk lebar maksimal dan Gambar 3.18 untuk lebar yang dikecilkan.

```

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>SPBRM | 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 Add
18 Logout
19
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("SPBRM", $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> " . $row['npm'] .
57 "</td>";
58 echo "<td> " . $row['nama'] . "</td>";
59 echo "<td> " . $row['pembayaran_terakhir'] . "</td>";
60 echo "</tr>";
61 }
62 echo "</table>";
63
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 list.php Dengan Ukuran Layar Penuh

### 3.6 Analisis Kebutuhan Perangkat Lunak

Pembahasan *use case* diagram, skenario, *entity relationship* diagram, dan data *flow* diagram yang akan digunakan pada penelitian.

Pilih NPM yang ingin dicari / tambah baru.

Add Logout

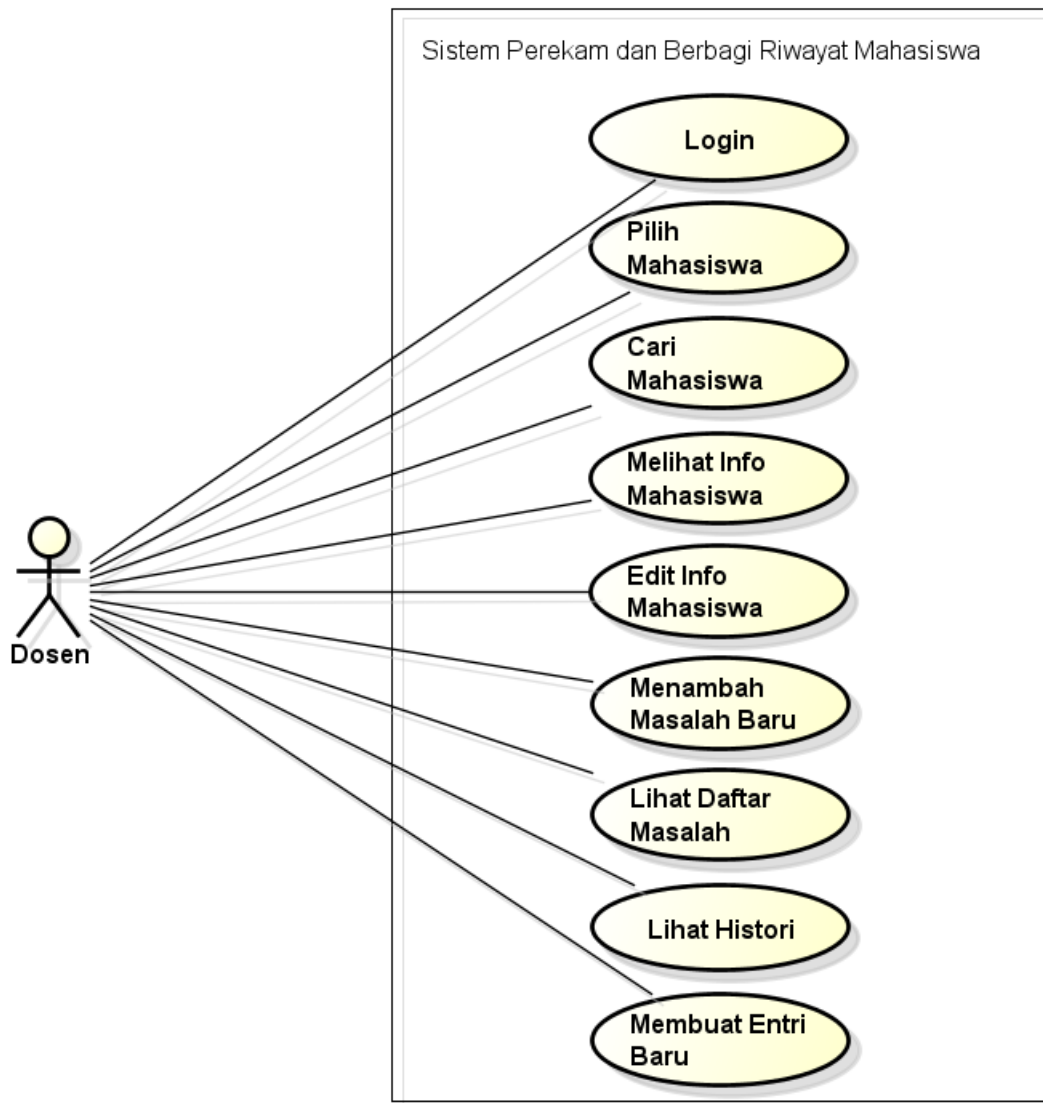
| NPM        | Nama             | Last Update         |
|------------|------------------|---------------------|
| 2010120031 | Kenneth Natanael | 2015-04-30 14:29:51 |
| 2010730001 | Andri Agustian   | 2015-04-29 15:09:33 |
| 2010730005 | Grady Ireneus    | 2015-04-30 14:31:19 |
| 2010730012 | Kevin PL         | 2015-04-24 00:55:57 |
| 2010730013 | Samuel Herman    | 2015-04-24 00:23:34 |
| 2010730039 | henry            | 2015-04-29 18:11:20 |
| 2010730040 | david            | 2015-04-29 18:10:12 |

Gambar 3.18: Tampilan list.php Dengan Ukuran Layar Diperkecil

### 3.6.1 Use Case Diagram

*Use case* diagram merupakan pemodelan yang menunjukkan kegiatan apa saja yang dapat dilakukan pengguna dan kegiatan yang dilakukan sistem. Berikut adalah deskripsi dari *use case* pada Gambar 3.19.

- *Login*  
*Use case* ini memungkinkan pengguna untuk *login* menggunakan akun dosen dan dibantu untuk autentikasi dan otorisasi menggunakan Google OAuth.
- Pilih Mahasiswa  
*Use case* ini memungkinkan pengguna untuk memilih mahasiswa yang ingin dilihat informasinya. Selain itu pengguna juga bisa menemukan tombol "*Add*" untuk membuat entri baru.
- Cari Mahasiswa  
*Use case* ini memungkinkan pengguna untuk mencari mahasiswa yang ingin dicari untuk mendapatkan informasi mahasiswa tersebut.
- Melihat Info Mahasiswa  
*Use case* ini memungkinkan pengguna untuk melihat informasi mahasiswa. Selain itu pengguna bisa menekan tombol "*Edit*" untuk mengedit informasi mahasiswa yang diketahuinya dan pengguna juga bisa menekan tombol "Lihat Histori" untuk melihat histori.
- *Edit* Info Mahasiswa  
*Use case* ini memungkinkan pengguna untuk mengubah informasi mahasiswa yang sebelumnya sedang dilihat oleh pengguna.
- Menambah Masalah Baru  
*Use case* ini memungkinkan pengguna untuk menambah catatan masalah baru seorang mahasiswa yang sebelumnya sedang dilihat oleh pengguna.



Gambar 3.19: *Use Case* Diagram Sistem Perekam dan Berbagi Riwayat Mahasiswa

- Lihat Daftar Masalah

*Use case* ini memungkinkan pengguna untuk melihat daftar masalah yang dimiliki oleh seorang mahasiswa.

- Lihat Histori

*Use case* ini memungkinkan pengguna untuk melihat histori untuk setiap perubahan terhadap data seorang mahasiswa dan aksi yang dilakukan pengguna baik melihat, mengedit, maupun membuat entri. Selain itu pengguna juga dapat melihat keterangan dari versi awal (pertama kali dibuat) hingga versi saat ini.

- Membuat Entri Baru

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

### 3.6.2 Skenario

Skenario untuk menjelaskan *use case* yang terpadat pada Gambar 3.19 dapat dilihat pada daftar di bawah ini.

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

Tabel 3.9: Skenario *Login*

|                |                                                                                                                               |                                                   |                                                                                             |
|----------------|-------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|---------------------------------------------------------------------------------------------|
| Nama           | <i>Login</i>                                                                                                                  |                                                   |                                                                                             |
| Aktor          | Pengguna                                                                                                                      |                                                   |                                                                                             |
| Deskripsi      | Melakukan <i>login</i> dengan menggunakan akun dosen, kemudian diautentikasi dan diotorisasi dengan menggunakan Google OAuth. |                                                   |                                                                                             |
| Kondisi Awal   | Masih berada pada index.php                                                                                                   |                                                   |                                                                                             |
| Kondisi Akhir  | Sudah berada pada list.php                                                                                                    |                                                   |                                                                                             |
| Skenario Utama | No                                                                                                                            | Aksi Aktor                                        | Reaksi Sistem                                                                               |
|                | 1                                                                                                                             | Pengguna melakukan <i>login</i> dengan akun dosen | Server akan mengirimkan pertanyaan untuk izin dari pihak pengguna                           |
|                | 2                                                                                                                             | Pengguna memberikan izin                          | Aplikasi mendapatkan otorisasi kode sebagai tanda pengguna diperbolehkan mengakses aplikasi |
| 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* Cari Mahasiswa, skenarionya dapat dilihat pada Tabel 3.11.
- Untuk *use case* Melihat Info Mahasiswa, skenarionya dapat dilihat pada Tabel 3.12.
- Untuk *use case* Edit Mahasiswa, skenarionya dapat dilihat pada Tabel 3.13.
- Untuk *use case* Menambah Masalah Baru, skenarionya dapat dilihat pada Tabel 3.14.
- Untuk *use case* Lihat Daftar Masalah, skenarionya dapat dilihat pada Tabel 3.15.
- Untuk *use case* Lihat Histori, skenarionya dapat dilihat pada Tabel 3.16.
- Untuk *use case* Membuat Entri Baru, skenarionya dapat dilihat pada Tabel 3.17.

### 3.6.3 Entity Relationship Diagram (ERD)

Entity Relationship Diagram Sistem Perekam dan Berbagi Riwayat Mahasiswa dapat dilihat pada Gambar 3.20.

Tabel 3.10: Skenario Pilih Mahasiswa

|                |                                                                                                 |                                              |                                                                       |
|----------------|-------------------------------------------------------------------------------------------------|----------------------------------------------|-----------------------------------------------------------------------|
| Nama           | Pilih Mahasiswa                                                                                 |                                              |                                                                       |
| Aktor          | Pengguna                                                                                        |                                              |                                                                       |
| Deskripsi      | Pengguna dapat memilih mahasiswa berdasarkan NPM                                                |                                              |                                                                       |
| Kondisi Awal   | Sebuah form dengan tabel yang berisi data mahasiswa (npm, nama, dan tanggal perubahan terakhir) |                                              |                                                                       |
| Kondisi Akhir  | Salah satu mahasiswa terpilih                                                                   |                                              |                                                                       |
| Skenario Utama | No                                                                                              | Aksi Aktor                                   | Reaksi Sistem                                                         |
|                | 1                                                                                               | Pengguna memilih mahasiswa berdasarkan NPM   | Sistem menampilkan daftar mahasiswa yang terekam pada <i>database</i> |
|                | 2                                                                                               | Pengguna mengklik NPM mahasiswa yang dipilih | Pindah ke halaman view.php sesuai dengan npm yang diklik pengguna     |
| Eksepsi        | -                                                                                               |                                              |                                                                       |

Pada ERD Sistem Perekam dan Berbagi Riwayat Mahasiswa terdapat tiga entitas yaitu info mahasiswa, masalah dan histori. Entitas info mahasiswa dengan relasi memiliki terbuhung dengan entitas histori. Entitas info mahasiswa dengan relasi memiliki terbuhung dengan entitas masalah. Entitas info mahasiswa untuk menampung data npm, nama, keterangan terakhir, dan waktu keterangan tersebut dibuat atau dirubah. Entitas masalah untuk menampung data masalah dan yang bersangkutan dengan masalah tersebut. Entitas histori untuk menampung data email pengguna, status (membuat entri, melihat, atau mengedit), keterangan yang disimpan agar dapat dilihat versi dari pertama kali dibuat hingga versi saat ini, dan tanggal dari keterangan tersebut.

### 3.6.4 Data Flow Diagram

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

#### (1) Data Context Diagram

Data Context Diagram Sistem Perekam dan Berbagi Riwayat Mahasiswa dapat dilihat pada Gambar 3.21.

#### (2) Data Flow Diagram Level 1

Data Flow Diagram Level 1 Sistem Perekam dan Berbagi Riwayat Mahasiswa dapat dilihat pada Gambar 3.22.

Untuk spesifikasi proses dapat dilihat di bawah ini.

- No dan Nama Proses : P1 *Login*

Deskripsi : Proses akses ke perangkat lunak dengan menggunakan akun dosen (@unpar.ac.id).

Data Input : Email, password

Tabel 3.11: Skenario Cari Mahasiswa

|                |                                                         |                                              |                                                                       |
|----------------|---------------------------------------------------------|----------------------------------------------|-----------------------------------------------------------------------|
| Nama           | Cari Mahasiswa                                          |                                              |                                                                       |
| Aktor          | Pengguna                                                |                                              |                                                                       |
| Deskripsi      | Pengguna dapat mencari mahasiswa berdasarkan NPM        |                                              |                                                                       |
| Kondisi Awal   | Sebuah form dengan tabel yang berisi data mahasiswa     |                                              |                                                                       |
| Kondisi Akhir  | Daftar mahasiswa terpilih sesuai dengan npm yang dicari |                                              |                                                                       |
| Skenario Utama | No                                                      | Aksi Aktor                                   | Reaksi Sistem                                                         |
|                | 1                                                       | Pengguna mencari mahasiswa berdasarkan NPM   | Sistem melakukan seleksi daftar mahasiswa berdasarkan NPM yang dicari |
|                | 2                                                       | Pengguna mengklik NPM mahasiswa yang dipilih | Pindah ke halaman view.php sesuai dengan npm yang diklik pengguna     |
| Eksepsi        | -                                                       |                                              |                                                                       |

Data Output : *login* valid

Proses : Mengecek email yang digunakan oleh pengguna diakhiri @unpar.ac.id atau tidak. Jika ya maka sistem akan mengarahkan ke halaman dimana pengguna harus memberi izin untuk mengakses sistem. Setelah pengguna memberi izin 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 : NPM yang terdapat pada tabel daftar mahasiswa di halaman utama

Data Output : Data mahasiswa berupa tabel daftar mahasiswa

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

- No dan Nama Proses : P3 Cari Mahasiswa

Deskripsi : Proses mencari mahasiswa

Data Input : NPM pada teks area untuk mencari mahasiswa

Data Output : Data mahasiswa berupa tabel daftar mahasiswa sesuai npm yang dicari

Proses : pengguna mencari mahasiswa yang ingin dicari berdasarkan npm dan memilih mahasiswa tersebut dari tabel hasil pencarian.

- No dan Nama Proses : P4 Melihat Info Mahasiswa

Deskripsi : Proses melihat informasi mahasiswa

Data Input : NPM, nama, deskripsi umum, catatan

Data Output : Data informasi mahasiswa, data histori

Proses : pengguna mendapatkan info mahasiswa dari mahasiswa yang telah pengguna pilih pada proses 2 atau cari pada proses 3. Proses ini dicatat pada histori sebagai aksi melihat.

- No dan Nama Proses : P5 Edit Info Mahasiswa

Deskripsi : Proses mengedit informasi mahasiswa

Tabel 3.12: Skenario Melihat Info Mahasiswa

|                |                                                                                                                                                                                                                                                                                                                                                    |                                 |                                   |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|-----------------------------------|
| Nama           | Melihat Info Mahasiswa                                                                                                                                                                                                                                                                                                                             |                                 |                                   |
| Aktor          | Pengguna                                                                                                                                                                                                                                                                                                                                           |                                 |                                   |
| Deskripsi      | Melihat info mahasiswa yang telah dipilih pada list.php                                                                                                                                                                                                                                                                                            |                                 |                                   |
| Kondisi Awal   | Menampilkan info yang dimiliki mahasiswa                                                                                                                                                                                                                                                                                                           |                                 |                                   |
| Kondisi Akhir  | Jika pengguna mengklik " <i>Edit</i> " maka pindah ke edit.php, jika pengguna mengklik "Tambah Masalah" maka pindah ke newproblem.php, jika pengguna mengklik "Lihat Masalah" maka pindah ke problem.php, jika pengguna mengklik "Lihat Histori" maka pindah ke history.php, jika pengguna mengklik "Logout" maka pengguna akan keluar dari sistem |                                 |                                   |
| Skenario Utama | No                                                                                                                                                                                                                                                                                                                                                 | Aksi Aktor                      | Reaksi Sistem                     |
|                | 1                                                                                                                                                                                                                                                                                                                                                  | Pengguna melihat info mahasiswa | Sistem menampilkan info mahasiswa |
| Eksepsi        | -                                                                                                                                                                                                                                                                                                                                                  |                                 |                                   |

Data Input : Deskripsi umum, catatan

Data Output : Deskripsi umum, catatan, data histori

Proses : pengguna memperbaharui informasi yang dimiliki mahasiswa yang telah pengguna pilih pada proses 2 atau cari pada proses 3. Proses ini dicatat pada histori sebagai aksi mengedit.

- No dan Nama Proses : P6 Menambah Masalah Baru

Deskripsi : Proses menambah catatan masalah baru

Data Input : Masalah dengan siapa, masalah

Data Output : Masalah dengan siapa, masalah, data histori

Proses : pengguna menambah catatan masalah baru mahasiswa yang telah pengguna pilih pada proses 2 atau cari pada proses 3. Proses ini dicatat pada histori sebagai aksi menambah catatan masalah.

- No dan Nama Proses : P7 Lihat Daftar Masalah

Deskripsi : Proses melihat daftar masalah

Data Input : Data masalah

Data Output : Data masalah

Proses : pengguna mendapatkan daftar masalah dari mahasiswa yang telah pengguna pilih pada proses 2 atau cari pada proses 3.

- No dan Nama Proses : P8 Lihat Histori

Deskripsi : Proses melihat histori

Data Input : Data histori

Data Output : Data histori

Proses : pengguna mendapatkan histori dari mahasiswa yang telah pengguna pilih pada proses 2 atau cari pada proses 3. Pengguna juga dapat melihat informasi mahasiswa versi sebelumnya.



Tabel 3.13: Skenario Edit Mahasiswa

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

Tabel 3.14: Skenario Menambah Masalah Baru

|                |                                                                        |                                 |                                                           |
|----------------|------------------------------------------------------------------------|---------------------------------|-----------------------------------------------------------|
| Nama           | Menambah Masalah Baru                                                  |                                 |                                                           |
| Aktor          | Pengguna                                                               |                                 |                                                           |
| Deskripsi      | Membuat catatan masalah baru seorang mahasiswa                         |                                 |                                                           |
| Kondisi Awal   | Menampilkan form untuk memasukkan data masalah yang dimiliki mahasiswa |                                 |                                                           |
| Kondisi Akhir  | Form dengan data yang telah diisi akan direkam                         |                                 |                                                           |
| Skenario Utama | No                                                                     | Aksi Aktor                      | Reaksi Sistem                                             |
|                | 1                                                                      | Pengguna mengisi data masalah   | Sistem menampilkan form masalah                           |
|                | 2                                                                      | Pengguna menyimpan data masalah | Sistem akan merekam data masalah ke dalam <i>database</i> |
| Eksepsi        | -                                                                      |                                 |                                                           |

- No dan Nama Proses : P9 Membuat Entri Baru

Deskripsi : Proses membuat entri baru

Data Input : NPM, nama, deskripsi umum, catatan

Data Output : NPM, nama, deskripsi umum, catatan, data histori

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

Tabel 3.15: Skenario Lihat Daftar Masalah

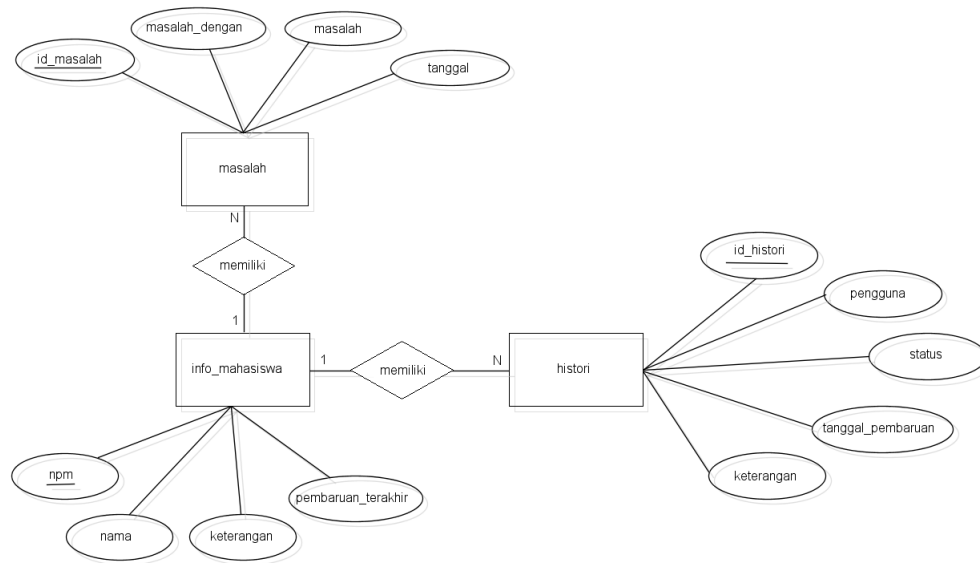
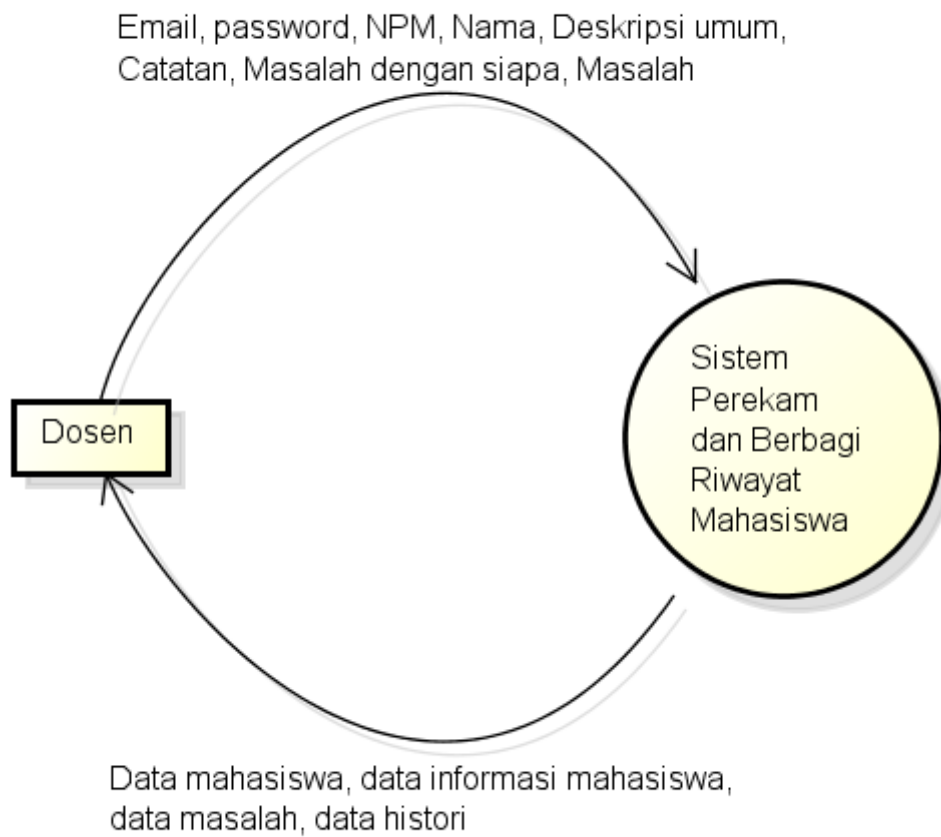
|                |                                                                                     |                                 |                                        |
|----------------|-------------------------------------------------------------------------------------|---------------------------------|----------------------------------------|
| Nama           | Lihat Daftar Masalah                                                                |                                 |                                        |
| Aktor          | Pengguna                                                                            |                                 |                                        |
| Deskripsi      | Melihat daftar masalah yang dimiliki seorang mahasiswa                              |                                 |                                        |
| Kondisi Awal   | Menampilkan daftar masalah seorang mahasiswa                                        |                                 |                                        |
| Kondisi Akhir  | Terus bertambah sesuai catatan masalah yang ditambah                                |                                 |                                        |
| Skenario Utama | No                                                                                  | Aksi Aktor                      | Reaksi Sistem                          |
|                | 1                                                                                   | Pengguna melihat daftar masalah | Sistem akan menampilkan daftar masalah |
| Eksepsi        | Mahasiswa yang tidak memiliki masalah akan diberi keterangan tidak memiliki masalah |                                 |                                        |

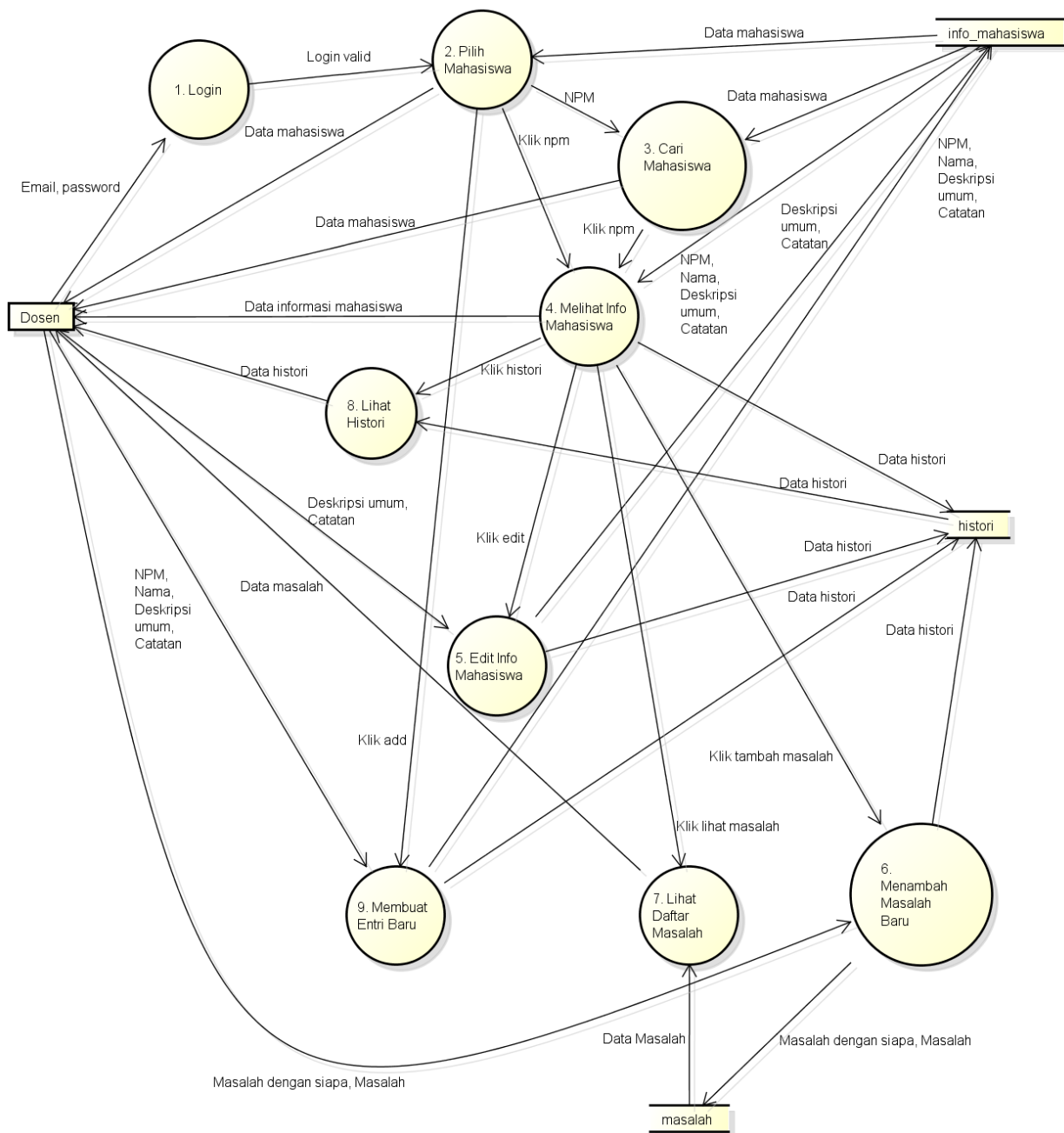
Tabel 3.16: Skenario Lihat Histori

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

Tabel 3.17: Skenario Membuat Entri Baru

|                |                                                                                                                             |                                                  |                                                                |
|----------------|-----------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|----------------------------------------------------------------|
| Nama           | Membuat Entri Baru                                                                                                          |                                                  |                                                                |
| Aktor          | Pengguna                                                                                                                    |                                                  |                                                                |
| Deskripsi      | Membuat entri baru yang belum ada pada <i>database</i>                                                                      |                                                  |                                                                |
| Kondisi Awal   | Menampilkan form untuk menambah entri baru                                                                                  |                                                  |                                                                |
| Kondisi Akhir  | <i>Input</i> pada form akan dimasukkan kedalam <i>database</i>                                                              |                                                  |                                                                |
| Skenario Utama | No                                                                                                                          | Aksi Aktor                                       | Reaksi Sistem                                                  |
|                | 1                                                                                                                           | Pengguna mengisi form entri baru                 | Sistem menampilkan form entri baru                             |
|                | 2                                                                                                                           | Pengguna menyimpan masukkan dari form entri baru | Sistem akan merekam masukkan pengguna ke dalam <i>database</i> |
| Eksepsi        | Jika data yang dimasukkan sudah ada di <i>database</i> , maka sistem akan menampilkan keterangan bahwa data telah terdaftar |                                                  |                                                                |

Gambar 3.20: *Entity Relationship Diagram*Gambar 3.21: Data *Context* Diagram Sistem Perekam dan Berbagi Riwayat Mahasiswa



Gambar 3.22: Data Flow Diagram Sistem Perekam dan Berbagi Riwayat Mahasiswa

## 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 4.1.



Gambar 4.1: Desain Antarmuka Halaman Awal

Keterangan :

- (1) Bagian ini merupakan judul halaman awal perangkat lunak dan terdapat keterangan dari SPBRM.
- (2) Bagian ini merupakan *link* berupa teks yang dapat diklik untuk melakukan login dengan menggunakan akun Google.

#### 4.1.2 Tampilan Web Pilih Mahasiswa

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

1. **Masukkan dan pilih NPM yang ingin dicari / tambah baru.**

2.

3.

4. 

| NPM        | Nama     | Last Update         |
|------------|----------|---------------------|
| 2010730001 | Andri AG | 2015-02-23 21:32:43 |
| 2010730012 | Kevin PL | 2015-03-24 22:33:44 |
| 2010730013 | Samuel H | 2015-04-25 23:34:45 |

Gambar 4.2: Desain Antarmuka Pilih Mahasiswa

Keterangan :

- (1) Bagian ini merupakan judul dari halaman untuk memilih mahasiswa.
- (2) Bagian ini merupakan area untuk memasukan npm yang ingin dicari oleh pengguna.
- (3) Bagian ini merupakan tombol untuk melakukan aksi cari npm, add maupun logout.
- (4) Bagian ini merupakan tempat menampilkan data mahasiswa dalam bentuk tabel. NPM dapat diklik oleh pengguna untuk memilih mahasiswa.

#### 4.1.3 Tampilan Web Cari Mahasiswa

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

1. **Masukkan dan pilih NPM yang ingin dicari / tambah baru.**

2.

3.

4. 

| NPM        | Nama     | Last Update         |
|------------|----------|---------------------|
| 2010730013 | Samuel H | 2015-04-25 23:34:45 |

Gambar 4.3: Desain Antarmuka Cari Mahasiswa

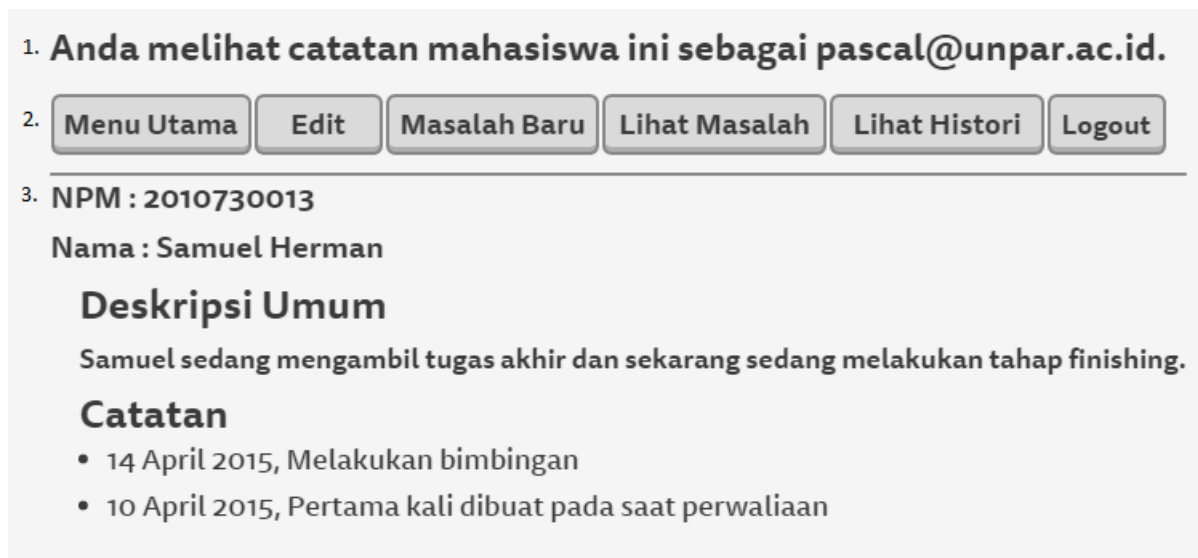
Keterangan :

- (1) Bagian ini merupakan judul dari halaman untuk mencari mahasiswa.
- (2) Bagian ini merupakan area untuk memasukan npm yang ingin dicari oleh pengguna.

- (3) Bagian ini merupakan tombol untuk melakukan aksi add atau logout.
- (4) Bagian ini merupakan tempat menampilkan data mahasiswa yang dicari oleh pengguna dalam bentuk tabel. NPM dapat diklik oleh pengguna untuk memilih mahasiswa.

#### 4.1.4 Tampilan *Web* Info Mahasiswa

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



Gambar 4.4: Desain Antarmuka Info Mahasiswa

Keterangan :

- (1) Bagian ini merupakan judul dari halaman info mahasiswa yang menampilkan keterangan pengguna yang sedang mengakses atau melihat info mahasiswa tersebut.
- (2) Bagian ini merupakan tombol untuk melakukan aksi pindah ke menu utama, mengedit, membuat catatan masalah baru, lihat daftar masalah, lihat histori dan logout.
- (3) Bagian ini merupakan tempat menampilkan info mahasiswa berupa npm, nama, deskripsi umum dan catatan yang berasal dari *database*.

#### 4.1.5 Tampilan *Web* Edit Info Mahasiswa

Perancangan tampilan *web* untuk edit info mahasiswa dapat dilihat pada Gambar 4.5.

Keterangan :

- (1) Bagian ini merupakan judul dari halaman edit info mahasiswa yang menampilkan keterangan pengguna yang sedang mengedit info mahasiswa tersebut.
- (2) Bagian ini merupakan teks yang menampilkan NPM dan nama mahasiswa yang telah dipilih oleh pengguna untuk diedit infonya.
- (3) Bagian ini merupakan tombol untuk melakukan aksi kembali ke info mahasiswa, simpan untuk perubahan yang telah dilakukan, pindah ke menu utama, dan logout.

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

2. **NPM 2010730013 Nama Samuel Herman**

3.

---

4. **Deskripsi Umum**

5.

6. **Catatan**

7.

Gambar 4.5: Desain Antarmuka Edit Info Mahasiswa

- (4) Bagian ini merupakan teks yang merupakan bagian untuk mengedit deskripsi umum.
- (5) Bagian ini merupakan tempat yang menampilkan deskripsi umum seorang mahasiswa yang telah dipilih oleh pengguna. Deskripsi umum yang ditampilkan pada bagian ini berasal dari *database* dan pengguna dapat melakukan perubahan pada deskripsi umum di bagian ini.
- (6) Bagian ini merupakan teks yang merupakan bagian untuk mengedit catatan.
- (7) Bagian ini merupakan tempat yang menampilkan catatan seorang mahasiswa yang telah dipilih oleh pengguna. Catatan yang ditampilkan pada bagian ini berasal dari *database* dan pengguna dapat melakukan perubahan pada catatan di bagian ini.

#### 4.1.6 Tampilan Web Tambah Masalah Baru

Perancangan tampilan *web* untuk info mahasiswa dapat dilihat pada Gambar 4.6.

Keterangan :

- (1) Bagian ini merupakan judul dari halaman tambah masalah baru yang menampilkan keterangan pengguna yang sedang menambah catatan masalah baru.
- (2) Bagian ini merupakan teks yang menampilkan NPM dan nama mahasiswa yang telah dipilih oleh pengguna untuk ditambah catatan masalah.
- (3) Bagian ini merupakan tombol untuk melakukan aksi kembali ke info mahasiswa, edit, pindah ke menu utama, dan logout.
- (4) Bagian ini merupakan teks yang merupakan bagian masalah bersangkutan dengan siapa / mata kuliah apa.



1. Anda menambah catatan masalah mahasiswa ini sebagai **pascal@unpar.ac.id**.

2. **NPM 2010730013 Nama Samuel Herman**

3. **Kembali** **Simpan** **Menu Utama** **Logout**

---

4. **Masalah dengan siapa / mata kuliah apa**

5.

6. **Masalah**

7.

Gambar 4.6: Desain Antarmuka Tambah Masalah Baru

- (5) Bagian ini merupakan tempat untuk memasukkan pihak yang bersangkutan oleh pengguna. Pihak yang bersangkutan dapat diisi dengan nama dosen atau nama mata kuliah.
- (6) Bagian ini merupakan teks yang merupakan bagian masalahnya.
- (7) Bagian ini merupakan tempat untuk memasukkan penjelasan atau rincian masalah yang dimiliki mahasiswa tersebut oleh pengguna.

#### 4.1.7 Tampilan *Web* Lihat Daftar Masalah

Perancangan tampilan *web* untuk info mahasiswa dapat dilihat pada Gambar 4.7.

1. **Daftar masalah yang dimiliki NPM 2010730013 Nama Samuel Herman**

2. **Kembali** **Menu Utama** **Logout**

---

3. 

- 2015-05-27 11:22:13 Memiliki masalah dengan / pada kelas Skripsi, masalahnya adalah menunda sidang dan mengambil topik baru
- 2015-04-30 15:14:13 Memiliki masalah dengan / pada kelas DAA, masalahnya adalah sering terlambat mengumpulkan tugas

Gambar 4.7: Desain Antarmuka Lihat Daftar Masalah

Keterangan :

- (1) Bagian ini merupakan judul dari halaman lihat daftar masalah yang menampilkan keterangan npm dan nama dari mahasiswa yang dipilih oleh pengguna.
- (2) Bagian ini merupakan tombol untuk melakukan aksi kembali ke info mahasiswa, pindah ke menu utama, dan logout.

- (3) Bagian ini merupakan tempat menampilkan daftar masalah dari mahasiswa yang dipilih oleh pengguna. Daftar masalah tersebut berasal dari *database*.

#### 4.1.8 Tampilan Web Lihat Histori

Perancangan tampilan *web* untuk lihat histori dapat dilihat pada Gambar 4.8.



Gambar 4.8: Desain Antarmuka Lihat Histori

Keterangan :

- (1) Bagian ini merupakan teks yang menampilkan keterangan NPM dan nama mahasiswa yang telah dipilih untuk dilihat historinya.
- (2) Bagian ini merupakan tombol untuk melakukan aksi kembali ke info mahasiswa, pindah ke menu utama, dan logout.
- (3) Bagian ini merupakan daftar histori dari mahasiswa yang telah dipilih. Daftar histori tersebut berasal dari *database*.
- (4) Bagian ini merupakan *link* yang dapat digunakan pengguna untuk mengakses keterangan versi dari awal sampai yang terakhir.

#### 4.1.9 Tampilan Web Lihat Versi Ini

Perancangan tampilan *web* untuk lihat versi ini dapat dilihat pada Gambar 4.9.

Keterangan :

- (1) Bagian ini merupakan tombol untuk melakukan aksi kembali ke lihat histori dan logout.
- (2) Bagian ini tempat menampilkan info dari mahasiswa yang telah dipilih. Info yang ditampilkan adalah deskripsi umum dan catatan. Info mahasiswa tersebut berasal dari *database*.

#### 4.1.10 Tampilan Web Entri Baru

Perancangan tampilan *web* untuk entri baru dapat dilihat pada Gambar 4.10.

Keterangan :

1. **Kembali** **Logout**

---

2. **Deskripsi Umum**  
Samuel sedang mengambil tugas akhir dan sekarang sedang melakukan tahap finishing.

**Catatan**

- 14 April 2015, Melakukan bimbingan
- 10 April 2015, Pertama kali dibuat pada saat perwalian

Gambar 4.9: Desain Antarmuka Lihat Versi Ini

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

2. **Kembali** **Simpan** **Menu Utama** **Logout**

---

3. **NPM**

4. **Nama**

5. **Deskripsi Umum**

6. **Catatan**

Gambar 4.10: Desain Antarmuka Entri Baru

- (1) Bagian ini merupakan judul dari halaman entri baru yang menampilkan keterangan pengguna yang sedang menambah entri baru.
- (2) Bagian ini merupakan tombol untuk melakukan aksi kembali ke pilih mahasiswa, simpan, pindah ke menu utama, dan logout.
- (3) Bagian ini merupakan teks dan tempat untuk memasukkan npm seorang mahasiswa.
- (4) Bagian ini merupakan teks dan tempat untuk memasukkan nama seorang mahasiswa.
- (5) Bagian ini merupakan teks dan tempat untuk memasukkan deskripsi umum seorang mahasiswa.
- (6) Bagian ini merupakan teks dan tempat untuk memasukkan catatan seorang mahasiswa.

## 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> dari akun dosen (@unpar.ac.id) kemudian sistem akan melakukan autentikasi dan otorisasi menggunakan Google Oauth. |

### 4.2.2 Modul Pilih Mahasiswa

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

Tabel 4.2: Modul Pilih Mahasiswa

|                    |                                                                                                                                                                                                                                                                             |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nama Modul         | list.php                                                                                                                                                                                                                                                                    |
| Input              | npm                                                                                                                                                                                                                                                                         |
| Output             | Tabel mahasiswa                                                                                                                                                                                                                                                             |
| Tabel yang diakses | InfoMahasiswa                                                                                                                                                                                                                                                               |
| Deskripsi          | Pengguna memilih npm yang terdapat pada tabel untuk mahasiswa yang ingin dicari, kemudian dengan menekan npm yang terdapat pada tabel sebagai masukkan yang akan diteruskan ke modul info mahasiswa dan pengguna juga dapat membuat entri baru dengan menekan tombol "Add". |

### 4.2.3 Modul Cari Mahasiswa

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

### 4.2.4 Modul Info Mahasiswa

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

### 4.2.5 Modul Edit Mahasiswa

Modul *edit* mahasiswa yang dilakukan oleh pengguna (dosen) dapat dilihat pada Tabel 4.5.

Tabel 4.3: Modul Cari Mahasiswa

|                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nama Modul         | search.php                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Input              | npm                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Output             | Tabel mahasiswa                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Tabel yang diakses | InfoMahasiswa                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Deskripsi          | Pengguna mencari npm yang ingin dicari dengan memasukkan npm pada tempat yang disediakan untuk memasukkan npm. Kemudian pengguna menekan tombol "Cari NPM" dan pengguna akan mendapatkan daftar mahasiswa sesuai dengan npm yang dimasukkan sebelumnya. Pengguna dapat menekan npm yang terdapat pada tabel sebagai masukkan yang akan diteruskan ke modul info mahasiswa dan pengguna juga dapat membaut entri baru dengan menekan tombol "Add". |

Tabel 4.4: 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 melakukan enam aksi pada modul ini. Pertama, kembali ke halaman utama dengan menekan tombol "Menu Utama". Kedua, dapat merubah info mahasiswa yang ada dengan menekan tombol "Edit". Ketiga, dapat menambah catatan masalah baru dengan menekan tombol "Masalah Baru". Keempat, dapat melihat daftar masalah yang dimiliki mahasiswa tersebut dengan menekan tombol "Lihat Masalah". Kelima, dapat melihat histori mahasiswa tersebut dengan menekan tombol "Lihat Histori". Keenam, dapat keluar dari sistem dengan menekan tombol "Logout". |

#### 4.2.6 Modul Menambah Masalah Baru

Modul menambah masalah baru yang dilakukan oleh pengguna (dosen) dapat dilihat pada Tabel [4.6](#).

#### 4.2.7 Modul Lihat Daftar Masalah

Modul lihat daftar masalah yang dilakukan oleh pengguna (dosen) dapat dilihat pada Tabel [4.7](#).

#### 4.2.8 Modul Lihat Histori

Modul lihat histori yang dilakukan oleh pengguna (dosen) dapat dilihat pada Tabel [4.8](#).

#### 4.2.9 Modul Entri Baru

Modul entri baru yang dilakukan oleh pengguna (dosen) dapat dilihat pada Tabel [4.9](#).

Tabel 4.5: Modul *Edit* Mahasiswa

|                    |                                                                                                                                                                                                                                                                               |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nama Modul         | edit.php                                                                                                                                                                                                                                                                      |
| Input              | teks                                                                                                                                                                                                                                                                          |
| Output             | -                                                                                                                                                                                                                                                                             |
| Tabel yang diakses | InfoMahasiswa dan Histori                                                                                                                                                                                                                                                     |
| Deskripsi          | Pengguna memasukkan atau merubah keterangan mahasiswa pada teks area lalu pengguna menyimpan dengan menekan tombol "Simpan" untuk menaruh perubahan yang dilakukan. Pengguna dapat kembali ke modul info mahasiswa tanpa melakukan perubahan dengan menekan tombol "Kembali". |

Tabel 4.6: Modul Menambah Masalah Baru

|                    |                                                                                                                                                                                                                                                                                                                      |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nama Modul         | newproblem.php                                                                                                                                                                                                                                                                                                       |
| Input              | teks                                                                                                                                                                                                                                                                                                                 |
| Output             | -                                                                                                                                                                                                                                                                                                                    |
| Tabel yang diakses | Masalah dan Histori                                                                                                                                                                                                                                                                                                  |
| Deskripsi          | Pengguna memasukkan pihak yang bersangkutan dan masalah yang dimiliki mahasiswa pada teks area yang telah disediakan lalu pengguna menyimpan catatan masalah tersebut dengan menekan tombol "Simpan". Pengguna dapat kembali ke modul info mahasiswa tanpa menambah catatan masalah dengan menekan tombol "Kembali". |

### 4.3 Perancangan Diagram Relasional

Berdasarkan ERD pada sub sub bab 3.6.3, maka dapat dihasilkan perancangan diagram relasional yang dapat dilihat pada Gambar 4.11.

### 4.4 Perancangan Tabel Sistem Perekam dan Berbagi Riwayat Mahasiswa

#### 4.4.1 Perancangan Tabel Info Mahasiswa

Untuk rancangan tabel info mahasiswa dapat dilihat pada Tabel 4.10.

#### 4.4.2 Perancangan Tabel Histori

Untuk rancangan tabel histori dapat dilihat pada Tabel 4.11.

#### 4.4.3 Perancangan Tabel Masalah

Untuk rancangan tabel masalah dapat dilihat pada Tabel 4.12.

Tabel 4.7: Modul Lihat Daftar Masalah

|                    |                                                                                                                                                                                                                                                                                                     |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nama Modul         | problem.php                                                                                                                                                                                                                                                                                         |
| Input              | -                                                                                                                                                                                                                                                                                                   |
| Output             | Daftar masalah mahasiswa                                                                                                                                                                                                                                                                            |
| Tabel yang diakses | Masalah                                                                                                                                                                                                                                                                                             |
| Deskripsi          | Pengguna mendapatkan laporan berupa daftar masalah yang dimiliki seorang mahasiswa. Daftar masalah yang dapat dilihat antara lain pihak yang bersangkutan dengan masalah dan rincian / penjelasan masalah tersebut. Pengguna dapat kembali ke modul info mahasiswa dengan menekan tombol "Kembali". |

Tabel 4.8: Modul Lihat Histori

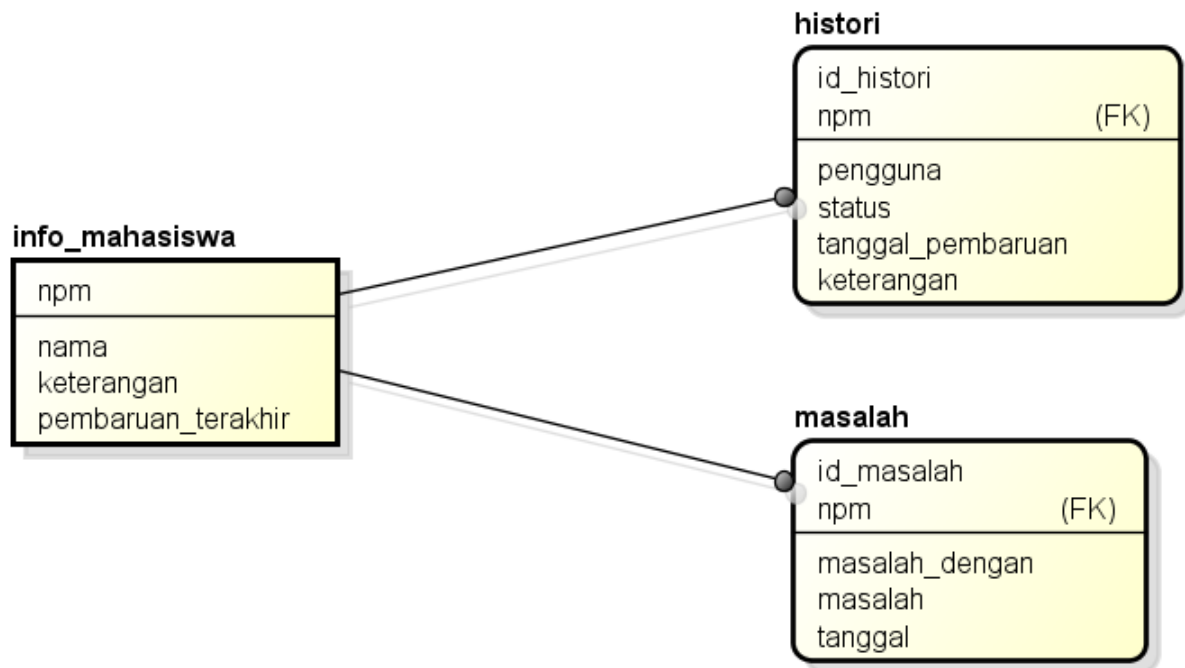
|                    |                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nama Modul         | history.php                                                                                                                                                                                                                                                                                                                                                                                              |
| Input              | -                                                                                                                                                                                                                                                                                                                                                                                                        |
| Output             | Daftar histori mahasiswa                                                                                                                                                                                                                                                                                                                                                                                 |
| Tabel yang diakses | Histori                                                                                                                                                                                                                                                                                                                                                                                                  |
| Deskripsi          | Pengguna mendapatkan laporan berupa daftar histori yang dimiliki setiap mahasiswa. Histori yang dapat dilihat antara lain aksi yang dilakukan pengguna baik aksi membuat entri, melihat dan mengedit. Kemudian berbagai versi keterangan yang dapat dilihat, mulai dari versi pertama kali dibuat hingga versi saat ini. Pengguna dapat kembali ke modul info mahasiswa dengan menekan tombol "Kembali". |

Tabel 4.9: 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, deskripsi umum dan catatan mahasiswa pada teks area yang telah disediakan. Lalu pengguna menyimpan data-data yang telah dimasukkan dengan menekan tombol "Simpan" untuk membuat entri baru tersebut. Pengguna dapat kembali ke modul pilih mahasiswa tanpa melakukan perubahan dengan menekan tombol "Kembali". |

Tabel 4.10: 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          | -          |



Gambar 4.11: Diagram Relasional

Tabel 4.11: 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          | -              |

Tabel 4.12: Rancangan Tabel Masalah

| Atribut        | Tipe Data | Ukuran | Primary Key | Foreign Key | Keterangan     |
|----------------|-----------|--------|-------------|-------------|----------------|
| id_masalah     | int       | 5      | yes         | no          | AUTO_INCREMENT |
| npm            | varchar   | 10     | no          | yes         | -              |
| masalah_dengan | varchar   | 25     | no          | no          | -              |
| masalah        | text      | -      | no          | no          | -              |
| tanggal        | datetime  | -      | 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 dan hasil implementasi perangkat lunak. Bagian implementasi basis data berisi penjelasan basis data yang telah diimplementasi, yang akan digunakan pada 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. Sub bab 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 Mesin Pencari: 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 Lapidan A.

## 5.2 Implementasi Basis Data

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

```
1|CREATE DATABASE 'spbrm' ;
```

Basis data sistem perekam dan berbagi riwayat mahasiswa menggunakan tiga 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| 'catatan' text NOT NULL,
6| 'pembaruan_terakhir' datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
7| PRIMARY KEY ('npm'),
8|) ENGINE=InnoDB DEFAULT CHARSET=latin1 ;
```

- Tabel Masalah, digunakan untuk menyimpan semua data masalah, antara lain masalah dengan siapa / mata kuliah apa dan masalahnya apa. Untuk kode dapat dilihat di bawah ini.

```
1| CREATE TABLE 'masalah' (
2| 'id_masalah' int(5) NOT NULL AUTO_INCREMENT,
3| 'npm' varchar(10) NOT NULL,
4| 'masalah_dengan' varchar(25) NOT NULL,
5| 'masalah' text NOT NULL,
6| 'tanggal' datetime NOT NULL,
7| PRIMARY KEY ('id_masalah'),
8| KEY 'npm' ('npm'),
9|) 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 'catatan' text NOT NULL,
9 PRIMARY KEY ('id_histori'),
10 KEY 'npm' ('npm'),
11) 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 menguji perangkat lunak langsung ke pengguna. Pada bagian pengujian untuk pengujian fungsional 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 fungsional perangkat lunak dapat dilihat pada Lampiran B.

### 5.3.1 Lingkungan Pengujian Perangkat Keras

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

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

### 5.3.2 Lingkungan Pengujian Perangkat Lunak

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

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

- (e) Aplikasi Mesin Pencari: 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 "bleack@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 mencari mahasiswa

Pengujian fungsi ini dilakukan untuk memastikan pengguna dapat mencari mahasiswa yang

ingin dicarinya. Pada halaman list.php terdapat tempat untuk memasukkan npm dari mahasiswa yang ingin dicari. Setelah memasukkan npm yang ingin dicari, kemudian pengguna menekan tombol "Cari NPM".

Contoh kasus adalah melakukan pencarian mahasiswa dengan memasukkan npm seutuhnya dan sebagian npm. Pengujian pertama, pengguna memasukkan npm 2010730013 sebagai npm yang ingin dicari, dapat dilihat pada Gambar 5.9. Kemudian pengguna menekan tombol "Cari NPM" sehingga pengguna mendapatkan hasil dari pencarian. Untuk hasil pencarian dapat dilihat pada Gambar 5.10. Pengujian kedua, pengguna memasukkan sebagian npm 2010 sebagai npm yang ingin dicari, dapat dilihat pada Gambar 5.11. Kemudian pengguna menekan tombol "Cari NPM" sehingga pengguna mendapatkan hasil dari pencarian. Untuk hasil pencarian dapat dilihat pada Gambar 5.12.

(4) 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. Untuk gambar dapat dilihat pada Gambar 5.13. Hal ini menunjukkan fungsi melihat info mahasiswa sudah berjalan dengan baik.

(5) 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". Untuk gambar dapat dilihat pada Gambar 5.14. Setelah menyimpan perubahan, pengguna akan dibawa kembali ke halaman list.php. Hal ini menunjukkan fungsi mengedit info mahasiswa sudah berjalan dengan baik.

(6) Fungsi menambah masalah baru

Pengujian fungsi ini dilakukan untuk memastikan pengguna dapat menambah masalah baru dari seorang mahasiswa. Contoh pengujian menambahkan masalah baru dari seorang mahasiswa dengan mengisi form masalah baru, masalah dengan mata kuliah "AJK2" dan masalahnya "jarang mengikuti kelas praktikum dan jarang mengumpulkan tugas". Untuk gambar dapat dilihat pada Gambar 5.15. Setelah menyimpan masalah baru, pengguna akan dibawa kembali ke halaman list.php. Hal ini menunjukkan fungsi menambah masalah baru sudah berjalan dengan baik.

(7) Fungsi melihat daftar masalah

Pengujian fungsi ini dilakukan untuk memastikan adanya daftar masalah dari seorang mahasiswa yang dipilih. Contoh pengujian melihat daftar masalah dari seorang mahasiswa yang memiliki npm 2010730013 dapat dilihat pada Gambar 5.16. Hal ini menunjukkan fungsi melihat daftar masalah sudah berjalan dengan baik.

(8) Fungsi melihat histori

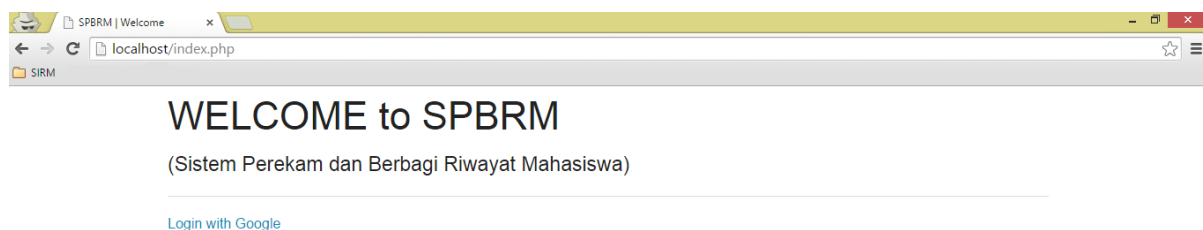
Pengujian fungsi ini dilakukan untuk memastikan adanya histori dari seorang mahasiswa yang dipilih dan dapat melihat versi keterangan yang pertama kali dibuat dan versi-versi berikutnya yang sudah dirubah. Contoh pengujian melihat histori dari seorang mahasiswa yang memiliki npm 2010730013 dapat dilihat pada Gambar 5.17 dan juga melihat keterangan versi pertama berserta versi berikutnya dapat dilihat pada Gambar 5.18 dan Gambar 5.19. Hal ini menunjukkan fungsi melihat histori sudah berjalan dengan baik.

(9) Fungsi membuat entri baru

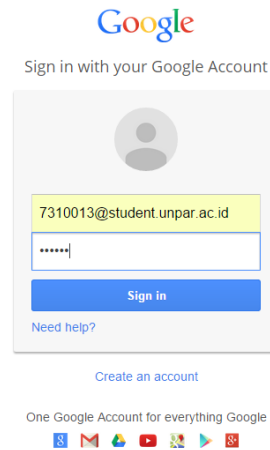
Pengujian fungsi ini dilakukan untuk memastikan pada saat membuat entri baru terdapat format penulisan yang telah dibuat dengan Markdown dan berhasil menyimpan entri baru tersebut. Contoh pengujian menambahkan entri baru untuk mahasiswa yang memiliki npm "2010730005", nama "Grady Ireneus", deskripsi umum "Mahasiswa ini telah menyelesaikan setengah skripsi yang diambil. Sisanya akan diselesaikan semester depan" dan catatan "6 Juli 2015, Melakukan perwalian". Mengisi data npm, nama, deskripsi umum dan catatan sesuai penjabaran sebelumnya, untuk gambar dapat dilihat pada Gambar 5.20. 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 2010730005 dan nama Grady Ireneus telah masuk kedalam tabel dapat dilihat pada Gambar 5.21. Hal ini menunjukkan fungsi membuat entri baru sudah berjalan dengan baik.

(10) Antarmuka yang responsif

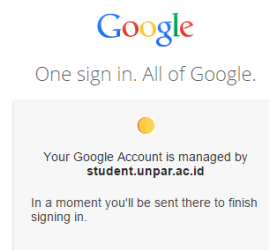
Pengujian antarmuka yang responsif dilakukan untuk memastikan tampilan antarmuka yang dibuat menggunakan Zurb Foundation berhasil. Contoh pengujian dilakukan dengan menggunakan mesin pencari yang telah dikecilkan ukurannya, lalu membuka index.php, list.php, dan new.php pada mesin pencari tersebut. Pengujian untuk index.php dapat dilihat pada Gambar 5.22, untuk list.php dapat dilihat pada Gambar 5.23, dan untuk new.php dapat dilihat pada Gambar 5.24. Hal ini menunjukkan antarmuka yang responsif sudah berjalan dengan baik.



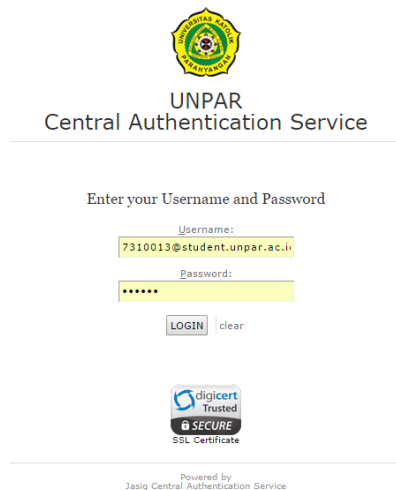
Gambar 5.1: Membuka Halaman index.php



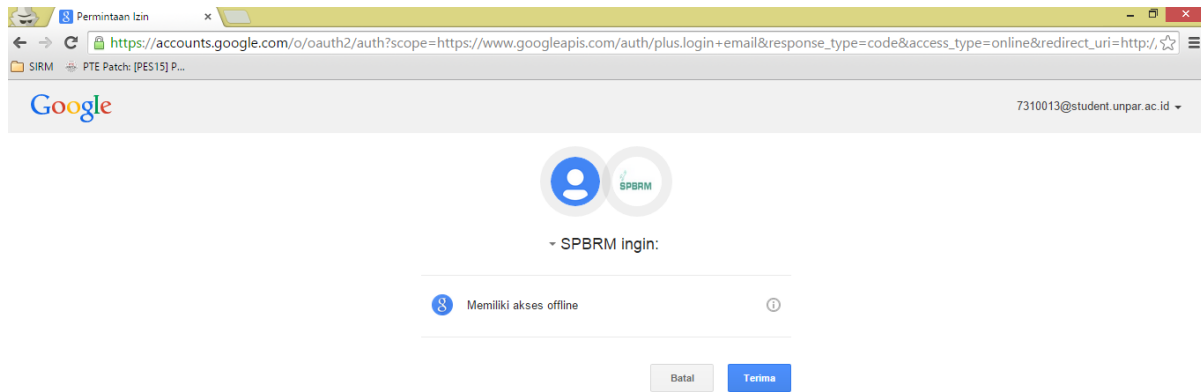
Gambar 5.2: Login Dengan Email yang Diakhiri "@student.unpar.ac.id"



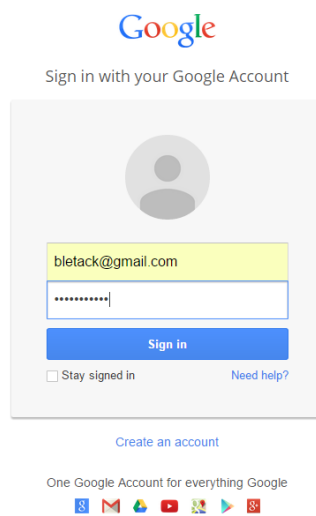
Gambar 5.3: Konfirmasi Email yang Dikelola oleh student.unpar.ac.id



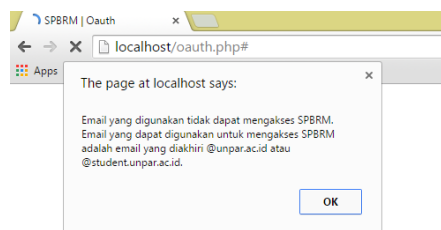
Gambar 5.4: CAS UNPAR



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 SPBRM



Masukkan dan pilih NPM yang ingin dicari / tambah baru.

Cari NPM Add Logout

| NPM                        | Nama     | Last Update         |
|----------------------------|----------|---------------------|
| <a href="#">2010730001</a> | Andri AG | 2015-07-06 21:46:33 |
| <a href="#">2010730012</a> | Kevin PL | 2015-07-06 21:47:44 |
| <a href="#">2010730013</a> | Samuel H | 2015-07-06 22:06:38 |

localhost/view.php?npm=2010730013

Gambar 5.8: Memilih Mahasiswa

Masukkan dan pilih NPM yang ingin dicari / tambah baru.

Cari NPM Add Logout

| NPM                        | Nama     | Last Update         |
|----------------------------|----------|---------------------|
| <a href="#">2010730001</a> | Andri AG | 2015-07-06 21:46:33 |
| <a href="#">2010730012</a> | Kevin PL | 2015-07-06 21:47:44 |
| <a href="#">2010730013</a> | Samuel H | 2015-07-06 22:06:38 |

Gambar 5.9: Mencari Mahasiswa Dengan NPM Seutuhnya

localhost/search.php?search=2010730013&submit=Cari+NPM

Masukkan dan pilih NPM yang ingin dicari / tambah baru.

Cari NPM Add Menu Utama Logout

| NPM                        | Nama     | Last Update         |
|----------------------------|----------|---------------------|
| <a href="#">2010730013</a> | Samuel H | 2015-07-06 22:06:38 |

Gambar 5.10: Hasil Mencari Mahasiswa Dengan NPM Seutuhnya

Masukkan dan pilih NPM yang ingin dicari / tambah baru.

[Cari NPM](#)
[Add](#)
[Menu Utama](#)
[Logout](#)

| NPM        | Nama     | Last Update         |
|------------|----------|---------------------|
| 2010730013 | Samuel H | 2015-07-06 22:06:38 |

Gambar 5.11: Mencari Mahasiswa Dengan Sebagian NPM

localhost/search.php?search=2010&submit=Cari+NPM

Masukkan dan pilih NPM yang ingin dicari / tambah baru.

[Cari NPM](#)
[Add](#)
[Menu Utama](#)
[Logout](#)

| NPM        | Nama     | Last Update         |
|------------|----------|---------------------|
| 2010730001 | Andri AG | 2015-07-06 21:46:33 |
| 2010730012 | Kevin PL | 2015-07-06 21:47:44 |
| 2010730013 | Samuel H | 2015-07-06 22:06:38 |

Gambar 5.12: Hasil Mencari Mahasiswa Dengan Sebagian NPM

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

[Menu Utama](#)
[Edit](#)
[Masalah Baru](#)
[Lihat Masalah](#)
[Lihat Histori](#)
[Logout](#)

NPM : 2010730013  
Nama : Samuel H

### Deskripsi Umum

Mahasiswa ini sedang mengambil tugas akhir dan sekarang sedang melakukan tahap finishing.

### Catatan

- 6 Juli 2015, Melakukan bimbingan

Gambar 5.13: Melihat Info Mahasiswa

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

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

#### Deskripsi Umum

Mahasiswa ini sedang mengambil skripsi dan sekarang sedang melakukan tahap finishing.  
Sudah lulus sidang skripsi dan sekarang sedang menyelesaikan revisi supaya dapat mengikuti sidang yudisium.

#### Catatan

6 Juli 2015, Melakukan bimbingan

Gambar 5.14: Mengedit Info Mahasiswa

Anda menambah catatan masalah mahasiswa ini sebagai  
7310013@student.unpar.ac.id.  
NPM 2010730013 Nama Samuel H

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

#### Masalah dengan siapa / mata kuliah apa

AJK2

#### Masalah

jarang mengikuti kelas praktikum dan jarang mengumpulkan tugas

Gambar 5.15: Menambah Catatan Masalah Baru

Daftar masalah yang dimiliki NPM 2010730013 Nama Samuel H

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

- 2015-07-06 22:41:51 Memiliki masalah dengan/pada kelas AJK2, masalahnya adalah jarang mengikuti kelas praktikum dan jarang mengumpulkan tugas.

Gambar 5.16: Melihat Daftar Masalah

## Daftar histori yang dimiliki NPM 2010730013 Nama Samuel H

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

- 2015-07-06 22:47:11 7310013@student.unpar.ac.id melihat 2010730013
- 2015-07-06 22:47:05 7310013@student.unpar.ac.id mengedit 2010730013 [\[lihat versi ini\]](#)
- 2015-07-06 22:47:01 7310013@student.unpar.ac.id melihat 2010730013
- 2015-07-06 22:46:33 7310013@student.unpar.ac.id melihat 2010730013
- 2015-07-06 22:46:20 7310013@student.unpar.ac.id melihat 2010730013
- 2015-07-06 22:43:44 7310013@student.unpar.ac.id melihat 2010730013
- 2015-07-06 22:41:57 7310013@student.unpar.ac.id melihat 2010730013
- 2015-07-06 22:41:52 7310013@student.unpar.ac.id menambah catatan masalah 2010730013
- 2015-07-06 22:36:44 7310013@student.unpar.ac.id melihat 2010730013
- 2015-07-06 22:35:57 7310013@student.unpar.ac.id mengedit 2010730013 [\[lihat versi ini\]](#)
- 2015-07-06 22:26:50 7310013@student.unpar.ac.id melihat 2010730013
- 2015-07-06 22:06:38 7310013@student.unpar.ac.id membuat entri 2010730013 [\[lihat versi ini\]](#)

Gambar 5.17: Melihat Histori

|         |        |
|---------|--------|
| Kembali | Logout |
|---------|--------|

## Deskripsi Umum

Mahasiswa ini sedang mengambil tugas akhir dan sekarang sedang melakukan tahap finishing.

## Catatan

- 6 Juli 2015, Melakukan bimbingan

Gambar 5.18: Keterangan Versi Pertama

|         |        |
|---------|--------|
| Kembali | Logout |
|---------|--------|

## Deskripsi Umum

Mahasiswa ini sedang mengambil skripsi dan sekarang sedang melakukan tahap finishing.

Sudah lulus sidang skripsi dan sekarang sedang menyelesaikan revisi supaya dapat mengikuti sidang yudisium.

## Catatan

- 6 Juli 2015, Melakukan bimbingan

Gambar 5.19: Keterangan Versi Kedua

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

Kembali Simpan Menu Utama Logout

NPM

2010730005

Nama

Grady Ireneus

Deskripsi Umum

Mahasiswa ini telah menyelesaikan setengah skripsi yang diambil. Sisanya akan diselesaikan semester depan.

Catatan

6 Juli 2015, Melakukan perwalian

Gambar 5.20: Membuat Entri Baru

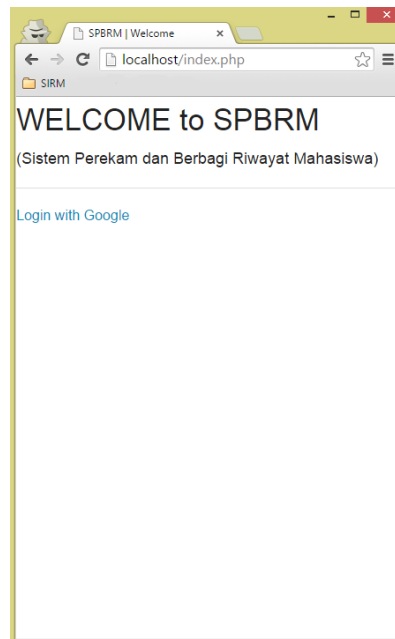
Masukkan dan pilih NPM yang ingin dicari / tambah baru.

Masukkan NPM yang ingin dicari

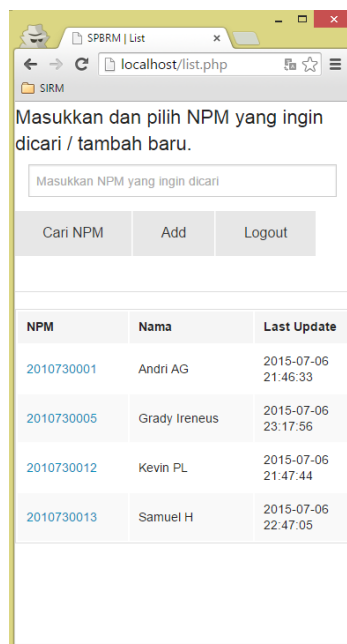
Cari NPM Add Logout

| NPM        | Nama          | Last Update         |
|------------|---------------|---------------------|
| 2010730001 | Andri AG      | 2015-07-06 21:46:33 |
| 2010730005 | Grady Ireneus | 2015-07-06 23:17:56 |
| 2010730012 | Kevin PL      | 2015-07-06 21:47:44 |
| 2010730013 | Samuel H      | 2015-07-06 22:47:05 |

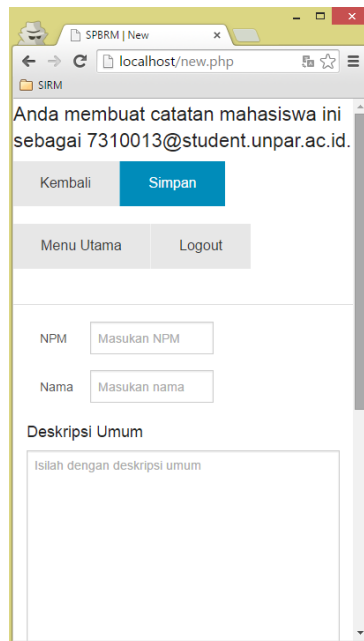
Gambar 5.21: Entri Baru Berhasil Dibuat



Gambar 5.22: Antarmuka Responsif index.php



Gambar 5.23: Antarmuka Responsif list.php



Gambar 5.24: Antarmuka Responsif new.php

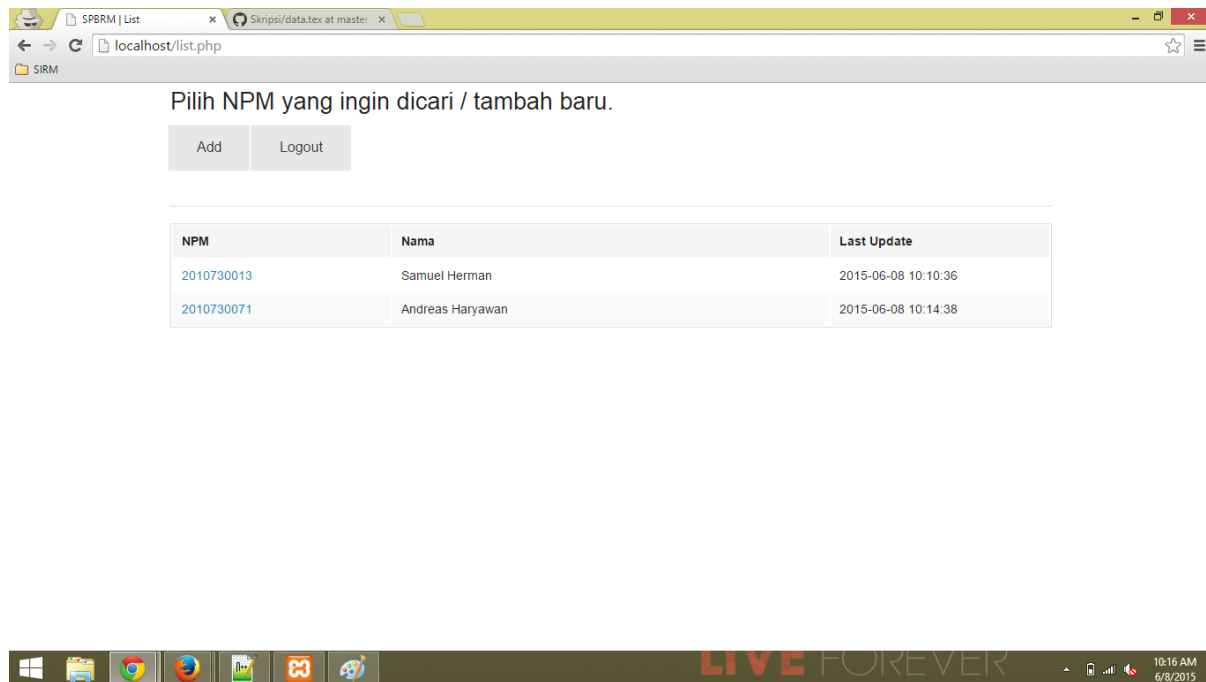
#### 5.3.4 Pengujian Eksperimental

Pengujian eksperimental dilakukan langsung ke lima orang dosen Teknik Informatika. Kelima dosen menguji dengan cara mencoba semua fitur yang dimiliki SPBRM. Kelima dosen juga menjalankan SPBRM dengan memasukkan data riwayat mahasiswa yang sebenarnya. Setelah melakukan pengujian diakhiri dengan kuesioner, untuk kuesioner pengujian eksperimental dapat dilihat pada Lampiran E. Untuk beberapa *screenshot* pada pengujian eksperimental dapat dilihat pada Gambar 5.25-5.40. *Screenshot* tersebut merupakan beberapa bagian dari pengujian eksperimental yang dilakukan oleh penguji pertama, kedua, dan ketiga. Berdasarkan *screenshot* tersebut juga dibuktikan bahwa SPBRM berhasil mengautentikasi dan mengotorisasi akun dosen, lalu SPBRM juga berhasil merakam dan membagikan semua riwayat mahasiswa ke setiap dosen.

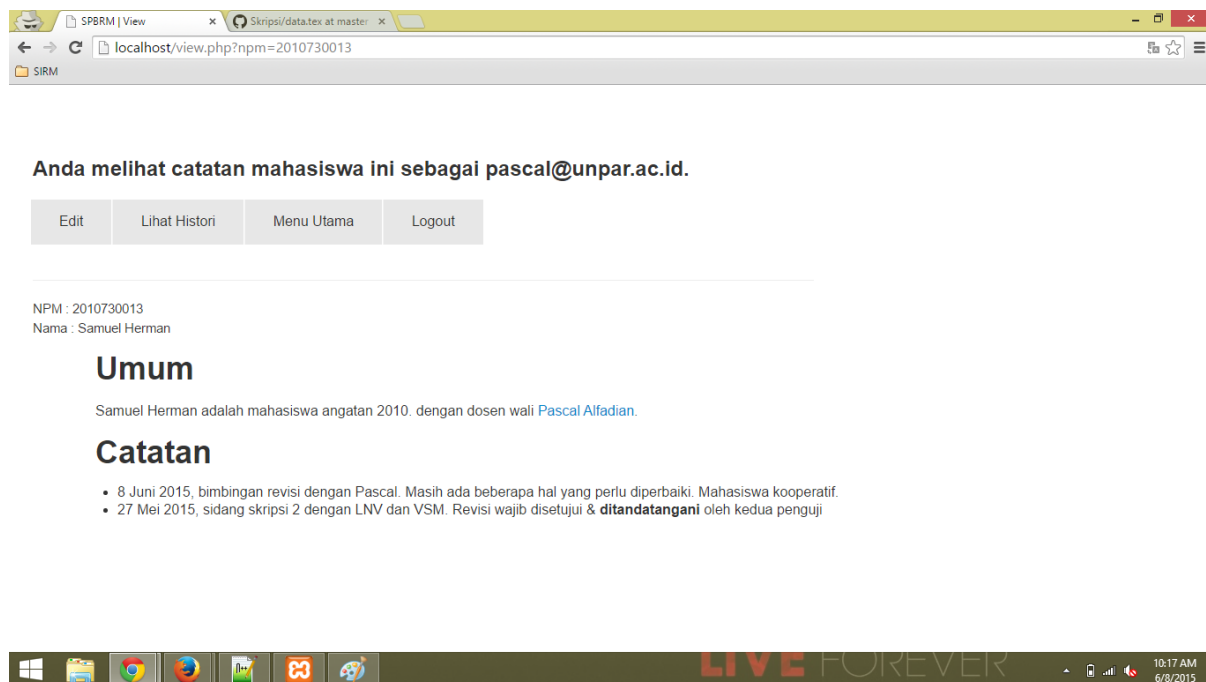
Untuk kuesioner dibuat menggunakan Google Form dan kuesioner disebarkan menggunakan email sehingga urutan penguji kemungkinan tidak sesuai. Berikut data hasil kuesioner pengujian eksperimental, dapat dilihat pada Tabel 5.1-5.5.

Tabel 5.1: Tabel Jawaban Pertanyaan Pertama, SPBRM Membantu Mengingat

| No Penguji | Sangat Setuju | Setuju | Netral | Tidak Setuju | Sangat Tidak Setuju |
|------------|---------------|--------|--------|--------------|---------------------|
| 1          |               |        | ✓      |              |                     |
| 2          |               | ✓      |        |              |                     |
| 3          |               | ✓      |        |              |                     |
| 4          | ✓             |        |        |              |                     |
| 5          | ✓             |        |        |              |                     |



Gambar 5.25: Penguji Pertama Mengakses Halaman Utama

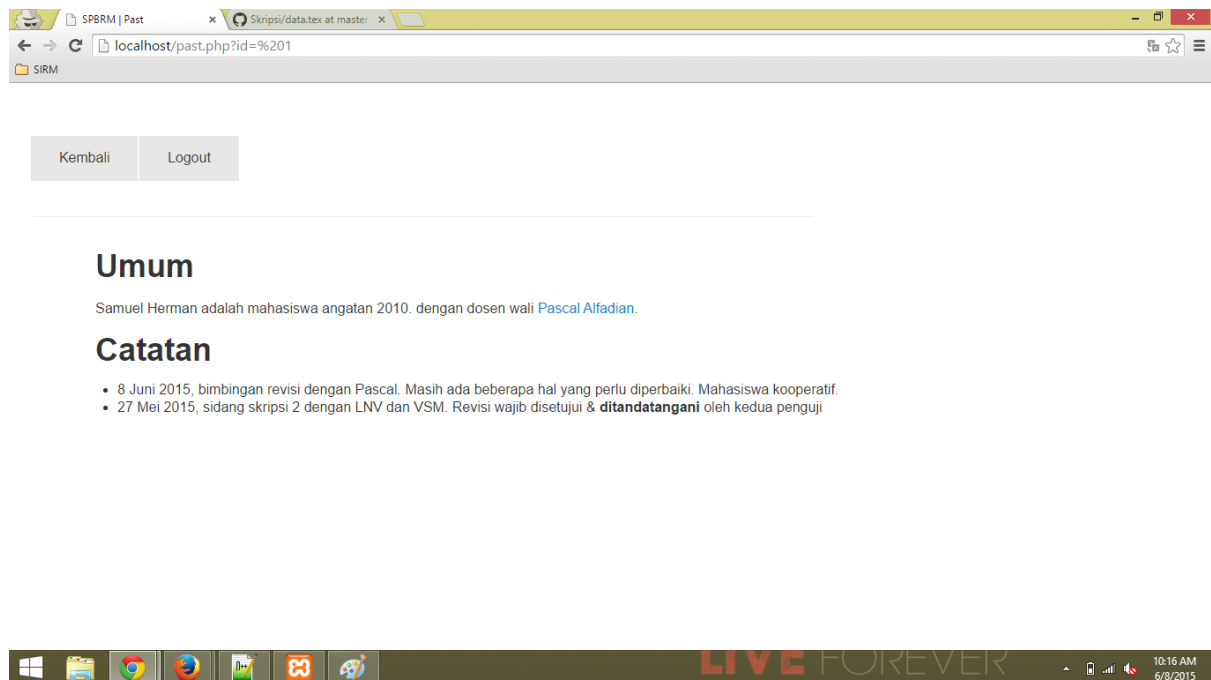


Gambar 5.26: Penguji Pertama Mengakses Info Mahasiswa Dengan NPM 2010730013

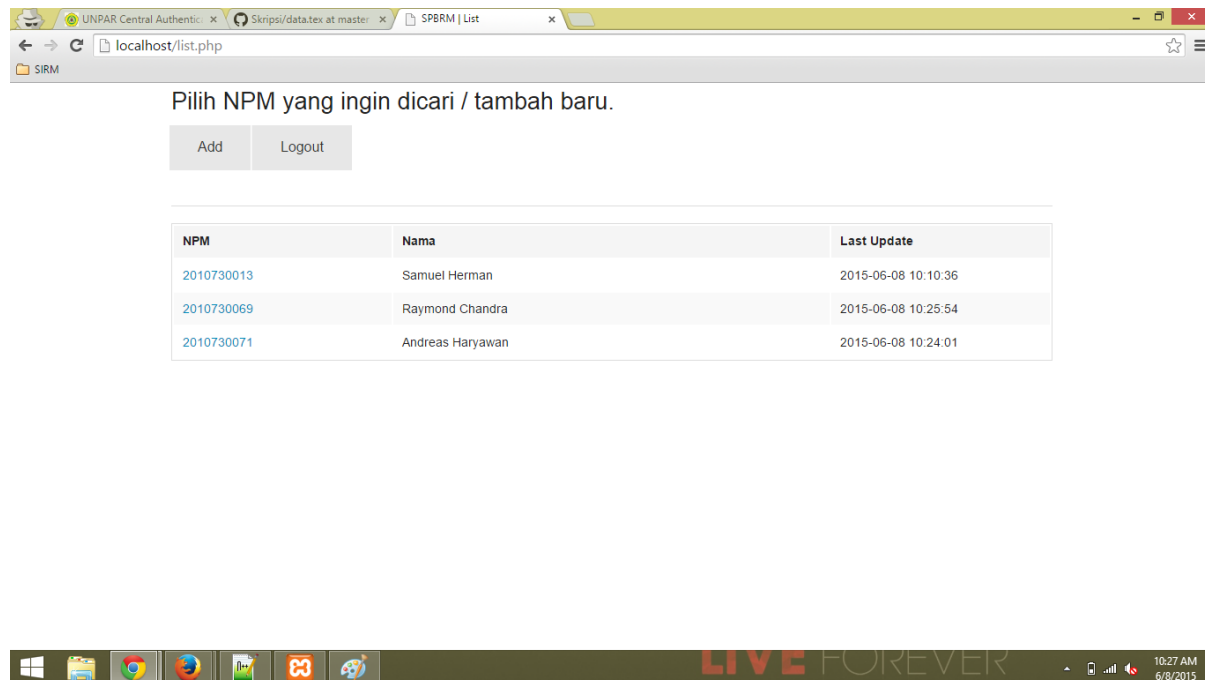




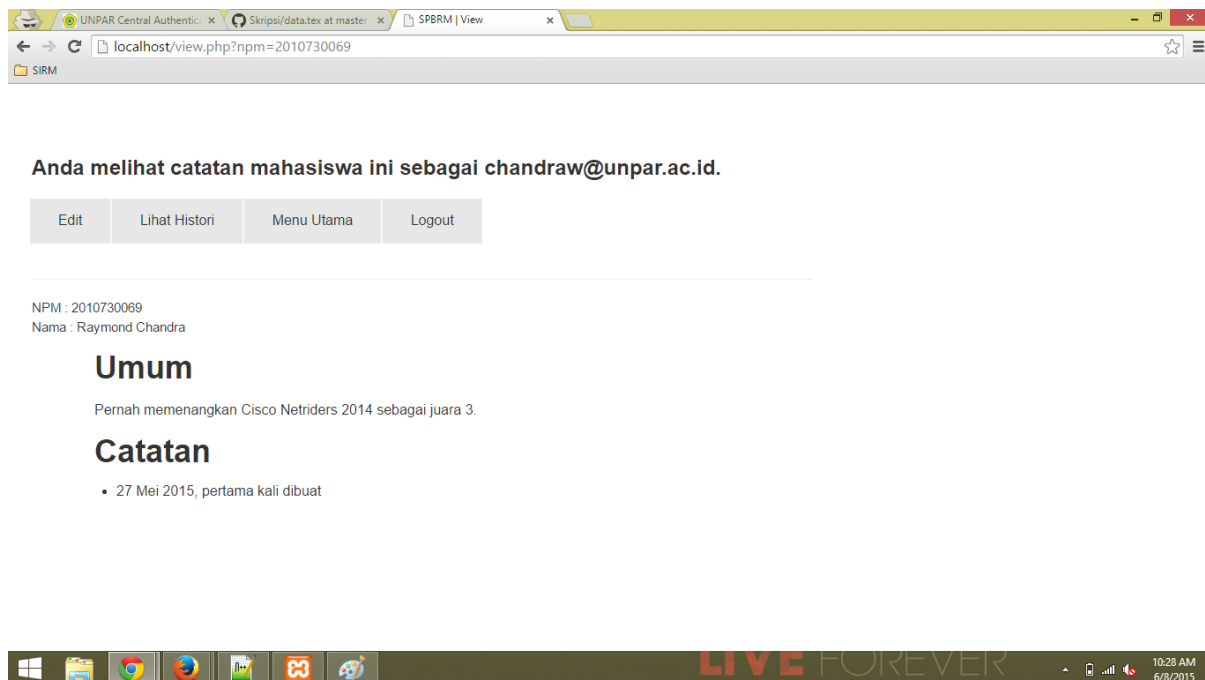
Gambar 5.27: Penguji Pertama Mengakses *History* Mahasiswa Dengan NPM 2010730013



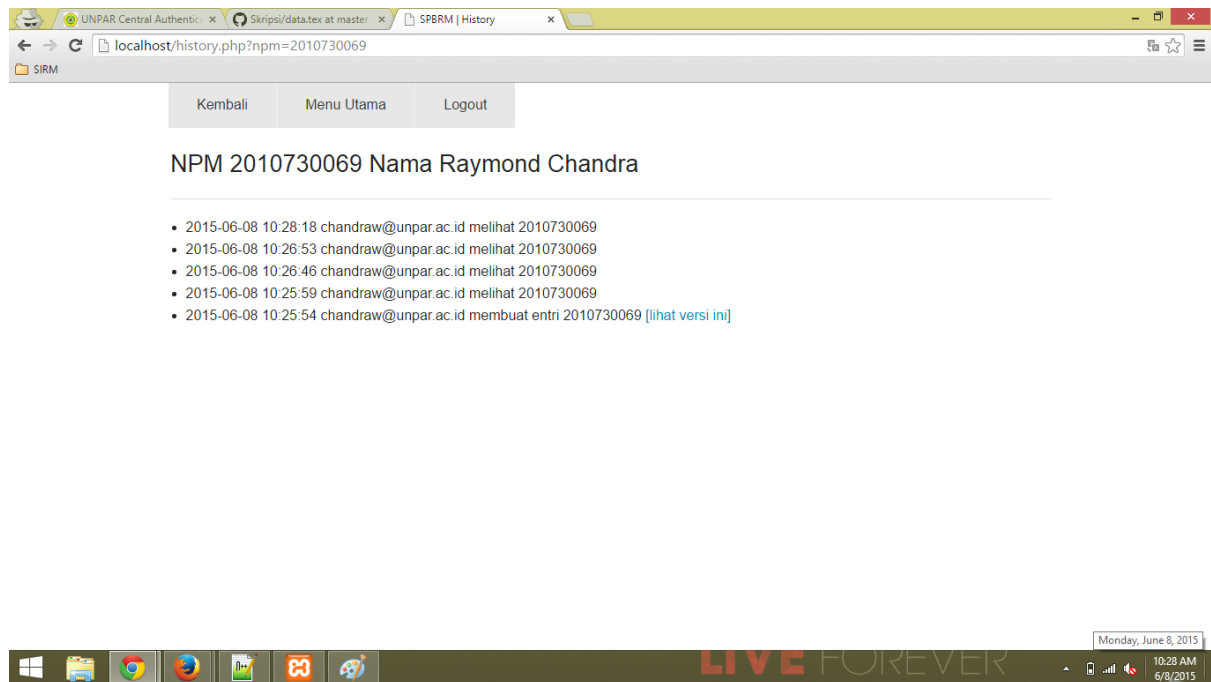
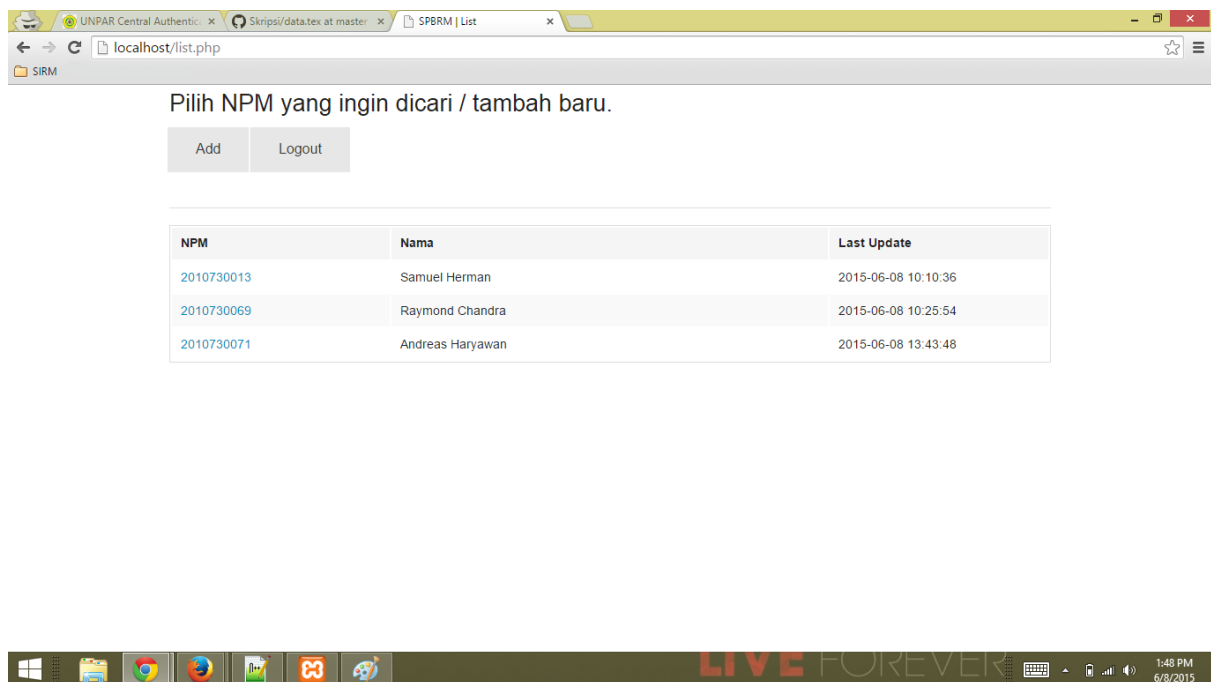
Gambar 5.28: Penguji Pertama Melihat Versi Ini Untuk Mahasiswa Dengan NPM 2010730013



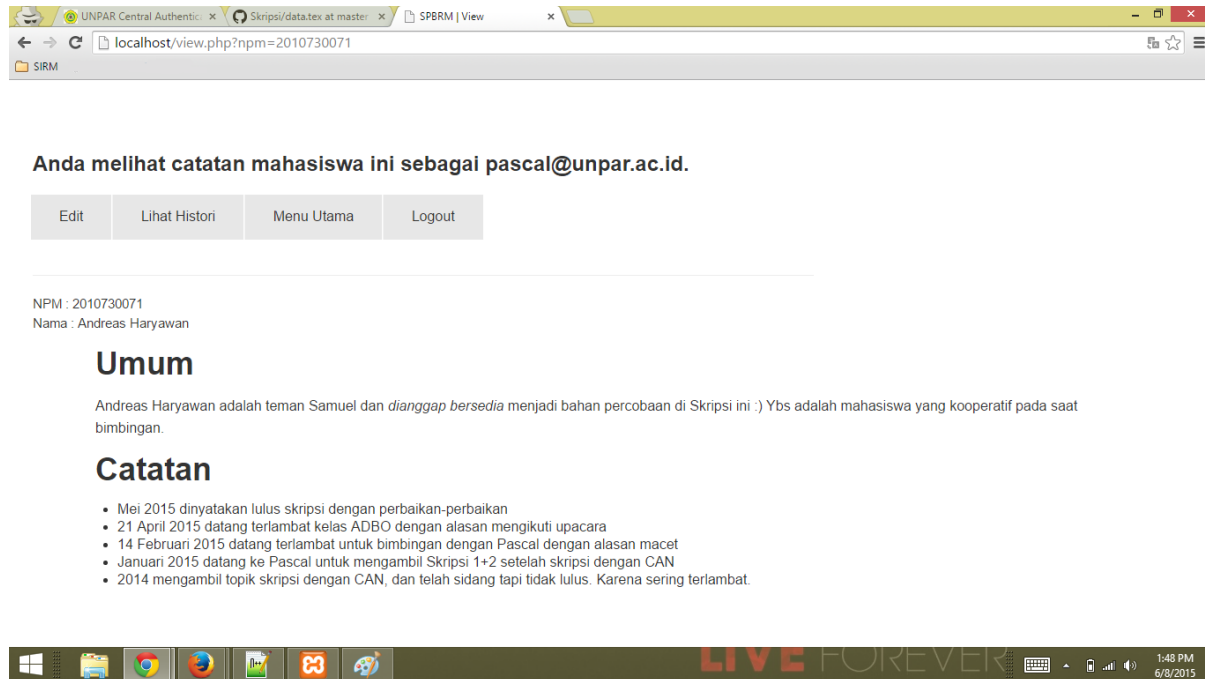
Gambar 5.29: Penguji Kedua Mengakses Halaman Utama



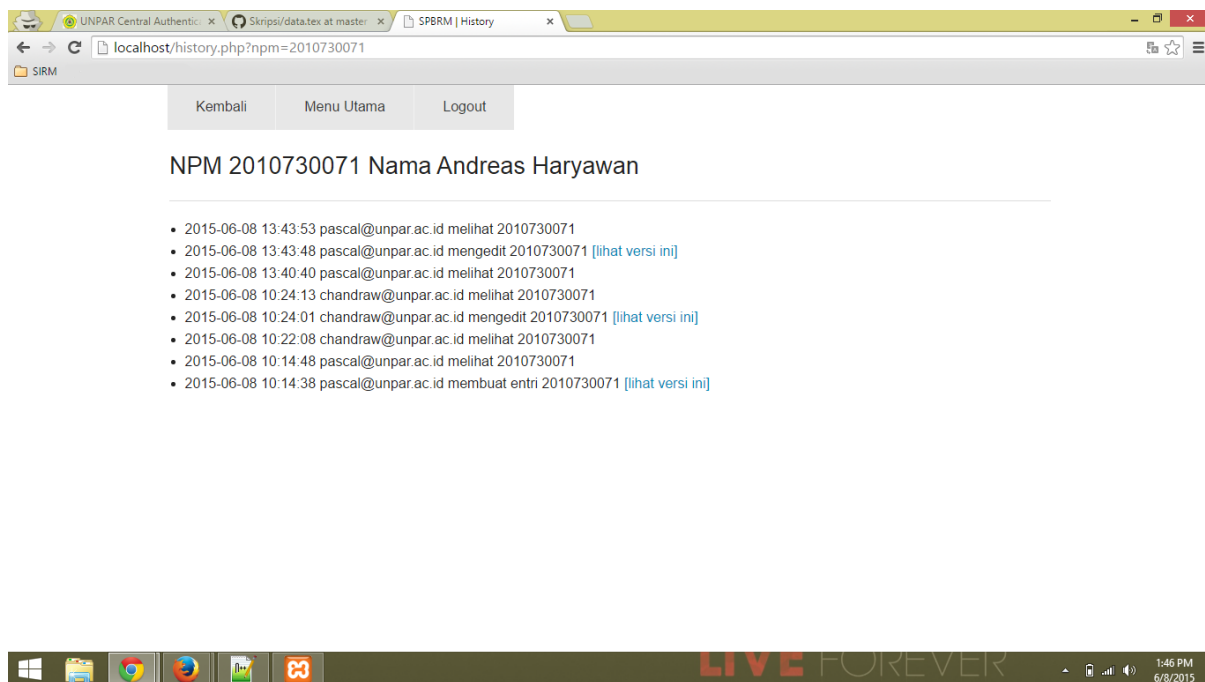
Gambar 5.30: Penguji Kedua Mengakses Info Mahasiswa Dengan NPM 2010730069

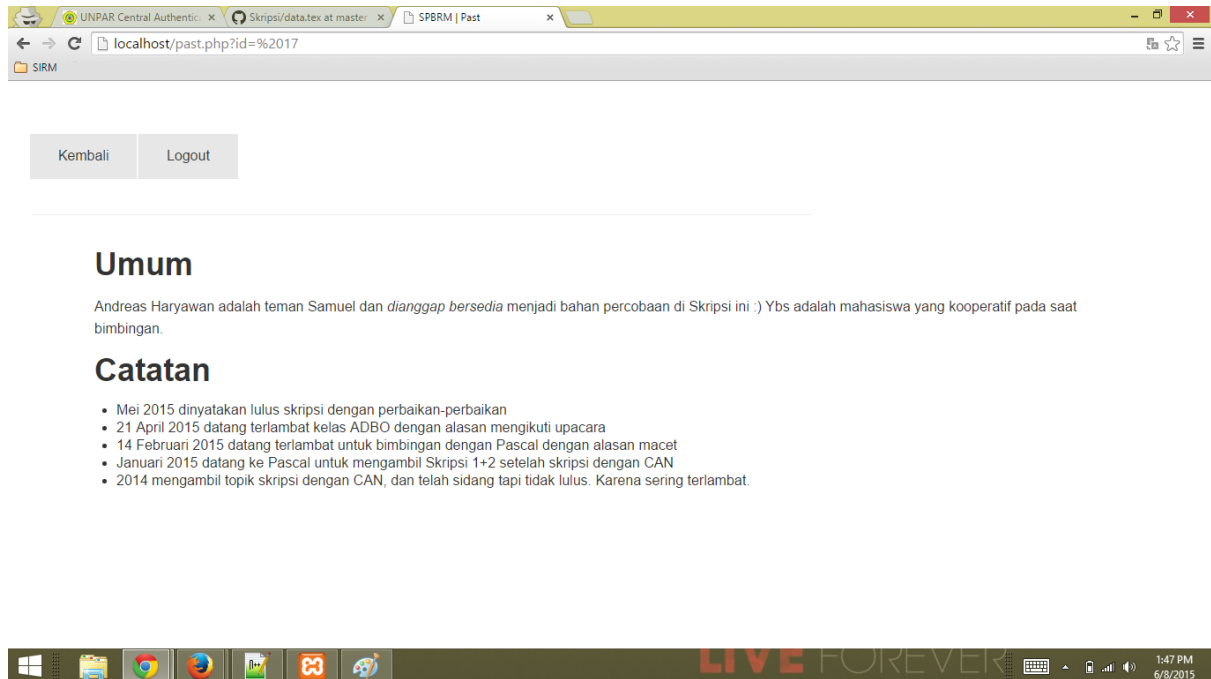
Gambar 5.31: Penguji Kedua Mengakses *History* Mahasiswa Dengan NPM 2010730069

Gambar 5.32: Penguji Pertama Mengakses Halaman Utama Untuk Kedua Kali

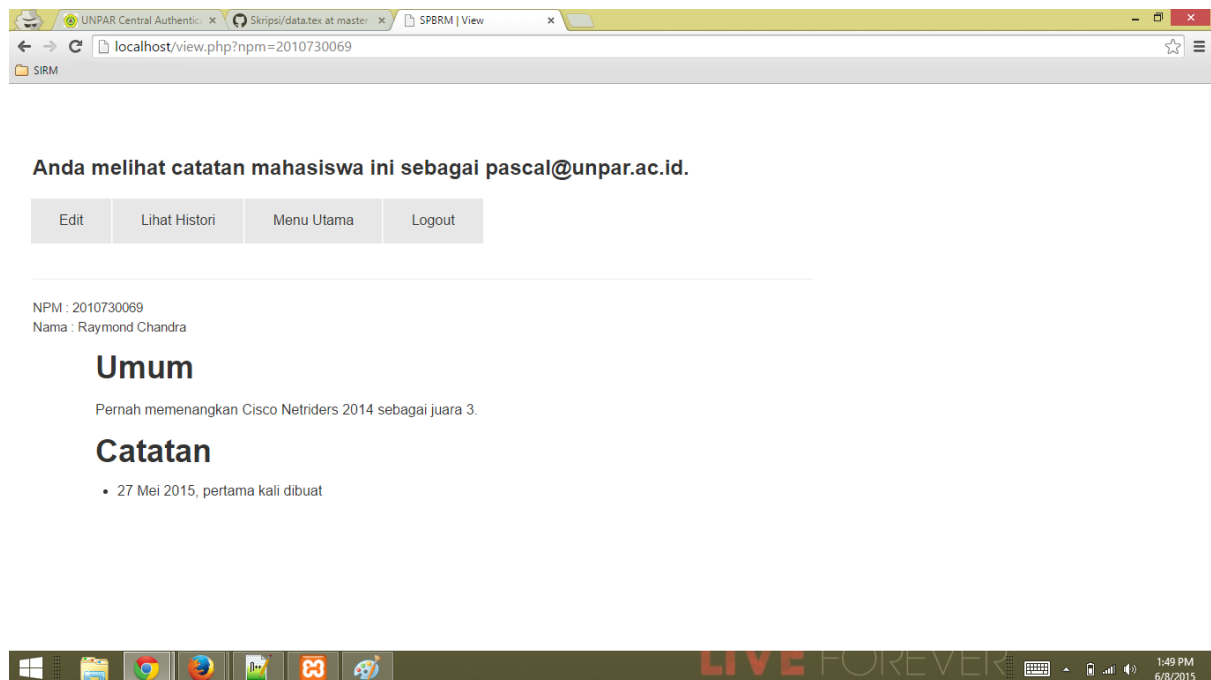


Gambar 5.33: Penguji Pertama Mengakses Info Mahasiswa Dengan NPM 2010730071

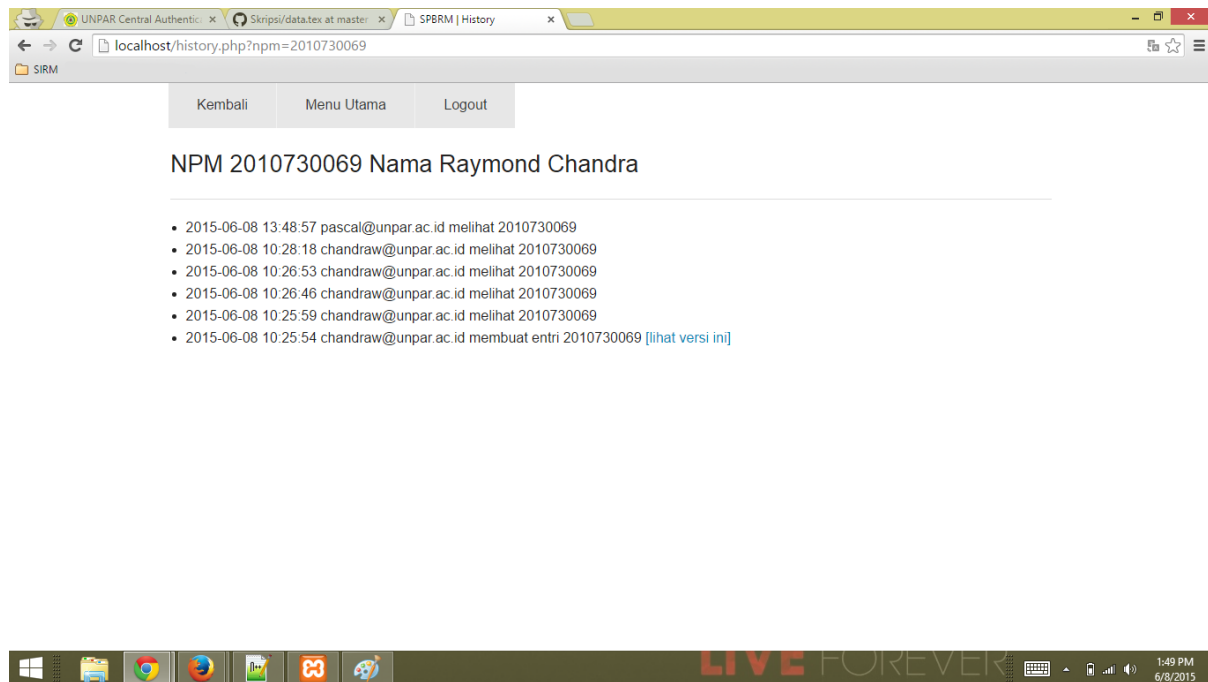
Gambar 5.34: Penguji Pertama Mengakses *History* Mahasiswa Dengan NPM 2010730071



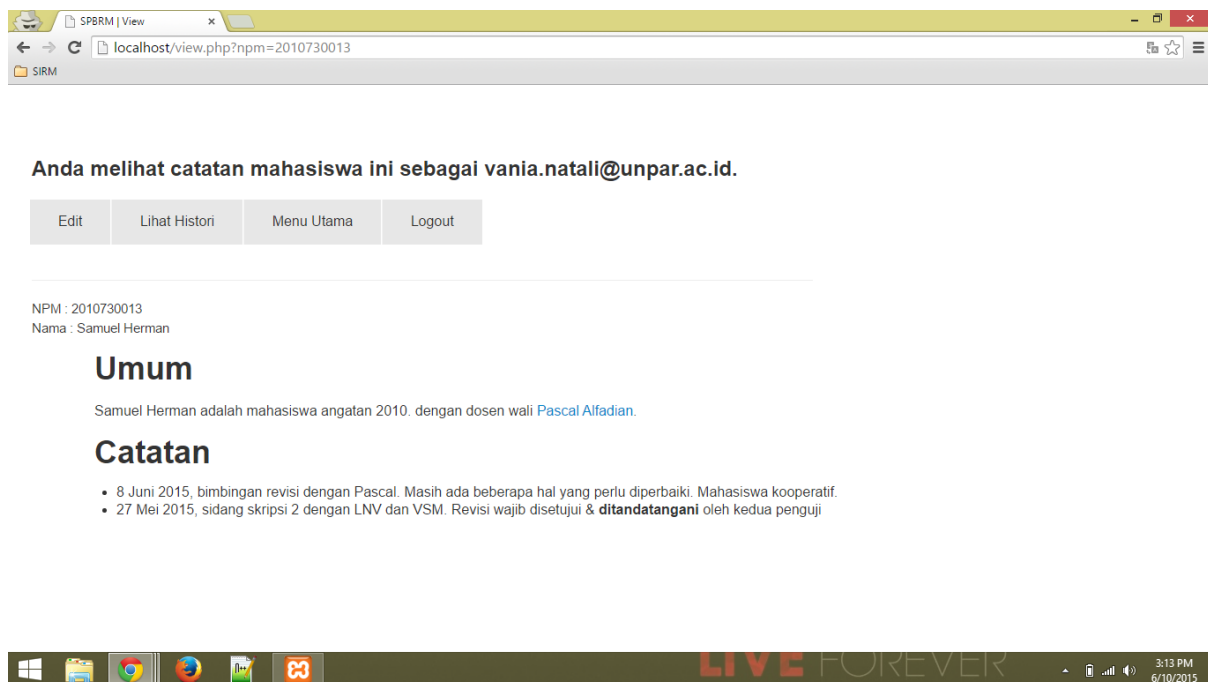
Gambar 5.35: Penguji Pertama Melihat Versi Ini Untuk Mahasiswa Dengan NPM 2010730071



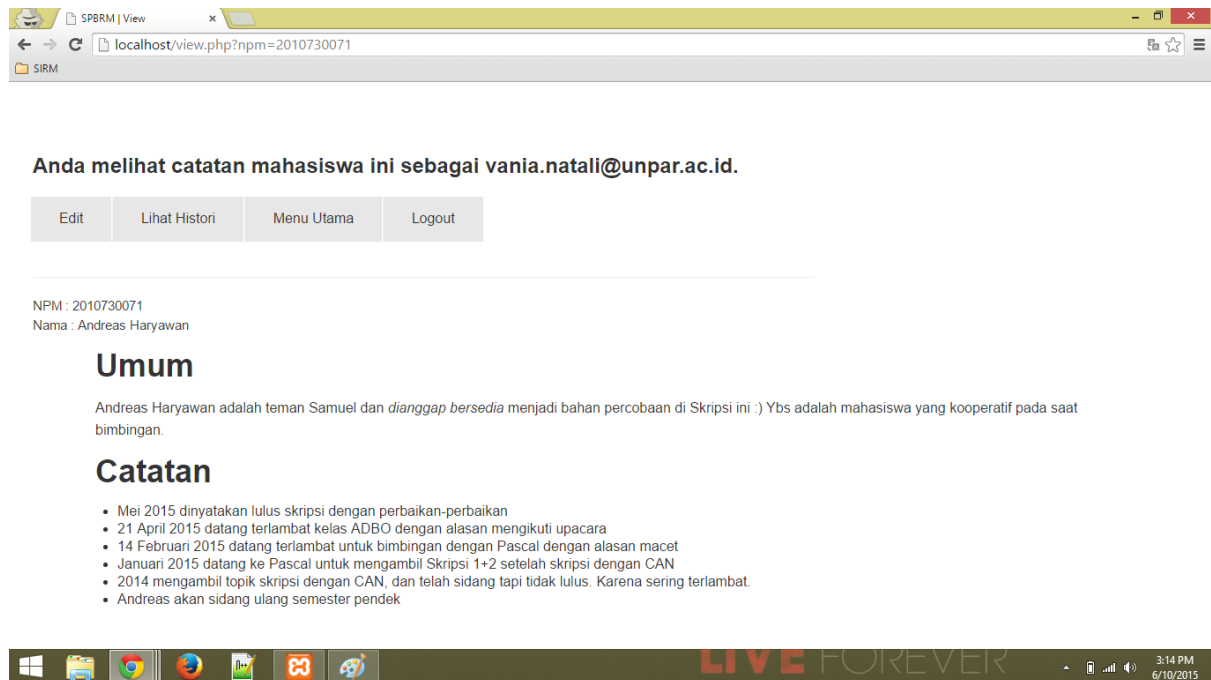
Gambar 5.36: Penguji Pertama Mengakses Info Mahasiswa Dengan NPM 2010730069



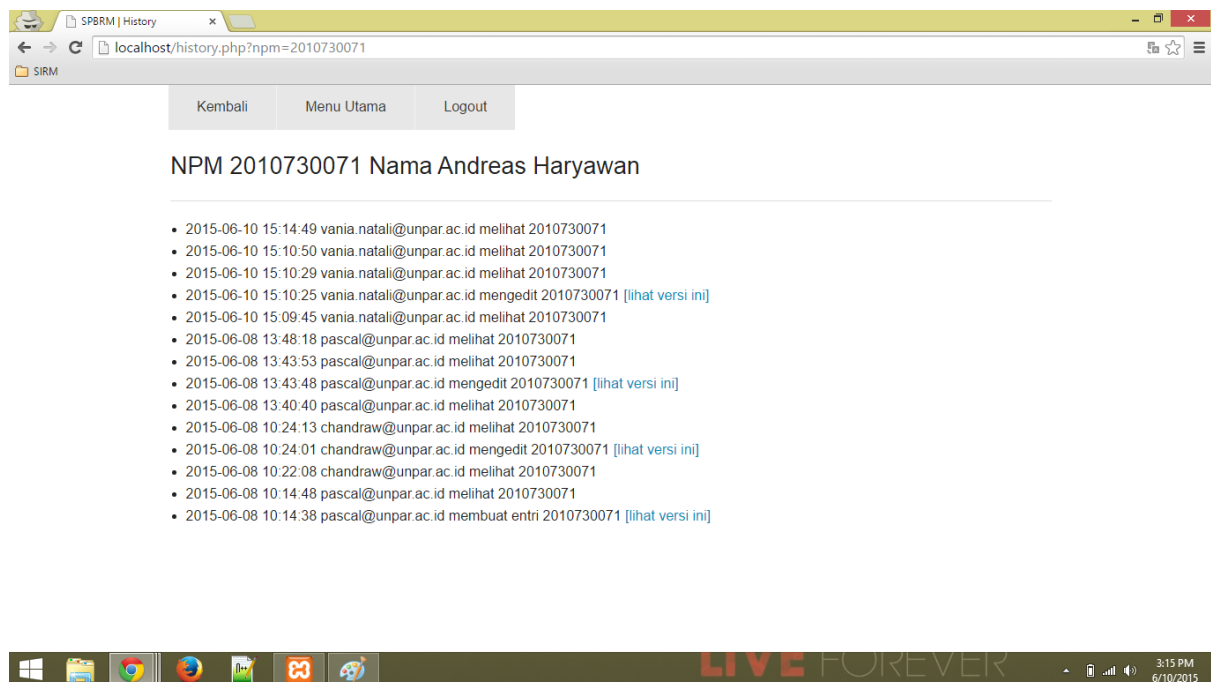
Gambar 5.37: Penguji Pertama Mengakses *History* Mahasiswa Dengan NPM 2010730069



Gambar 5.38: Penguji Ketiga Mengakses Info Mahasiswa Dengan NPM 2010730013



Gambar 5.39: Penguji Ketiga Mengakses Info Mahasiswa Dengan NPM 2010730071

Gambar 5.40: Penguji Ketiga Mengakses *History* Mahasiswa Dengan NPM 2010730071

Tabel 5.2: Tabel Jawaban Pertanyaan Kedua, Kemudahan SPBRM

| No Penguji | Sangat Setuju | Setuju | Netral | Tidak Setuju | Sangat Tidak Setuju |
|------------|---------------|--------|--------|--------------|---------------------|
| 1          |               | ✓      |        |              |                     |
| 2          |               |        | ✓      |              |                     |
| 3          |               |        |        | ✓            |                     |
| 4          |               |        | ✓      |              |                     |
| 5          | ✓             |        |        |              |                     |

Tabel 5.3: Tabel Jawaban Pertanyaan Ketiga, SPBRM Membantu Pemahaman

| No Penguji | Sangat Setuju | Setuju | Netral | Tidak Setuju | Sangat Tidak Setuju |
|------------|---------------|--------|--------|--------------|---------------------|
| 1          |               | ✓      |        |              |                     |
| 2          |               | ✓      |        |              |                     |
| 3          |               |        | ✓      |              |                     |
| 4          |               | ✓      |        |              |                     |
| 5          | ✓             |        |        |              |                     |

Tabel 5.4: Tabel Jawaban Pertanyaan Keempat, SPBRM Efektif

| No Penguji | Sangat Setuju | Setuju | Netral | Tidak Setuju | Sangat Tidak Setuju |
|------------|---------------|--------|--------|--------------|---------------------|
| 1          |               |        | ✓      |              |                     |
| 2          |               | ✓      |        |              |                     |
| 3          | ✓             |        |        |              |                     |
| 4          | ✓             |        |        |              |                     |
| 5          | ✓             |        |        |              |                     |

Tabel 5.5: Tabel Jawaban Pertanyaan Kelima, SPBRM Efisien

| No Penguji | Sangat Setuju | Setuju | Netral | Tidak Setuju | Sangat Tidak Setuju |
|------------|---------------|--------|--------|--------------|---------------------|
| 1          |               |        | ✓      |              |                     |
| 2          |               |        | ✓      |              |                     |
| 3          | ✓             |        |        |              |                     |
| 4          | ✓             |        |        |              |                     |
| 5          | ✓             |        |        |              |                     |

Menurut kelima penguji, hampir semua penguji SPBRM sudah setuju dengan menggunakan SPBRM membantu dalam mengingat setiap riwayat mahasiswa. Pengguna dapat mengingat riwayat mahasiswa dengan mudah tanpa perlu mengkhawatirkan seberapa banyak interaksi yang harus dilakukan dan diingat dengan mahasiswa. Terdapat dua orang penguji yang setuju dengan kemudahan dalam mengoperasikan SPBRM. Selain penguji tersebut menyatakan netral dan ada seorang penguji menyatakan tidak setuju. Maka dari itu, untuk pengembangan perangkat lunak harus dikaji ulang pada bagian antarmuka, sehingga semua pengguna dapat setuju dengan kemudahan dalam mengoperasikan SPBRM.

Setelah menggunakan SPBRM, kebanyakan pengguna merasa dibantu untuk meningkatkan pemahaman dalam mengelola riwayat mahasiswa. Kemudian pengguna merasa lebih efektif jika bekerja dengan SPBRM dalam mengelola riwayat mahasiswa. Pengguna juga merasa lebih efisien jika bekerja dengan SPBRM dalam mengelola riwayat mahasiswa. Namun untuk pengembangan



---

perangkat lunak akan ditingkatkan aspek efisien yang dimiliki SPBRM, karena pengguna masih ragu dengan efisiensi yang dimiliki SPBRM.



## 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 Perekam dan Berbagi Riwayat Mahasiswa (SPBRM) anatara lain :

- (1) Autentikasi dan otorisasi akun dosen UNPAR dapat menggunakan Google OAuth. Perangkat lunak menggunakan Google OAuth pada fungsi login.
- (2) Format penulisan yang seragam dapat dibuat dengan menggunakan Markdown. Perangkat lunak menggunakan Markdown untuk menuliskan riwayat mahasiswa pada fitur mengedit info mahasiswa dan fitur membuat entri baru.
- (3) Menampilkan teks dengan format Markdown ke HTML dapat menggunakan StrapdownJS. Perangkat lunak menggunakan StrapdownJS untuk menampilkan riwayat mahasiswa yang terdapat fitur melihat info mahasiswa dan fitur melihat versi ini pada histori.
- (4) Penggunaan Zurb Foundation berhasil membuat tampilan antarmuka perangkat lunak Sistem Perekam dan Berbagi Riwayat Mahasiswa (SPBRM) menjadi responsif.
- (5) Pada tahap pengujian dapat disimpulkan bahwa perangkat lunak Sistem Perekam dan Berbagi Riwayat Mahasiswa (SPBRM) sudah berjalan dengan baik dan memberikan keluaran sesuai yang diharapkan pengguna.

#### 6.2 Saran

Saran yang dapat diberikan untuk perbaikan dan pengembangan Sistem Perekam dan Berbagi Riwayat Mahasiswa (SPBRM) antara lain :

- (1) Tampilan antarmuka masih kurang menarik terutama untuk tata letak tombol 'Simpan' pada edit.php dan new.php, untuk pengembangan dapat menggunakan navigasi dan *plugins* yang dimiliki Zurb Foundation.

- (2) Pada fungsi membuat entri baru, untuk pengembangan data npm terhubung dengan data nama. Jadi cukup dengan memasukkan npm seorang mahasiswa, lalu pada bagian nama otomatis mengeluarkan nama mahasiswa dengan npm tersebut.
- (3) Pada fungsi histori, untuk pengembangan dapat ditambahkan keterangan yang membandingkan versi baru dan versi lama. Jadi pengguna mengetahui bagian mana yang dihapus dan bagian mana yang ditambah atau dirubah.

## DAFTAR REFERENSI

- [1] Google, Inc, *Google Identity Platform*. <https://developers.google.com/accounts/docs/OAuth2>, 2013.
- [2] Google, Inc, *Google+ Platform*. <https://developers.google.com/+api/oauth>, 2013.
- [3] Google, Inc, *API Client Library for PHP*. [https://developers.google.com/api-client-library/php/start/get\\_started](https://developers.google.com/api-client-library/php/start/get_started), 2015.
- [4] A. Kumar, *Markdown Guide*. Self-published, 2015.
- [5] GitHub, Inc, *GitHub Flavored Markdown*. <https://help.github.com/articles/github-flavored-markdown/>, 2015.
- [6] A. Adib, *Strapdown.js - Instant and elegant Markdown documents*. <http://strapdownjs.com/>, 2014.
- [7] ZURB, Inc, *Foundation Documentation*. <http://foundation.zurb.com/docs/>, 2015.
- [8] 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.



# 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>SPBRM | 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 SPBRM <h4>(Sistem Perekam dan Berbagi 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 $plus = new Google_Service_Oauth2($client);
26
27 if (isset($_REQUEST['logout']))
28 {
29 unset($_SESSION['access_token']);
30 header('Location: https://www.google.com/accounts/Logout?continue=https://appengine.
31 google.com/_ah/logout?continue=http:// ' . $_SERVER['HTTP_HOST'] . $_SERVER['
32 PHP_SELF']);
33 }
34
35 if (isset($_GET['code']))
36 {
37 $client->authenticate($_GET['code']);
38 $_SESSION['access_token'] = $client->getAccessToken();
39 header('Location: http:// ' . $_SERVER['HTTP_HOST'] . $_SERVER['PHP_SELF']);
40 }
41
42 if (isset($_SESSION['access_token']))
43 {
44 $client->setAccessToken($_SESSION['access_token']);
45 }
46
47 if ($client->getAccessToken())
48 {
49 }
50 else
51 {
52 $authUrl = $client->createAuthUrl();
53 }
54
55 if (isset($authUrl))
56 {
57 echo "Login_with_Google";
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>SPBRM | 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($status == "valid_email")
54 {
55 header("Location: _list.php");
56 exit;
57 }
58 else
59 {
60 echo "<script>alert('Email_yang_digunakan_tidak_dapat_mengakses_SPBRM._Email_yang_dapat_
61 digunakan_untuk_mengakses_SPBRM_adalah_email_yang_diakhiri_@unpar.ac.id. ');window.
62 location.href='index.php?logout';</script>";
63 exit;
64 }
65 <?php is_valid_email($email); ?>
66 </body>
67 </html>

```

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>SPBRM | 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 Add
19 Logout
20
21 <hr/>
22 </div>
23
24 <div class="row">
25 <?php
26 include_once "configDatabase.php";
27
28 if(! $id_mysql)
29 {
30 die("Database_tidak_bisa_dibuka");
31 }

```



```

32| if (! mysql_select_db("spbrm", $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> . $row['npm'] . "</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>SPBRM | 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'];>.</h3>
18| <ul class="button-group">
19| <a href="edit.php?npm=<?php echo $npm?>" class="button_secondary">Edit
20| <a href="history.php?npm=<?php echo $npm?>" class="button_secondary">Lihat Histori
21|
22| Menu Utama
23| Logout
24|
25| </div>
26| <?php
27| include_once "configDatabase.php";
28|
29| if (! $id_mysql)
30| {
31| die("Database_tidak_bisa_dibuka");
32| }
33|
34| if (! mysql_select_db("spbrm", $id_mysql))
35| {
36| die("Database_tidak_bisa_dipilih");
37| }
38|
39| $lihat = "INSERT INTO histori (npm, pengguna, status, tanggal_pembaruan, keterangan) VALUES (
40| ". mysql_real_escape_string($npm) . "', '$_SESSION['email']', '$_SESSION['melihat']', now(), '')";
41|
42| if (mysql_query($lihat) === TRUE)
43| {
44| }
45| else
46| {
47| echo "Error: . $lihat . "
 . $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| {
54| echo "NPM: . $row['npm']; echo "
";
55| echo "Nama: . $row['nama']; echo "
";
56| }
57| </div>
58| </body>
59| </html>

```

```

59|<?php
60|echo $row['keterangan'];
61|?>
62|</xmp>
63|
64| <?php
65| }
66| ?>
67| <script src="js/0.2/strapdown.js"></script>
68|</body>
69|</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>SPBRM | 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("spbrm", $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=now()_
46| WHERE_npm='$npm'";
47| $sql2 = "INSERT INTO_histori_(npm,_pengguna,_status,_tanggal_pembaruan,_keterangan).VALUES('
48| . mysql_real_escape_string($npm) . ','._'. $ _SESSION['email'] . ','._'mengedit' ,_now() ,_'.
49| 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 . "
" . $id_mysql->error;
58| echo "Error:_ " . $sql2 . "
" . $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']?>.

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| <a href="view.php?npm=<?php echo $npm?>" class="button_secondary">Kembali
73| <input class="button" type="submit" name="submit" value="Simpan">
74| Menu Utama
75| Logout
76|
77| <hr>
78| </div>
79| <div class="row">
80| <div class="small-12_columns">
81| <textarea style="height:_300px;" placeholder="<?php echo $cariketerangan; ?>" name="
82| keteranganbaru"><?php echo $cariketerangan; ?></textarea>
83| </div>
84| </div>

```

```

81 <div class="row">
82 <p>Format penulisan menggunakan Markdown, untuk sintaks Markdown dapat dilihat <a href='
https://help.github.com/articles/github-flavored-markdown/' target="_blank">disini .</p>
83 </div>
84 </form>
85 <?php
86 }
87 ?>
88 </body>
89 </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>SPBRM | 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("spbrm", $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 <a href="view.php?npm=<?php echo $npm?>" class="button_secondary">Kembali
43 Menu Utama
44 Logout
45
46 </div>
47 <div class="row">
48 <h3>NPM <?php echo $carinpm; ?> Nama <?php echo $carinama; ?>
49 <hr/>
50 <ul class="disc">
51 <?php
52 $hasil = mysql_query("SELECT_*_FROM_histori_WHERE_npm='$npm'_ORDER_BY_id_histori_DESC"
53 , $id_mysql);
54
55 if (! $hasil)
56 {
57 die("Permintaan_gagal");
58 }
59
60 while($row = mysql_fetch_array($hasil))
61 {
62 echo "" . $row['tanggal_pembaruan'] . " " . $row['pengguna'] . " " . $row['
status'] . " " . $row['npm'] . " " . ($row['keterangan'] != "" ? '<a href="
past.php?id=' . $row['id_histori'] . '">[lihat_versi_ini]' : "") . "";
63 }
64 ?>
65
66 <h3>
67 </div>
68 </body>
69 </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>SPBRM | 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 Kembali
19 Logout
20
21 <hr/>
22 <?php
23 include_once "configDatabase.php";
24
25 if(! $id_mysql)
26 {
27 die("Database_tidak_bisa_dibuka");
28 }
29
30 if(! mysql_select_db("spbrm", $id_mysql))
31 {
32 die("Database_tidak_bisa_dipilih");
33 }
34
35 $cari = mysql_query("SELECT_keterangan_FROM_histori_WHERE_id_histori='$id'", $id_mysql);
36 while($row = mysql_fetch_array($cari))
37 {
38 ?>
39 </div>
40 </div>
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>SPBRM | 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("spbrm", $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 $found = mysql_query($cek) or die(mysql_error());
36 if(mysql_num_rows($found)>0)
37 {
38 echo "<script>alert('Data_telah_terdftar..Silahkan_diulangi_dengan_data_yang_lain.');"
39 . window.location.href='new.php';</script>";
40 }
41 else
42 {
43 $sql1 = "INSERT INTO info_mahasiswa (npm, nama, keterangan) VALUES ('"
44 . mysql_real_escape_string($npm) . "', '"
45 . mysql_real_escape_string($nama) . "', '"
46 . mysql_real_escape_string($keterangan) . "')";
47
48 $sql2 = "INSERT INTO histori (npm, pengguna, status, tanggal_pembaruan, keterangan) VALUES
49 ('" . mysql_real_escape_string($npm) . "', '"
50 . $_SESSION['email'] . "', '"
51 . $_SESSION['membuat_entri'] . "', '"
52 . mysql_real_escape_string($keterangan) . "')";

```

```

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 . "
" . $id_mysql->error;
52| echo "Error:_" . $sql2 . "
" . $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| Kembali
66| <input class="button" type="submit" name="submit" value="Simpan">
67| Menu Utama
68| Logout
69|
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|
95| Isilah deskripsi umum mahasiswa disini.
96|
97| # Catatan
98|
99| * 8 Juni 2015, pertama kali dibuat
100| </textarea>
101| </div>
102| </div>
103| <div class="row">
104| <p>Format penulisan menggunakan Markdown, untuk sintaks Markdown dapat dilihat <a href
105| = 'https://help.github.com/articles/github-flavored-markdown/' target="_blank">
106| disini.</p>
107| </div>
108| </form>
109| <?php
110| }
111| ?>
</body>
</html>

```

Listing A.9: client.php

```

1| <?php
2| $client = new Google_Client();
3| $client->setClientId('568951368854-ufmbistn0pcaq0khubafola133orfge.apps.googleusercontent.com');
4| $client->setClientSecret('cSZ-AUmeQ9PaWWry_IpiBBi');
5| $client->setRedirectUri('http://localhost/oauth.php');
6| $client->setDeveloperKey('AIzaSyDRoDJAzUR_TsNUNRUeTYsBb7dFBQKZy7M');
7| ?>

```

Listing A.10: configDatabase.php

```

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

```



# LAMPIRAN B

## KODE PROGRAM UNTUK PENGUJIAN FUNGSIONAL

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>SPBRM | 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 SPBRM <h4>(Sistem Perekam dan Berbagi 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 $plus = new Google_Service_Oauth2($client);
26
27 if (isset($_REQUEST['logout']))
28 {
29 unset($_SESSION['access_token']);
30 header('Location:_https://www.google.com/accounts/Logout?continue=https://appengine.
31 google.com/_ah/logout?continue=http:// ' . $_SERVER['HTTP_HOST'] . $_SERVER['
32 PHP_SELF']);
33 }
34
35 if (isset($_GET['code']))
36 {
37 $client->authenticate($_GET['code']);
38 $_SESSION['access_token'] = $client->getAccessToken();
39 header('Location:_http:// ' . $_SERVER['HTTP_HOST'] . $_SERVER['PHP_SELF']);
40 }
41
42 if (isset($_SESSION['access_token']))
43 {
44 $client->setAccessToken($_SESSION['access_token']);
45 }
46
47 if ($client->getAccessToken())
48 {
49 }
50 else
51 {
52 $authUrl = $client->createAuthUrl();
53 }
54
55 if (isset($authUrl))
56 {
57 echo "Login_with_Google";
58 }
59 </div>
60 </body>
61 </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>SPBRM | 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_SPBRM._Email_yang_dapat_digunakan_untuk_mengakses_SPBRM_adalah_email_yang_diakhiri_@unpar.ac.id_atau_@student.unpar.ac.id.');

```

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>SPBRM | 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 Add
19 Logout
20
21 <hr/>
22 </div>
23
24 <div class="row">
25 <?php
26 include_once "configDatabase.php";
27
28 if(! $id_mysql)
29 {
30 die("Database_tidak_bisa_dibuka");
31 }

```



```

32| if (! mysql_select_db("spbrm", $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> . $row['npm'] . "</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>SPBRM | 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'];>.</h3>
18| <ul class="button-group">
19| <a href="edit.php?npm=<?php echo $npm?>" class="button_secondary">Edit
20| <a href="history.php?npm=<?php echo $npm?>" class="button_secondary">Lihat Histori
21|
22| Menu Utama
23| Logout
24|
25| </div>
26| <?php
27| include_once "configDatabase.php";
28|
29| if (! $id_mysql)
30| {
31| die("Database_tidak_bisa_dibuka");
32| }
33|
34| if (! mysql_select_db("spbrm", $id_mysql))
35| {
36| die("Database_tidak_bisa_dipilih");
37| }
38|
39| $lihat = "INSERT INTO _histori_(npm, _pengguna, _status, _tanggal_pembaruan, _keterangan) VALUES (
40| ' . mysql_real_escape_string($npm) . ', ' . $_SESSION['email'] . ', ' . _melihat, _now(), ')";
41|
42| if (mysql_query($lihat) == TRUE)
43| {
44| }
45| else
46| {
47| echo "Error: " . $lihat . "
" . $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| {
54| echo "NPM: " . $row['npm']; echo "
";
55| echo "Nama: " . $row['nama']; echo "
";
56| }
57| </div>
58| <?div class="row">
59| <div class="small-11_small-centered_columns">
60| <h3>Anda melihat catatan mahasiswa ini sebagai <?php echo $_SESSION['email'];>.</h3>
61| <ul class="button-group">
62| <a href="edit.php?npm=<?php echo $npm?>" class="button_secondary">Edit
63| <a href="history.php?npm=<?php echo $npm?>" class="button_secondary">Lihat Histori
64|
65| Menu Utama
66| Logout
67|
68| </div>
69| </div>
70| </body>
71| </html>

```

```

59|<?php
60|echo $row['keterangan'];
61|?>
62|</xmp>
63|
64| <?php
65| }
66| ?>
67| <script src="js/0.2/strapdown.js"></script>
68|</body>
69|</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>SPBRM | 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("spbrm", $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=now()_
46| WHERE_npm='$npm'";
47| $sql2 = "INSERT INTO_histori_(npm,_pengguna,_status,_tanggal_pembaruan,_keterangan).VALUES(' '
48| . mysql_real_escape_string($npm) . "','_".$SESSION['email']. "','_mengedit',_now(),_ ' ' .
49| mysql_real_escape_string($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 . "
" . $id_mysql->error;
58| echo "Error:_ " . $sql2 . "
" . $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']?>.

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| <a href="view.php?npm=<?php echo $npm?>" class="button_secondary">Kembali
73| <input class="button" type="submit" name="submit" value="Simpan">
74| Menu Utama
75| Logout
76|
77| <hr>
78| </div>
79| <div class="row">
80| <div class="small-12_columns">
81| <textarea style="height:_300px;" placeholder="<?php echo $cariketerangan; ?>" name="
82| keteranganbaru"><?php echo $cariketerangan; ?></textarea>
83| </div>
84| </div>
85| </div>

```

```

81 <div class="row">
82 <p>Format penulisan menggunakan Markdown, untuk sintaks Markdown dapat dilihat <a href='
https://help.github.com/articles/github-flavored-markdown/' target="_blank">disini .</p>
83 </div>
84 </form>
85 <?php
86 }
87 ?>
88 </body>
89 </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>SPBRM | 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("spbrm", $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 <a href="view.php?npm=<?php echo $npm?>" class="button_secondary">Kembali
43 Menu Utama
44 Logout
45
46 </div>
47 <div class="row">
48 <h3>NPM <?php echo $carinpm; ?> Nama <?php echo $carinama; ?>
49 <hr/>
50 <ul class="disc">
51 <?php
52 $hasil = mysql_query("SELECT_*_FROM_histori_WHERE_npm='$npm'_ORDER_BY_id_histori_DESC"
53 , $id_mysql);
54
55 if (! $hasil)
56 {
57 die("Permintaan_gagal");
58 }
59
60 while($row = mysql_fetch_array($hasil))
61 {
62 <div>
63 <div>
64 <div>
65 <div>
66 <div>
67 <div>
68 <div>
69

```

### 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>SPBRM | 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 Kembali
19 Logout
20
21 <hr/>
22 <?php
23 include_once "configDatabase.php";
24
25 if(! $id_mysql)
26 {
27 die("Database_tidak_bisa_dibuka");
28 }
29
30 if(! mysql_select_db("spbrm", $id_mysql))
31 {
32 die("Database_tidak_bisa_dipilih");
33 }
34
35 $cari = mysql_query("SELECT keterangan FROM histori WHERE id_histori='$id'", $id_mysql);
36 while($row = mysql_fetch_array($cari))
37 {
38 ?>
39 </div>
40 </div>
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 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>SPBRM | 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("spbrm", $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 $found = mysql_query($cek) or die(mysql_error());
36 if(mysql_num_rows($found)>0)
37 {
38 echo "<script>alert('Data telah terdaftar. Silahkan diulangi dengan data yang lain.');"
39 . window.location.href='new.php';</script>";
40 }
41 else
42 {
43 $sql1 = "INSERT INTO info_mahasiswa (npm, nama, keterangan) VALUES ('" .
44 mysql_real_escape_string($npm) . "', '" . mysql_real_escape_string($nama) . "', '" .
45 mysql_real_escape_string($keterangan) . "')";
46
47 $sql2 = "INSERT INTO histori (npm, pengguna, status, tanggal_pembaruan, keterangan) VALUES
48 ('" . mysql_real_escape_string($npm) . "', '" . $_SESSION['email'] . "', '" . 'membuat entri',
49 now(), '" . mysql_real_escape_string($keterangan) . "')";

```

```

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 . "
" . $id_mysql->error;
52| echo "Error:_" . $sql2 . "
" . $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| Kembali
66| <input class="button" type="submit" name="submit" value="Simpan">
67| Menu Utama
68| Logout
69|
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|
95| Isilah deskripsi umum mahasiswa disini.
96|
97| # Catatan
98|
99| * 8 Juni 2015, pertama kali dibuat
100| </textarea>
101| </div>
102| </div>
103| <div class="row">
104| <p>Format penulisan menggunakan Markdown, untuk sintaks Markdown dapat dilihat <a href
105| = 'https://help.github.com/articles/github-flavored-markdown/' target="_blank">
106| disini.</p>
107| </div>
108| </form>
109| <?php
110| }
111| ?>
</body>
</html>

```

Listing B.9: client.php

```

1| <?php
2| $client = new Google_Client();
3| $client->setClientId('568951368854-ufmbistn0pcaq0khubafola133orfge.apps.googleusercontent.com');
4| $client->setClientSecret('cSZ-AUmeQ9PaWWry_IpiBBi');
5| $client->setRedirectUri('http://localhost/oauth.php');
6| $client->setDeveloperKey('AIzaSyDRoDJAzUR_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

### WAWANCARA

Hasil wawancara dengan beberapa dosen teknik informatika di fakultas teknik informasi dan sains.

1. Berikut hasil wawancara dengan seorang dosen Teknik Informatika bernama Pascal Alfadian pada tanggal 16 Juni 2015.

Pertanyaan : Bapak memiliki berapa anak wali?

Jawaban : 34

Pertanyaan : Kesulitan apa yang sering dialami dalam menjadi dosen wali?

Jawaban : Masalah administrasi seperti tanda tangan pada saat perwalian dan mahasiswa tidak memiliki sangsi administrasi jika tidak datang perwalian.

Pertanyaan : Apakah bapak sudah merasa optimal pada saat melakukan proses perwalian?

Jawaban : Masih bisa diperbaiki, selama ini saya melakukan survei ke 2-3 dosen untuk mengetahui informasi seorang mahasiswa.

Pertanyaan : Menurut bapak/ibu apa yang dapat membuat proses perwalian menjadi optimal?

Jawaban : Mendatangi dosen baik dosen yang memiliki relasi yang dekat dengan saya maupun yang tidak dekat dan keterbatasan waktu untuk membicarakan hal tersebut.

Pertanyaan : Apakah bapak merasa perlu untuk dibuatkan sebuah perangkat lunak yang membantu dalam pencatatan riwayat mahasiswa?

Jawaban : Ide yang menarik.

Pertanyaan : Fitur apa yang diharapkan? Jawaban : Fitur yang mendukung agar saya menjadi efisien dalam menjadi dosen wali.

2. Berikut hasil wawancara dengan seorang dosen Teknik Informatika bernama Gede Karya pada tanggal 16 Juni 2015.

Pertanyaan : Bapak memiliki berapa anak wali?

Jawaban : 38

Pertanyaan : Kesulitan apa yang sering dialami dalam menjadi dosen wali?

Jawaban : Tidak ingat semua riwayat mahasiswa.

Pertanyaan : Apakah bapak sudah merasa optimal pada saat melakukan proses perwalian?

Jawaban : Ya, sudah tercapai.

Pertanyaan : Menurut bapak/ibu apa yang dapat membuat proses perwalian menjadi optimal?

Jawaban : Jadwal yang baik.

Pertanyaan : Apakah bapak merasa perlu untuk dibuatkan sebuah perangkat lunak yang membantu dalam pencatatan riwayat mahasiswa?

Jawaban : Ya, ide yang baik.

Pertanyaan : Fitur apa yang diharapkan?

Jawaban : Kebebasan dalam mencatat apa pun kemudian komentar positif dan komentar negatif.

3. Berikut hasil wawancara dengan seorang dosen Teknik Informatika bernama Cecilia Esti Nugraheni pada tanggal 16 Juni 2015.

Pertanyaan : Ibu memiliki berapa anak wali?

Jawaban : 30-40

Pertanyaan : Kesulitan apa yang sering dialami dalam menjadi dosen wali?

Jawaban : Tidak ada kesulitan untuk anak wali yang sejak awal saya pegang, namun ada kesulitan untuk mengenal mahasiswa yang mengambil topik skripsi saya, dan yang terakhir jika ada persoalan namun tidak cerita.

Pertanyaan : Apakah ibu sudah merasa optimal pada saat melakukan proses perwalian?

Jawaban : Belum, karena terbatas oleh waktu.

Pertanyaan : Menurut ibu apa yang dapat membuat proses perwalian menjadi optimal?

Jawaban : Menyediakan waktu khusus.

Pertanyaan : Apakah ibu merasa perlu untuk dibuatkan sebuah perangkat lunak yang membantu dalam pencatatan riwayat mahasiswa?

Jawaban : Saya setuju karena akan tertolong.

Pertanyaan : Fitur apa yang diharapkan?

Jawaban : Bisa bebas mencatat apa pun dan catatan khusus dosen wali.

4. Berikut hasil wawancara dengan seorang dosen Teknik Informatika bernama Joanna Helga pada tanggal 16 Juni 2015.

Pertanyaan : Ibu memiliki berapa anak wali?

Jawaban : 21 dari angkatan 2014 dan 5/6 mahasiswa yang sedang bimbingan.

Pertanyaan : Kesulitan apa yang sering dialami dalam menjadi dosen wali?

Jawaban : Ada mahasiswa yang tidak datang pada saat perwalian sehingga timbul pertanyaan mahasiswa tersebut akan datang atau tidak, dan apa perlu diingatkan tidak. Jadi saya perlu bertanya ke rekan dosen untuk mengetahui kondisi mahasiswa tersebut dikelas.

Pertanyaan : Apakah ibu sudah merasa optimal pada saat melakukan proses perwalian?

Jawaban : Tidak terlalu, karena sistem unpar aksesnya lama untuk mengetahui nilai mahasiswa kemudian anak wali malas cerita sehingga saya harus menkonfirmasi dengan dosen lain.



Pertanyaan : Menurut bapak/ibu apa yang dapat membuat proses perwalian menjadi optimal?

Jawaban : Kalau anak wali tidak memiliki masalah dan lebih mudah dalam mengakses permasalahan yang dimiliki seorang anak wali. Misalnya masalah keuangan dan kesulitan dalam kuliah programming.

Pertanyaan : Apakah ibu merasa perlu untuk dibuatkan sebuah perangkat lunak yang membantu dalam pencatatan riwayat mahasiswa?

Jawaban : Ya setuju.

Pertanyaan : Fitur apa yang diharapkan?

Jawaban : Catatan umum seperti telat kelas, sering tidak masuk kelas, dan sering tidak mengumpulkan tugas bukan ditulis supaya bisa direkap agar lebih kelihatan. Kemudian mudah dalam pengaksesan.

5. Berikut hasil wawancara dengan seorang dosen Teknik Informatika bernama Chandra Wijaya pada tanggal 17 Juni 2015.

Pertanyaan : Bapak memiliki berapa anak wali?

Jawaban : 40-50

Pertanyaan : Kesulitan apa yang sering dialami dalam menjadi dosen wali?

Jawaban : Kadang harus disuruh-suruh untuk perwalian bahkan tidak datang.

Pertanyaan : Apakah bapak sudah merasa optimal pada saat melakukan proses perwalian?

Jawaban : Belum, karena harus mengingat *track record* setiap mahasiswa.

Pertanyaan : Menurut bapak apa yang dapat membuat proses perwalian menjadi optimal?

Jawaban : Setiap mahasiswa memiliki catatan, agar dapat dilihat memiliki masalah apa dan akibatnya apa.

Pertanyaan : Apakah bapak merasa perlu untuk dibuatkan sebuah perangkat lunak yang membantu dalam pencatatan riwayat mahasiswa?

Jawaban : Merasa sangat terbantu, apa lagi kalau dijaga oleh setiap dosen wali.

Pertanyaan : Fitur apa yang diharapkan?

Jawaban : Fitur yang ditawarkan sudah cukup.

6. Berikut hasil wawancara dengan seorang dosen Teknik Informatika bernama Veronica Sri Moertini pada tanggal 17 Juni 2015.

Pertanyaan : Ibu memiliki berapa anak wali?

Jawaban : 30

Pertanyaan : Kesulitan apa yang sering dialami dalam menjadi dosen wali?

Jawaban : Sulit menghubungi mahasiswa, kemudian pada saat perwalian adanya menitipkan pada mahasiswa lain atau menghubungi melalui email atau whatsapp sehingga kurang intensif.

Pertanyaan : Apakah ibu sudah merasa optimal pada saat melakukan proses perwalian?

Jawaban : Sudah dengan adanya SIA, namun sisi diluar akademis tidak tercantum pada SIA seperti asal sekolah, bagaimana kondisi keluarga, dan tidak ada data diri.

Pertanyaan : Menurut ibu apa yang dapat membuat proses perwalian menjadi optimal?

Jawaban : Kebutuhan *cross check* dengan dosen lain misalnya kenapa nilai mahasiswa ini nilainya segini. Kemudian sulit untuk mengakses masalah yang dimiliki mahasiswa.

Pertanyaan : Apakah ibu merasa perlu untuk dibuatkan sebuah perangkat lunak yang membantu dalam pencatatan riwayat mahasiswa?

Jawaban : Setuju.

Pertanyaan : Fitur apa yang diharapkan?

Jawaban : Form mengakses masalah mahasiswa dan form kehadiran kelas yang mungkin dibutuhkan akses dari tata usaha atau asisten dosen.

## LAMPIRAN D

### 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 E

### 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. |         |   |   |    |     |