



TUGAS AKHIR - KS 141501

RANCANG BANGUN APLIKASI OTENTIKASI MEDIA SOSIAL DENGAN MIKROTIK ROUTEROS API UNTUK HOTSPOT

DEVELOPMENT OF SOCIAL MEDIA AUTHENTICATION APPLICATION USING MIKROTIK ROUTEROS API FOR HOTSPOT

RYSMA ADITYA WIDJAJA
NRP 052114 4000 0117

Dosen Pembimbing
Bekti Cahyo Hidayanto, S.Si, M.Kom

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KS 141501

RANCANG BANGUN APLIKASI OTENTIKASI MEDIA SOSIAL DENGAN MIKROTIK ROUTEROS API UNTUK HOTSPOT

RYSMA ADITYA WIDJAJA
NRP 052114 4000 0117

Dosen Pembimbing
Bekti Cahyo Hidayanto, S.Si, M.Kom

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KS 141501

*DEVELOPMENT OF SOCIAL MEDIA
AUTHENTICATION APPLICATION USING
MIKROTIK ROUTEROS API FOR HOTSPOT*

RYSMA ADITYA WIDJAJA

NRP 052114 4000 0117

Supervisors

Bekti Cahyo Hidayanto, S.Si, M.Kom

INFORMATION SYSTEMS DEPARTMENT

Information Technology and Communication Faculty

Sepuluh Nopember Institut of Technology

Surabaya 2018

LEMBAR PENGESAHAN

**RANCANG BANGUN APLIKASI OTENTIKASI MEDIA
SOSIAL DENGAN MIKROTIK ROUTEROS API UNTUK
HOTSPOT**

TUGAS AKHIR

Disusun untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Departemen Sistem Informasi
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:

RYSMA ADITYA WIDJAJA
NRP 05211440000117

Surabaya, Juli 2018

**KEPALA
DEPARTEMEN SISTEM INFORMASI**



Dr. Ir. Aris Pjahyanto, M.Kom.
NIP 19650310 199102 1 001

LEMBAR PERSETUJUAN

RANCANG BANGUN APLIKASI OTENTIKASI MEDIA SOSIAL DENGAN MIKROTIK ROUTEROS API UNTUK HOTSPOT

TUGAS AKHIR

Disusun untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Departemen Sistem Informasi
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:

RYSMA ADITYA WIDJAJA

NRP 05211440000117

Disetujui Tim Penguji: Tanggal Ujian : 6 Juli 2018

Periode Wisuda : September 2018

Bekti Cahyo Hidayanto, S.Si, M.Kom

(Pembimbing I)

Dr.Eng. Febriliyan Samopa, S.Kom., M.Kom

(Penguji I)

Nisfu Asrul Sani S.Kom., M.Sc

(Penguji II)

RANCANG BANGUN APLIKASI OTENTIKASI MEDIA SOSIAL DENGAN MIKROTIK ROUTEROS API UNTUK HOTSPOT

Nama Mahasiswa : Rysma Aditya Widjaja
NRP : 05211440000117
Departemen : Sistem Informasi FTIK – ITS
Dosen Pembimbing 1 : Bakti Cahyo Hidayanto, S.Si, M.Kom

ABSTRAK

Penggunaan hotspot yang kian tinggi, dikutip dari Peta interaktif iPass 2014, yang menunjukkan data yang diberikan oleh analis Maravedis Rethink, menunjukkan bahwa pada bulan Desember 2014 terdapat 46.000.000 hotspot di seluruh dunia dan lebih dari 22.000.000 hotspot. Tingginya penggunaan hotspot global ini salah satunya disebabkan karena kebutuhan akses internet yang tinggi pula. Tidak dapat dipungkiri, Indonesia termasuk didalam enam negara dengan penggunaan internet yang tinggi dengan total 123 juta orang di survey tahun 2018 yang menduduki peringkat keenam. Inilah yang menyebabkan pula makin banyaknya tempat – tempat yang menyediakan hotspot, karena internet dinilai sebagai suatu kebutuhan yang baru. Hotspot yang sering dijumpai, biasanya tersedia di café, mall, hotel, dan public area lain rata – rata masih menggunakan kata sandi untuk mendapatkan akses internet gratis, sehingga value yang didapat oleh penyedia hotspot dirasa masih kurang, mereka tidak bisa memantau siapa saja yang menggunakan layanan hotspot mereka. Melihat tren digital di era global, social media

merupakan sebuah tools yang powerful, dan Indonesia merupakan salah satu negara yang memiliki tingkat pengguna sosial media yang tinggi. Indonesia saat ini mencapai 63 juta orang. Dari angka tersebut, 95 persennya menggunakan internet untuk mengakses jejaring sosial. Melihat data pengguna media sosial dan hotspot yang terus meningkat, ada inovasi lain yang dapat diterapkan oleh penyedia layanan hotspot agar memanfaatkan sosial media sebagai channel marketing melalui hotspot yang mereka sediakan kepada pengunjung. Pengunjung yang hendak menggunakan free hotspot yang disediakan, harus terotentikasi via social media agar mereka dapat menikmati layanan internet, hal ini juga bisa disebut dengan hotspot social authentication, dimana semua pengguna hotspot akan terdaftar kedalam sistem, penyedia hotspot memiliki data social media untuk kebutuhan CRM (customer relationship management) dan juga bisa melakukan tracking pengguna harian, bulanan, tahunan, sehingga ada value lebih yang dirasakan, dan juga pengguna tidak perlu lagi menanyakan kata sandi wifi karena bisa melakukan otentikasi melalui social media. Platform ini memanfaatkan Mikrotik routerboard, RouterOS API, media sosial, yang diintegrasikan dengan perangkat lunak yang ditanam di dalam web server.

Kata Kunci: Mikrotik, RouterOS API, Hotspot Social Authentication

**DEVELOPMENT OF SOCIAL MEDIA
AUTHENTICATION APPLICATION USING
MIKROTIK *ROUTEROS* API FOR HOTSPOT**

Student Name : Rysma Aditya Widjaja
NRP : 05211440000117
Departement : Sistem Informasi FTIK – ITS
Supervisor 1 : Bekti Cahyo Hidayanto, S.Si, M.Kom

ABSTRACT

The increasingly high use of hotspots, quoted from iPass 2014 interactive map, which shows data provided by analyst Maravedis Rethink, shows that by December 2014 there were 46 million hotspots worldwide and over 22,000,000 hotspots. The high use of this global hotspot is one of them due to the need for high internet access as well. It is undeniable that Indonesia is included in six countries with high internet usage with a total of 123 million people in the 2018 survey which ranked sixth. This is what causes more and more places that provide hotspots, because the internet is considered as a new requirement. Hotspots are often found in cafes, malls, hotels, and other public areas still use average passwords to get free internet access, so the value obtained by hotspot providers is still lacking, they can not monitor anyone who uses their hotspot service. Looking at digital trends in the global era, social media is a powerful tool, and Indonesia is one country that has a high level of social media users. Indonesia currently reaches 63 million people. Of that number, 95 percent use the internet to access social networking. Looking at the ever-increasing data of social media users and hotspots, there are other innovations that hotspot service providers can take advantage of social

media as a marketing channel through the hotspots they provide to visitors. Visitors who want to use the free hotspot provided must be authenticated via social media so they can enjoy internet service, this can also be called hotspot social authentication, where all hotspot users will be registered into the system, hotspot providers have social media data for CRM (customer relationship management) and can also track daily users, monthly, yearly, so there is more value perceived, and also penggua no longer need to ask for a wifi password because it can authenticate through social media. The platform utilizes Mikrotik routerboard, RouterOS API, social media, which is integrated with software embedded within the web server.

Keywords: Mikrotik, RouterOS API, Hotspot Social Authentication

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT yang telah melimpahkan rahmat, taufik serta hidayah-Nya sehingga penulis dapat menyelesaikan buku Tugas Akhir dengan tepat waktu yang berjudul “RANCANG BANGUN APLIKASI OTENTIKASI MEDIA SOSIAL DENGAN MIKROTIK *ROUTEROS* API UNTUK HOTSPOT”

Pada kesempatan kali ini, penulis ingin menyampaikan ucapan terima kasih kepada semua pihak yang telah mendukung, mengarahkan, membimbing, membantu, dan memberikan semangat kepada penulis, yaitu antara lain kepada:

1. Allah SWT yang telah melimpahkan rahmat dan karunianya kepada penulis sehingga dapat menyelesaikan buku Tugas Akhir dengan tepat waktu.
2. Bapak Bakti Cahyo Hidayanto selaku dosen pembimbing penulis yang telah meluangkan waktu dan pikiran untuk membimbing penulis untuk menuju hasil yang terbaik dan memuaskan dalam menyelesaikan buku Tugas Akhir ini.
3. Ibu Mahendrawathi selaku dosen wali penulis yang telah mendukung dan mengarahkan penulis selama masa perkuliahan dan penelitian Tugas Akhir ini.
4. Orang tua, yaitu Bapak Mayor Adi Wijaya, Ibu Sri Umiati, dan Adik Ailsa Irma Adiani Wijaya, dan adik yang lain, yaitu Marani Bafianti Suwito serta orang terdekat penulis yang selalu mendoakan dan memberikan semangat dan kasih sayang dalam segala proses penyelesaian buku Tugas Akhir ini.
5. Mas Nanok selaku admin laboratorium IKTI (Infrastruktur dan Keamanan Sistem Informasi) yang telah membantu dan mengarahkan penulis dalam segala hal penyelesaian administrasi untuk Tugas Akhir ini.

6. Semua bapak dan ibu dosen dan karyawan tata usaha jurusan Sistem Informasi yang telah memberikan ilmu pengetahuan kepada penulis selama masa studi empat tahun ini.
7. Karyawan DPTSI selaku responden kuesioner yang telah meluangkan waktu untuk mengisi kuesioner dan membantu menyelesaikan Tugas Akhir ini dengan cepat.
8. Rama, Inan, Ammar, Alim, Calvin, Firman, Prasetyo, Adrian selaku sahabat-sahabat terdekat penulis yang senantiasa mendukung, memberikan semangat, dan menemani penulis dalam menyelesaikan Tugas Akhir ini.
9. Teman-teman seperjuangan Lab IKTI dan OSIRIS yang telah menemani selama masa perkuliahan ini dan menjadi keluarga pertama penulis di lingkungan baru ini serta selalu mendukung untuk segera menyelesaikan Tugas Akhir ini.
10. Teman – teman satu kontrakan Wisma Permai, Alden, Leon, dkk.
11. Pihak lainnya yang telah membantu dan mendukung demi kelancaran dan kesuksesan penyelesaian buku Tugas Akhir ini.

Tidak ada sesuatu hal yang sempurna kecuali Allah SWT, tidak terkecuali juga untuk penyusunan buku Tugas Akhir ini. Penyusunan Tugas Akhir ini masih banyak kekurangan dan jauh dari kata sempurna. Oleh karena itu penulis menerima segala kritik dan saran demi kesempurnaan untuk perbaikan di masa mendatang. Semoga buku Tugas Akhir ini dapat memberikan manfaat bagi pembaca dan menjadi sebuah kontribusi bagi ilmu pengetahuan.

DAFTAR ISI

ABSTRAK.....	v
ABSTRACT.....	vii
KATA PENGANTAR	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xv
DAFTAR TABEL.....	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Relevansi Tugas Akhir	4
BAB II TINJAUAN PUSTAKA	5
2.1 Penelitian Sebelumnya	5
2.2 Dasar Teori.....	8
2.2.1 Router	8
2.2.2 Hotspot Network Access System Method.....	8
2.2.3 API (Application Programming Interface).....	10
2.2.4 Facebook.....	11
2.2.4.1 Facebook Graph API.....	11
2.2.4.2 Facebook OAuth API.....	12
2.2.5 Google	14
2.2.6 Twitter	16

2.2.7 Mikrotik	20
2.2.7.1 Mikrotik <i>RouterOS</i>	20
2.2.7.2 Fitur Mikrotik <i>RouterOS</i>	20
2.2.7.3 Mikrotik <i>RouterOS</i> API.....	23
BAB III METODOLOGI	29
3.1 <i>Flowchart</i> Metodologi.....	29
3.2 Tahap Pengerjaan Tugas Akhir.....	31
3.2.1 Analisa Kebutuhan.....	31
3.2.2 Desain Sistem.....	31
3.2.3 Implementasi.....	32
3.2.4 Pengujian	34
3.2.5 Implementasi.....	35
3.2.6 Dokumentasi	35
BAB IV PERANCANGAN	37
4.1 Analisa Kebutuhan	37
4.2 Desain Topologi dan Skenario.....	38
4.2.1 Desain Topologi Jaringan.....	38
4.2.2 Perancangan Skenario Otentikasi <i>Hotspot</i>	39
4.3 Perancangan Perangkat Lunak.....	40
4.3.1 <i>Framework</i> dan <i>Library</i>	40
4.3.2 Use Case Diagram.....	42
4.3.3 Sequence Diagram	49
4.3.3.1 Facebook OAuth Sequence Diagram	49
4.3.3.2 Google OAuth Sequence Diagram.....	50
4.3.3.3 Twitter OAuth Sequence Diagram.....	50
4.3.3.4 Memasukkan Data User.....	51
4.3.3.5 Router API Connection.....	52

4.4	Penggunaan Modul RouterOS	53
BAB V IMPLEMENTASI		55
5.1	Lingkungan Implementasi	55
5.2	Perancangan Aplikasi	56
5.2.1	Konfigurasi Aplikasi	58
5.2.1.1	Konfigurasi Google OAuth <i>Library</i>	58
5.2.1.2	Konfigurasi Facebook OAuth <i>Library</i>	59
5.2.1.3	Konfigurasi Facebook OAuth <i>Library</i>	60
5.2.2	Pembuatan <i>Model</i>	60
5.2.2.1	<i>User Models</i>	60
5.2.2.2	<i>Admin Models</i>	66
5.2.3	Pembuatan <i>Controller</i>	68
5.2.3.1	<i>Start Controller</i>	69
5.2.3.2	<i>Google Controller</i>	78
5.2.3.3	<i>Facebookauth Controller</i>	81
5.2.3.4	<i>Twitter Controller</i>	84
5.2.3.5	<i>Admin Controller</i>	89
5.3	Konfigurasi Router	97
5.3.1	Konfigurasi DHCP Client	97
5.3.2	Konfigurasi Address List	98
5.3.3	Konfigurasi NAT (<i>Network Address Translation</i>) 99	
5.3.4	Konfigurasi DHCP <i>Server</i>	101
5.3.5	Konfigurasi <i>Hotspot</i>	102
5.3.6	Konfigurasi <i>Hotspot User Profile</i>	110
5.3.7	Konfigurasi <i>Hotspot Walled garden</i>	111
5.3.8	Konfigurasi <i>Hotspot Redirect</i>	112

5.3.9	Konfigurasi <i>Static DNS</i>	113
5.4	Media Implementasi	113
5.4.1	Layout Media Implementasi	114
5.4.2	Kebutuhan Media Implementasi	119
BAB VI HASIL DAN ANALISIS		121
6.1	Aplikasi <i>Social Hotspot Authentication</i>	121
6.2	Pengujian Perangkat Lunak	128
6.3	Pembahasan	131
6.4	Analisa Bisnis	131
6.4.1	Analisa Paket Bisnis Implementasi	131
6.4.2	Analisa Biaya Awal	133
6.4.3	Analisa Biaya Tahunan	134
6.4.4	Analisa Keuntungan, ROI, dan <i>Payback Period</i> 138	
6.4.5	Analisa Biaya NPV (<i>Net Present Value</i>).....	147
BAB VII KESIMPULAN DAN SARAN		151
7.1	Kesimpulan.....	151
7.2	Saran.....	152
DAFTAR PUSTAKA.....		153
BIODATA PENULIS.....		157

DAFTAR GAMBAR

Gambar 2.1 Network Access Method.....	9
Gambar 2.2 Facebook Graph API	11
Gambar 2.3 Facebook Get Login Status Function.....	13
Gambar 2.4 Get Login Status JSON Return	13
Gambar 2.5 Skema Google OAuth API	15
Gambar 2.6 Twitter OAuth Sequence Diagram.....	19
Gambar 2.7 Contoh Command Word.....	25
Gambar 2.8 Contoh Query Word	26
Gambar 3.1 Tahapan Pengerjaan Tugas Akhir	30
Gambar 3.2 RouterOS API Repository	33
Gambar 4.1 Desain Topologi	38
Gambar 4.2 Skenario <i>Hotspot Authentication</i>	39
Gambar 4.3 <i>Use Case Diagram</i>	42
Gambar 4.4 Facebook OAuth Sequence Diagram	49
Gambar 4.5 Twitter OAuth Sequence Diagram.....	50
Gambar 4.6 Insert User Data Sequence Diagram	51
Gambar 4.7 Router API Connection Sequence Diagram	52
Gambar 5.1 Codeigniter <i>Folder Structure</i>	57
Gambar 5.2 Google OAuth <i>Library Configuration</i>	58
Gambar 5.3 Facebook OAuth <i>Library Configuration</i>	59
Gambar 5.4 Twitter OAuth <i>Library Configuration</i>	60
Gambar 5.5 <i>Get Router Config</i>	60
Gambar 5.6 <i>Get Router Config</i>	61
Gambar 5.7 <i>Update or Insert User</i>	61
Gambar 5.8 <i>Matching IP Address</i>	62
Gambar 5.9 <i>Matching User Session</i>	63
Gambar 5.10 <i>Check User Session</i>	64
Gambar 5.11 <i>Delete User Session</i>	65
Gambar 5.12 <i>Check Login Function</i>	66
Gambar 5.13 <i>Get User Function</i>	66
Gambar 5.14 Update Admin	67
Gambar 5.15 <i>Update Router Setting</i>	68
Gambar 5.16 <i>Function Construct</i>	69

Gambar 5.17 Start Controller Function Index	70
Gambar 5.18 Checking Session & Get IP Address	71
Gambar 5.19 Send Access Code via Email	72
Gambar 5.20 Get Router Config	73
Gambar 5.21 Start Controller Insert User Data to Mikrotik Hotspot	74
Gambar 5.22 Get Device IP Address	75
Gambar 5.23 Generate Random Character	76
Gambar 5.24 Function Input Unique Code	77
Gambar 5.25 Google Controller Function Index Check Session	78
Gambar 5.26 Google Controller Authenticate and Get User Data	79
Gambar 5.27 Google Controller Set User Data Session and Database Check	80
Gambar 5.28 Google Controller Logout Function	81
Gambar 5.29 Facebook Controller Function Index Check Session	81
Gambar 5.30 Facebook Controller Authenticate and Get User Data	82
Gambar 5.31 Facebook Controller Set User Data Session and Database Check	83
Gambar 5.32 Twitter Controller Logout Function	84
Gambar 5.33 Twitter Controller Function Index Check Session	84
Gambar 5.34 Twitter Controller Callback Function	85
Gambar 5.35 Function Success Twitter Auth Get User Data	86
Gambar 5.36 Twitter Controller Set User Data Session and Database Check	87
Gambar 5.37 Twitter Controller Failure Function	88
Gambar 5.38 Twitter Controller Logout Function	89
Gambar 5.39 Admin Controller Function Construct	89
Gambar 5.40 Admin Controller Index Function	90
Gambar 5.41 Admin Controller Login Function	91

Gambar 5.42 Admin <i>Controller Account Function</i>	92
Gambar 5.43 Admin <i>Controller Update Account Function</i> ...	93
Gambar 5.44 Admin <i>Controller Settings Function</i>	94
Gambar 5.45 Admin <i>Controller Update Setting Function</i>	95
Gambar 5.46 Admin <i>Controller View Visitor Function</i>	96
Gambar 5.47 Admin <i>Controller Logout Function</i>	97
Gambar 5.48 Konfigurasi <i>DHCP Client</i>	98
Gambar 5.49 <i>Address List Configuration</i>	98
Gambar 5.50 <i>NAT Configuration</i>	100
Gambar 5.51 <i>NAT Rules</i>	100
Gambar 5.52 <i>NAT DNS Setup</i>	101
Gambar 5.53 <i>DHCP Server Configuration</i>	101
Gambar 5.54 <i>Hotspot Server Configuration</i>	102
Gambar 5.55 <i>Hotspot Server Profile Configuration</i>	103
Gambar 5.56 <i>Hotspot Configuration Wizard</i>	104
Gambar 5.57 <i>Hotspot Interface Configuration</i>	104
Gambar 5.58 <i>Hotspot Local Address Configuration</i>	105
Gambar 5.59 <i>Hotspot Address Pool Setup</i>	105
Gambar 5.60 <i>Hotspot Certificate Configuration</i>	106
Gambar 5.61 <i>Hotspot SMTP Server</i>	106
Gambar 5.62 <i>Hotspot DNS Configuration</i>	107
Gambar 5.63 <i>Hotspot DNS Name</i>	107
Gambar 5.64 <i>Hotspot User Configuration</i>	108
Gambar 5.65 <i>Hotspot Setup Wizard Finish</i>	108
Gambar 5.66 <i>Default Login Page Hotspot Mikrotik</i>	109
Gambar 5.67 <i>Hotspot User Profile</i>	110
Gambar 5.68 <i>Hotspot Walled Garden Configuration</i>	111
Gambar 5.69 <i>Mikrotik Hotspot Web Server Redirect</i>	112
Gambar 5.70 Konfigurasi <i>Static DNS</i>	113
Gambar 5.71 <i>Layout Small Cafe Kategori 1</i>	115
Gambar 5.72 <i>Layout Large Cafe Kategori 2</i>	116
Gambar 5.73 <i>Layout Convention Hall Kategori 3</i>	117
Gambar 5.74 <i>Layout Convention Hall Kategori 4</i>	118
Gambar 6.1 <i>Landing Page Web Server</i>	121

Gambar 6.2 Pengguna Terotentikasi.....	122
Gambar 6.3 <i>Authenticated Users</i>	122
Gambar 6.4 <i>Active User Hotspot Mikrotik</i>	123
Gambar 6.5 <i>Mikrotik Hotspot User List</i>	123
Gambar 6.6 Dashboard Total Pengguna per Media Sosial ..	124
Gambar 6.7 <i>Datatables Social User Data</i>	125
Gambar 6.8 Contoh <i>User Data</i>	125
Gambar 6.9 Tautan Laman Profil Pengguna.....	126
Gambar 6.10 Update Konfigurasi <i>Router</i>	126
Gambar 6.11 Update Konfigurasi Akun	127
Gambar 6.12 Hasil <i>User Hotspot Database</i>	127
Gambar 6.13 Rumus ROI.....	139
Gambar 6.14 Rumus <i>Payback Period</i>	139

DAFTAR TABEL

Tabel 2.1 Daftar Penelitian Terdahulu.....	5
Tabel 2.2 API Word Length.....	24
Tabel 4.1 Tabel Kebutuhan Fungsional.....	37
Tabel 4.2 <i>Use Case Description</i> Otentikasi Media Sosial.....	43
Tabel 4.3 <i>Use Case Description</i> Keluar Media Sosial.....	44
Tabel 4.4 <i>Use Case Description Log Off Browsing Session</i> ..	45
Tabel 4.5 <i>Use Case Description Update Account</i>	46
Tabel 4.6 <i>Use Case Diagram Update Router Settings</i>	47
Tabel 4.7 <i>Use Case Diagram View Captured User Data</i>	48
Tabel 5.1 Daftar Teknologi Yang Digunakan Aplikasi	55
Tabel 5.2 Justifikasi Media Implementasi	114
Tabel 5.3 Perincian Harga Kebutuhan.....	119
Tabel 5.4 Tabel Perencanaan Kebutuhan per Kategori.....	120
Tabel 6.1 Hasil Pengujian Metode Otentikasi Pada Pengguna	128
Tabel 6.2 Hasil Pengujian Koneksi API ke <i>Router</i>	129
Tabel 6.3 Hasil Pengujian Otentikasi Halaman Admin	130
Tabel 6.4 Hasil Pengujian Form Update Konfigurasi <i>Router</i> dan Akun	130
Tabel 6.5 Bisnis Paket Implementasi Aplikasi	131
Tabel 6.6 Perincian Harga Kebutuhan.....	132
Tabel 6.7 Tabel Perencanaan Kebutuhan per Paket.....	133
Tabel 6.8 Tabel Estimasi Harga per Paket.....	134
Tabel 6.9 Biaya Kebutuhan per Tahun	134
Tabel 6.10 Total Biaya Operasional per Tahun	135
Tabel 6.11 Total Operasional Paket 1	136
Tabel 6.12 Total Operasional Paket 2	136
Tabel 6.13 Total Operasional Paket 3	137
Tabel 6.14 Total Operasional Paket 4	137
Tabel 6.15 Parameter Potential <i>Benefit</i>	140
Tabel 6.16 <i>Total Reach Justification</i>	141
Tabel 6.17 <i>Potential Benefit</i>	143
Tabel 6.18 ROI & <i>Benefit</i> Paket 1	144

Tabel 6.19 ROI & <i>Benefit</i> Paket 2	144
Tabel 6.20 ROI & <i>Benefit</i> Paket 3	145
Tabel 6.21 ROI & <i>Benefit</i> Paket 4	145
Tabel 6.22 Perhitungan NPV Paket 1	148
Tabel 6.23 Perhitungan NPV Paket 2	148
Tabel 6.24 Perhitungan NPV Paket 3	148
Tabel 6.25 Perhitungan NPV Paket 4	149

BAB I

PENDAHULUAN

Pada bab pendahuluan akan diuraikan proses identifikasi masalah penelitian yang meliputi latar belakang masalah, perumusan masalah, batasan masalah, tujuan tugas akhir, manfaat kegiatan tugas akhir dan relevansi terhadap pengerjaan tugas akhir. Berdasarkan uraian pada bab ini, harapannya gambaran umum permasalahan dan pemecahan masalah pada tugas akhir dapat dipahami.

1.1 Latar Belakang

Sebuah riset yang telah dilakukan oleh Janrain, perusahaan yang bergerak di bidang *customer profiling* dan *identity management* di Portland, Oregon, mengemukakan bahwa 87% pengguna media sosial pernah melihat metode otentikasi menggunakan media sosial di mobile atau web platform, dan 52% dari pengguna sudah pernah menggunakan metode otentikasi tersebut, dan mayoritas, sebanyak 27% menggunakan Facebook sebagai media sosial untuk melakukan otentikasi. Hal ini dirasa efektif, karena riset lain yang juga dilakukan oleh Janrain dan *Blue Reseach* terhadap 600 pengguna internet pada tahun 2011, 75% pengguna bosan dengan proses registrasi dan lebih suka disambut dengan otentikasi social media karena lebih praktis dan mudah, dan 66% menginginkan adanya otentikasi media sosial pada situs [1].

Menurut hasil survey APJII (Asosiasi Penyelenggara Jasa Internet Indonesia) pada tahun 2016 pengguna internet di Indonesia mencapai 132,7 juta[2]. Hal ini merupakan kenaikan angka yang cukup besar dibandingkan pengguna internet pada tahun 2014 yang mencapai 88,1 juta[2]. Hampir 90% dari pengguna internet di Indonesia menggunakan sosial media dengan angka mencapai 129,2 juta pengguna[2].

Berdasar pada hasil survey tersebut, maka sistem otentikasi menggunakan media sosial ini akan diterapkan sebagai

pelengkap otentikasi konvensional pada *hotspot* / wifi apabila hendak tersambung ke jaringan internet, hal ini tentu saja menguntungkan penyedia *hotspot* / wifi, karena dengan cara seperti ini, pemilik *hotspot* mengetahui okupansi pemakaian *hotspot* dan siapa saja yang telah terhubung ke jaringan dilihat dari akun sosial media yang terekam dan juga dapat memanfaatkan data yang telah terekam untuk melakukan promosi dan memberikan info terkait suatu hal yang lebih bersifat komersial dengan seizin *user*. Penyedia *hotspot* dapat merekam *user* yang telah terotentikasi *hotspot* yang mereka sediakan, dan juga pengguna hanya perlu melakukan otentikasi via social media yang mereka miliki.

1.2 Rumusan Masalah

Rumusan masalah yang akan dibahas pada penelitian tugas akhir ini berdasarkan pada latar belakang diatas yaitu antara lain:

1. Bagaimana membuat aplikasi web dengan menggunakan otentikasi sosial media?
2. Bagaimana menghubungkan aplikasi web tersebut dengan *router* Mikrotik?
3. Kondisi seperti apa yang cocok sebagai media implementasi *social media hotspot authentication*?
4. Bagaimana analisa bisnis dari segi *cost structure* dalam penerapan *social media hotspot authentication*?

1.3 Batasan Masalah

Batasan masalah yang digunakan dalam penelitian tugas akhir ini adalah:

1. Pembangunan aplikasi ini menggunakan *RouterOS* API dari Mikrotik
2. *Router* yang digunakan adalah Mikrotik RB series.

3. Pembangunan aplikasi dilakukan menggunakan bahasa pemrograman PHP.
4. Provider media sosial yang digunakan untuk otentikasi ada 3, yaitu gmail, Facebook, dan twitter.
5. Dari hasil otentikasi akan diambil *user profile* umum yang diizinkan oleh seperti nama, email, gender, dan avatar.
6. Sesi browsing akan didapatkan setelah *user* berhasil melakukan otentikasi.
7. Pemilihan *media placement* untuk
8. Implementasi yang cocok untuk penerapan *social media hotspot authentication* akan dijelaskan secara rinci pada bab perancangan dalam sub bab analisa bisnis dan *cost structure*.
9. Bentuk proteksi keamanan yang digunakan pada pengembangan sistem ini ada proteksi password pada SSID

1.4 Tujuan Penelitian

Tujuan dari pelaksanaan penelitian tugas akhir ini antara lain:

1. Membangun aplikasi otentikasi media sosial berbasis web.
2. Mengimplementasikan aplikasi yang telah dibuat dengan Mikrotik RouterOS API untuk mendapatkan *browsing session* setelah berhasil melakukan otentikasi.

1.5 Manfaat Penelitian

Manfaat yang diberikan dari penelitian tugas akhir ini adalah sebagai berikut :

Bagi Akademisi

Dapat dijadikan bahan untuk penelitian selanjutnya yaitu analisis manfaat praktis penerapan implementasi aplikasi, dan

juga analisis *performance* dari metode otentikasi media sosial dengan otentikasi biasa.

Bagi Organisasi

1. Bagi wifi *hotspot* provider, agar mengetahui profil orang yang terkoneksi dengan jaringan wifi, untuk alasan kontroling dan keamanan.
2. Selain itu, hal ini juga dapat dimanfaatkan untuk CRM karena terdapat profil pengguna yang terkoneksi jaringan.
3. Hal ini juga dapat digunakan sebagai sarana periklanan.

1.6 Relevansi Tugas Akhir

Pada penelitian tugas akhir ini akan diimplementasikan sebuah aplikasi web yang terhubung dengan *router* Mikrotik. Sehingga relevansi tugas akhir ini terhadap laboratorium Infrastruktur dan Keamanan Teknologi Informasi (IKTI) ialah berkaitan dengan penerapan mata kuliah bidang keilmuan laboratorium IKTI yaitu Teknologi Open-Source dan Terbaru, Analisa Desain, Konstruksi dan Perancangan Perangkat Lunak dan juga menyangkut masalah infrastruktur jaringan, yang terkait dengan mata kuliah Desain dan Manajemen Jaringan Komputer.

BAB II TINJAUAN PUSTAKA

Pada bab ini akan menjelaskan mengenai penelitian sebelumnya dan dasar teori yang akan dijadikan acuan atau landasan dalam pengerjaan tugas akhir ini.

2.1 Penelitian Sebelumnya

Subbab ini akan menjelaskan mengenai penelitian sebelumnya dimana dapat menambah pengetahuan yang terkait dengan penelitian ini sebagai referensi utama ataupun referensi pendukung penelitian. Daftar penelitian terdahulu akan dijelaskan pada Tabel 2.1.

Tabel 2.1 Daftar Penelitian Terdahulu

Penelitian 1	
Judul	One implementation of API interface for <i>RouterOS</i>
Nama Penulis	Stoitsov, Gencho Rangelov, Vasil
Tahun	2014
Gambaran Umum	Penelitian ini membahas tentang implementasi dari API <i>RouterOS</i> Mikrotik yang menggunakan bahasa pemrograman PHP, yang bertujuan untuk mendapatkan data untuk konfigurasi dan pengelolaan <i>routing</i> perangkat <i>router</i> yang menggunakan OS ini.
Keterkaitan Penelitian	Penelitian ini menjadi acuan karena sama – sama membahas tentang pemanfaatan Mikrotik

	<p><i>RouterOS</i> API dengan bahasa pemrograman PHP, namun dalam penelitian ini, API tidak berfungsi untuk pengelolaan <i>routing</i> namun lebih ke pemanfaatan untuk fungsi <i>hotspot</i> dengan <i>social media authentication</i></p>
Penelitian 2	
Judul	<p>SISTEM PENDAFTARAN HOTSPOT BERBASIS WEB PADA HOTSPOT MIKROTIK STMIK U ' BUDIYAH MENGGUNAKAN MIKROTIK APPLICATION PROGRAMMING INTERFACE (API), PHP, DAN MySQL Abstrak</p>
Nama Penulis	Zuhar Musliyana
Tahun	2014
Gambaran Umum	<p>Penelitian ini dikembangkan berbasis Web dengan memanfaatkan Bahasa Pemograman PHP, API PHP Class MikroTik, dan Database MySQL untuk pendaftaran, pengelolalaan, dan verifikasi data pengguna. Pengujian dilakukan pada proses pendaftaran pengguna <i>hotspot</i> yang di interintegrasikan dengan</p>

	database sentral kampus STMIK U ' BUDIYAH
Keterkaitan Penelitian	Penelitian ini lebih spesifik membahas role dan akses user untuk tiap – tiap pengguna <i>hotspot</i> yang ada di kampus. Penelitian ini juga menggunakan teknologi yang sama, yaitu <i>RouterOS</i> API dan bahasa pemrograman PHP, namun sistem otentikasi pada penelitian ini diintegrasikan dengan database kampus.
Penelitian 3	
Judul	Hotspot Network Access System and Method
Nama Penulis	Spencer, Ron Camps Tom Burchett Chris Gagne, Brad Madge, Rob
Tahun	2015
Gambaran Umum	Penelitian ini membahas mengenai berbagai macam metode dan skema yang dapat dilakukan ketika melakukan akses terhadap <i>hotspot</i> salah satunya membahas tentang metode dengan menggunakan <i>social media authentication</i>
Keterkaitan Penelitian	Penelitian ini digunakan sebagai acuan, karena pada penelitian tersebut membahas konseptual berbagai macam

	metode ketika melakukan akses ke <i>hotspot</i> , salah satunya yaitu dengan <i>social media authentication</i> dan <i>data capture</i> untuk <i>user profile</i>
--	---

2.2 Dasar Teori

Subbab ini akan menjelaskan mengenai konsep secara deskriptif mengenai segala hal yang berkaitan dengan penelitian yang dapat dijadikan sebagai pengetahuan dasar teoritis.

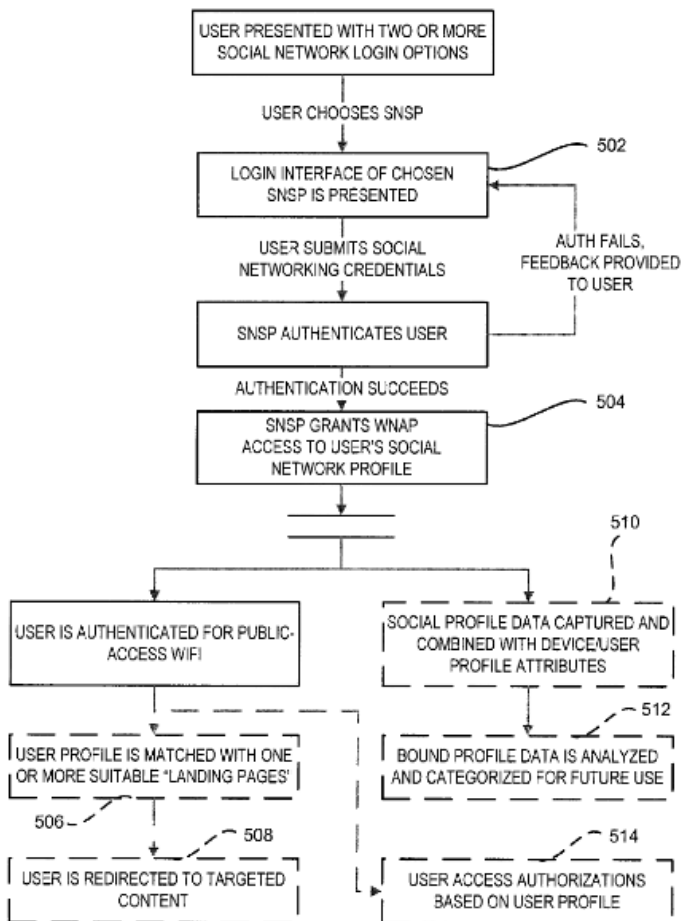
2.2.1 Router

Router atau penghalang merupakan suatu perangkat yang digunakan untuk mengirim paket data dari satu jaringan ke jaringan lain, dengan suatu proses yang disebut dengan penghalangan. Proses ini terjadi pada lapisan 3 (*internet protocol*) dari 7 *OSI layer*. *Router* memiliki beberapa jenis, namun yang paling umum digunakan adalah *router* yang berbasis teknologi TCP/IP, atau biasa dikenal dengan *IP router*. Internet merupakan contoh dari sebuah jaringan yang terdiri dari banyak *router* IP [3].

2.2.2 Hotspot Network Access System Method

Hotspot network access system membahas bagaimana suatu perangkat device dapat melakukan koneksi dengan perangkat yang menyediakan layanan jaringan secara nirkabel. Istilah "*hotspot*" disini digunakan untuk menentukan tempat akses publik, lokasi dan / atau wilayah geografis dimana *wireless access point* (WAP) menyediakan layanan jaringan nirkabel kepada pengunjung, yang menggunakan ponsel atau device lain melalui *wireless local area network* (WLAN), jaringan area metropolitan (MAN), *wide area network* (WAN), atau sejenisnya. Lokasi atau tempat *hotspot* banyak ditemukan pada restoran, stasiun kereta api, bandara, perpustakaan, kedai kopi,

toko buku, stasiun pengisian bahan bakar, universitas, sekolah, dan lokasi lainnya [4].



Gambar 2.1 Network Access Method

Banyak skema yang dapat digunakan untuk melakukan suatu konektivitas perangkat mobile / desktop ke suatu jaringan, skema yang paling umum digunakan adalah mendapatkan

konektivitas untuk mendapatkan layanan jaringan menggunakan *username* dan *password authentication*. Namun, juga ada skema lain yang dapat diterapkan, yaitu dengan menggunakan *social media authentication*, dimana akses ke layanan jaringan dapat dilakukan setelah pengguna melakukan otentikasi pada media sosial yang dimiliki seperti pada gambar 1. Secara garis besar, alur dari skema tersebut, pengguna diberikan opsi untuk melakukan otentikasi pada beberapa *social media network provider* (SNSP), SNSP yang digunakan untuk melakukan otentikasi, dimana sistem ini tertanam pada *web server* yang sudah disediakan. *Social network service provider* memiliki API agar dapat terkoneksi dengan akun media sosial yang dimiliki oleh pengguna. Tiap - tiap SNSP ini akan mengarahkan ke halaman otentikasinya masing - masing. Kemudian pengguna akan memasukkan *credentials* dan SNSP akan memunculkan notifikasi perizinan penggunaan *basic data* pengguna. Setelah otentikasi berhasil, maka pengguna tersebut diarahkan kembali ke halaman tertentu dari skema ini, dimana pengguna sudah terotentikasi untuk menggunakan akses layanan jaringan dengan *user profile* tertentu yang di *setting* dalam *router*, disisi lain, sistem dari *web server* akan melakukan penyimpanan data pribadi dari media sosial yang telah disetujui oleh pengguna.

2.2.3 API (Application Programming Interface)

API adalah singkatan dari *Application Programming Interface*. Secara umum, API adalah seperangkat metode komunikasi yang jelas agar memungkinkan adanya pertukaran informasi antar komponen aplikasi tanpa mengetahui apa yang terjadi dibalik layar, sehingga memudahkan dalam proses pembuatan aplikasi. API yang bagus memudahkan pengembangan program komputer dengan menyediakan semua komponen pengembangan, yang kemudian dapat digunakan oleh pengembang aplikasi sesuai dengan kebutuhannya. API tersedia untuk sistem berbasis web, sistem operasi, sistem basis data,

perangkat keras komputer atau berupa *software library*[5]. Dokumentasi untuk API biasanya disediakan untuk memudahkan penggunaan. Penggunaan API ini juga digunakan oleh *social media network provider* yang pada umumnya berbentuk *single sign-on* dengan menggunakan informasi yang ada dari layanan jejaring sosial seperti Facebook, Twitter atau Google+, untuk masuk ke situs web pihak ketiga tanpa harus membuat akun khusus untuk situs web tersebut. Hal ini dirancang untuk mempermudah otentikasi dan otorisasi bagi pengguna sekaligus memberikan informasi demografis yang lebih banyak dan lebih andal kepada para pengembang web. *Social media authentication* sudah sangat banyak digunakan pada semua situs [1].

2.2.4 Facebook

2.2.4.1 Facebook Graph API

Graph API adalah inti dari Platform Facebook, memungkinkan pengembang untuk membaca dan menulis data ke Facebook. *Graph* API menyajikan tampilan grafik sosial Facebook yang sederhana dan konsisten, yang secara merata mewakili objek dalam grafik (misalnya, orang, foto, acara, dan halaman) dan hubungan di antara mereka (misalnya, hubungan teman, konten bersama, dan tag foto) [6].



Gambar 2.2 Facebook Graph API

Graph API pada Facebook terdiri dari:

- a. **node** - pada dasarnya adalah "objek" seperti Pengguna, Foto, Halaman, Komentar
- b. **edge** - hubungan antar "objek" tersebut, seperti foto halaman, atau komentar foto
- c. **bidang** - info tentang "objek" tersebut, seperti hari ulang tahun seseorang, atau nama halaman

Graph API berbasis HTTP, sehingga *Graph* API dapat digunakan dengan berbagai bahasa yang mempunyai *library* HTTP, seperti *cURL* dan *urllib*. Berikut adalah contoh yang dihasilkan ketika melakukan permintaan untuk mendapatkan *user profile* dari data diri sendiri, dengan HTTP method GET, dengan URL `graph.facebook.com/me`.

`/me` merupakan endpoint khusus yang diterjemahkan menjadi ID pengguna seseorang yang token aksesnya digunakan untuk membuat permintaan, dan hal disini dapat diganti dengan mengganti url `/me` dengan username pengguna lain untuk mendapatkan basic profile serupa, tentu saja harus dengan seizin pengguna lain. Token akses yang dimaksud adalah, semacam *unique code string* yang mengidentifikasi pengguna, aplikasi, atau Halaman, dan dapat digunakan oleh aplikasi untuk membuat panggilan *Graph* API [7].

2.2.4.2 Facebook OAuth API

Facebook juga menyediakan *OAuth provider* yaitu SDK Javascript milik Facebook yang berfungsi untuk melakukan otentikasi pengguna agar dapat menggunakan *third party* platform yang memiliki fitur *login with* Facebook di platform tersebut, yang memungkinkan pengguna untuk memiliki akses dalam aplikasi dengan *credentials* Facebook mereka [8]. Sebelum menggunakan Facebook *OAuth Provider* harus memiliki ID aplikasi untuk pengembang. Dalam pembuatan ID

aplikasi ada beberapa pengaturan lain yang juga harus diperhatikan [9]

1. *Redirect URL*, halaman *redirect* ketika pengguna berhasil melakukan otentikasi.
2. *Check login status*, untuk melihat apakah seseorang sudah masuk ke aplikasi. Berikut adalah contoh *snippet* untuk mendapatkan *getLoginStatus* dan *return value* nya dalam bentuk JSON.

```
FB.getLoginStatus(function(response) {
  statusChangeCallback(response);
});
```

Gambar 2.3 Facebook Get Login Status Function

```
{
  status: 'connected',
  authResponse: {
    accessToken: '...',
    expiresIn: '...',
    signedRequest: '...',
    userID: '...'
  }
}
```

Gambar 2.4 Get Login Status JSON Return

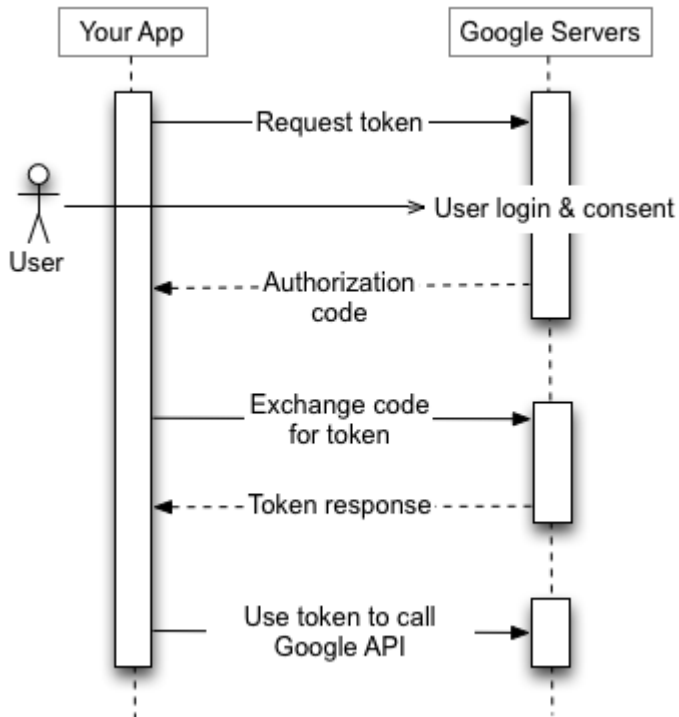
Status menetapkan status masuk orang yang menggunakan aplikasi. status dapat berupa salah satu dari berikut ini:

- a. *connected* - Pengguna tersebut masuk ke Facebook, dan telah masuk ke aplikasi.

- b. *not_authorized* - Pengguna tersebut masuk ke Facebook, namun menolak memberikan izin.
 - c. *unknown* - Pengguna tersebut tidak masuk ke Facebook, sehingga tidak tahu apakah pengguna tersebut telah masuk ke aplikasi atau belum. Atau `FB_logout()` telah dipanggil sebelumnya, dan oleh karenanya tidak dapat masuk ke Facebook.
3. *Get authorization*, memunculkan dialog aplikasi untuk meminta izin kepada pengguna untuk melakukan pengambilan profil dasar pengguna
4. *Logout*, untuk memungkinkan pengguna keluar dari aplikasi.

2.2.5 Google

Google API menggunakan protokol OAuth 2.0 untuk otentikasi dan otorisasi. Google mendukung skenario OAuth 2.0 yang umum seperti aplikasi web server, pemasangan, dan sisi klien. Untuk memulai, dapatkan kredensial klien OAuth 2.0 dari Konsol Google API. Kemudian aplikasi klien meminta token akses dari Server Otorisasi Google, mengekstrak token dari *response*, dan mengirimkan token ke Google API yang ingin diakses. Untuk demonstrasi interaktif menggunakan OAuth 2.0 dengan Google (termasuk opsi untuk menggunakan kredensial klien), bereksperimen dengan OAuth 2.0 Playground [8].



Gambar 2.5 Skema Google OAuth API

Urutan otorisasi dimulai dengan aplikasi yang membuat *web service request* ke URL Google untuk kode otorisasi. *Response* berisi beberapa parameter, termasuk URL dan kode yang ditunjukkan aplikasi kepada pengguna. Pengguna memperoleh URL dan kode dari perangkat, kemudian akan membuka browser dan menavigasi ke URL yang ditentukan, memasukkan *credentials*, dan memasukkan kode. Sementara itu, pengguna menyetujui akses, dan *response* dari server google terdapat dua token, yaitu *access token* dan *refresh token* [9]. Aplikasi harus menyimpan *refresh token* untuk penggunaan selanjutnya dan menggunakan *access token* untuk mengakses Google API. Setelah *access token* kadaluarsa, aplikasi menggunakan *refresh*

token untuk mendapatkan yang baru [10]. Beberapa hal yang dilakukan oleh Google OAuth JS library dalam melakukan otorisasi terhadap pengguna kurang lebih seperti berikut:

1. Membuat redirect URL untuk *authorization server* Google dan menyediakan metode untuk *redirecting* ke URL tersebut.
2. Mengendalikan redirect dari *authorization server* Google kembali ke aplikasi pengembang.
3. Melakukan validasi *access token* yang dikembalikan oleh *authorization server*.
4. Menyimpan *access token* yang telah didapatkan sebelumnya yang dikirim *authorization server* dan akan digunakan kembali.

2.2.6 Twitter

Twitter sebagai social network service provider menyediakan berbagai macam API yang dapat dimanfaatkan oleh pengguna untuk tujuan tertentu, contohnya adalah Twitter OAuth API dimana layanan ini memperbolehkan pengembang aplikasi untuk menggunakan aplikasinya dengan *credentials* akun twitter yang dimiliki pengguna. Untuk mendapatkan akses via OAuth API, terdapat grant type *Authorization Code*. Ada beberapa tahapan dalam prosesnya [11].

1. *Authorization code*

Proses yang pertama ini melibatkan pengembang aplikasi selaku *resource provider*, *client*, dan *OAuth Server*. Tujuan dari proses yang pertama ini yaitu mendapatkan *authorization code* yang akan digunakan untuk mendapatkan *access token*. Proses ini dimulai ketika *resource owner* menggunakan *Third Party Apps* untuk melakukan sesuatu [12]. Lalu *client* akan pergi ke OAuth Server dengan memanggil endpoint */authorize*. Terdapat

beberapa parameter yang dibawa ketika client request endpoint */authorize*:

a. **response_type**

REQUIRED. Diisi 'code' untuk standar OAuth2 *authorization flow*.

b. **client_id**

REQUIRED. Client id merupakan identitas dari *Third Party Apps*. Client id didapatkan setelah mendaftarkan *Third Party Apps*.

c. **Scope**

OPTIONAL. *Scope* digunakan untuk mendefinisikan informasi apa saja yang akan diminta. Pada dasarnya, informasi dasar yang akan didapat adalah nama, email, jenis kelamin.

d. **State**

RECOMMENDED. *Random String*, Di generate oleh client. Digunakan untuk mencegah serangan CSRF (baca: sea-surf) dan digunakan oleh klien untuk maintain antara request dan callback.

e. **redirect_uri**

Digunakan oleh OAuth Server untuk mengembalikan hasil. Redirect URI yang dikirim *Third Party Apps* ke *OAuth Server* harus sama dengan Redirect URI yang didaftarkan ketika *Third Party Apps* mendapatkan Client Id.

Apabila parameter yang diterima oleh *service provider* valid, maka pengguna akan di *redirect* ke halaman login twitter, dan apabila berhasil terotentikasi, maka akan muncul dialog persetujuan, dimana resource owner ingin mendapatkan *profile*

data client atau dalam hal ini disebut *scopes* dari akun twitternya.

2. *Get access token*

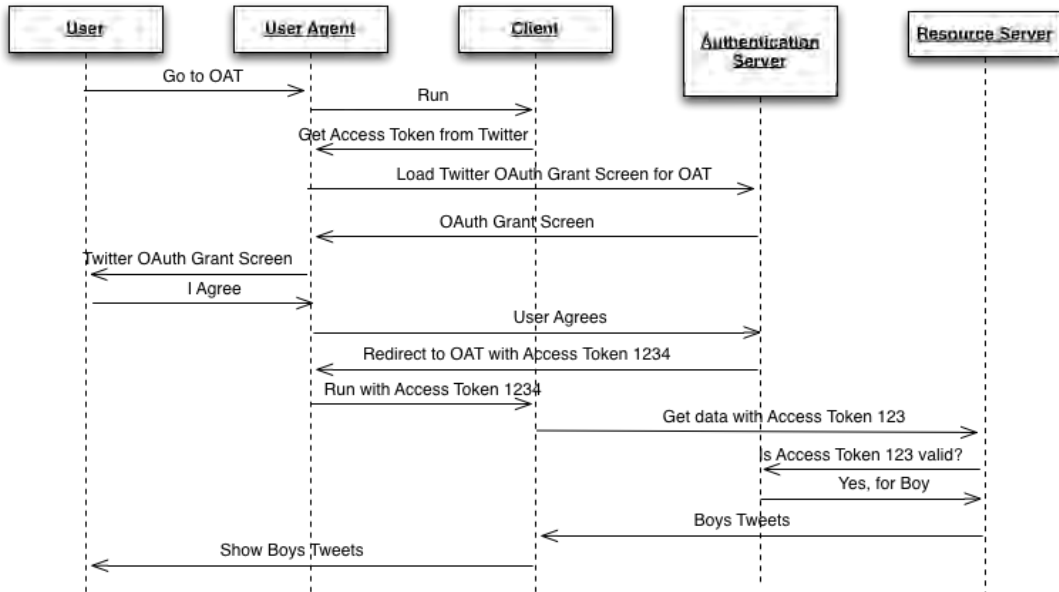
Setelah memiliki *authorization code* hal lain yang dibutuhkan yaitu *access token*, untuk mendapatkan *access token*, *request endpoint* bernama */token* dengan method POST. *OAuth Server* akan memeriksa validitas data yang dikirim oleh client [13]. Jika aman maka *OAuth Server* akan mengirimkan kembali hasilnya ke client.

- *access_token* digunakan untuk mengakses Resources menggunakan HTTP Bearer.
- *refresh_token* akan digunakan untuk me-generate Access Token yang baru, setelah *access token* kadaluarsa. Masa kadaluarsa *refresh token* akan lebih lama dibanding *access token*.
- *expires_in* merupakan masa kadaluarsa dari *access_token* dalam satuan detik.

3. *Access resource with access token*

Setelah mendapatkan *access token*, maka client dapat mengakses resource server, pada gambar, client mencoba untuk mengirimkan request kepada resource server dengan menggunakan *Bearer Authorization Header* dengan *access token*nya, kemudian, resource server melakukan validasi token tersebut ke *OAuth Server*, untuk memastikan validitasnya. Setelah melakukan validitas, *OAuth Server* mengirim kembali hasil validasinya kepada *resource server*, apabila valid, maka client akan mendapatkan data yang di *request*.

Access token disini memiliki batas kadaluarsa, maka dalam waktu tertentu *access token* perlu di refresh. Dengan refresh token kita tidak perlu melakukan *authorization* kembali untuk mendapatkan *access_token* yang baru



Gambar 2.6 Twitter OAuth Sequence Diagram

2.2.7 Mikrotik

MikroTik adalah perusahaan Latvia yang didirikan pada tahun 1996 untuk mengembangkan sistem *router* dan ISP nirkabel. MikroTik sekarang menyediakan perangkat keras dan perangkat lunak untuk konektivitas Internet di sebagian besar negara di seluruh dunia.

2.2.7.1 Mikrotik RouterOS

MikroTik *RouterOS*TM merupakan sistem operasi yang diperuntukkan sebagai *network router* yang dapat digunakan untuk menjadikan komputer biasa menjadi *router network* yang handal dan menggunakan sistem operasi Linux *base* yang mencakup berbagai fitur yang dibuat untuk *ip network* dan jaringan *wireless*. Fitur-fitur tersebut diantaranya: *Firewall & Nat, Routing, Hotspot, Point to Point Tunneling Protocol, DNS server, DHCP server, Hotspot*, dan masih banyak lagi fitur lainnya. MikroTik *routerOS* merupakan sistem operasi Linux *base* yang diperuntukkan sebagai *network router* [14]. Didesain untuk memberikan kemudahan bagi penggunaannya. Administrasinya bisa dilakukan melalui Windows Application (WinBox).

2.2.7.2 Fitur Mikrotik RouterOS

Mikrotik memiliki beberapa fitur yang berfungsi untuk melakukan pengelolaan jaringan, berikut adalah fitur – fitur yang dapat dilakukan oleh Mikrotik *router* [15].

1. **Address List:** Pengelompokan IP Address berdasarkan nama.
2. **Asynchronous:** Mendukung serial PPP dial-in / dial-out, dengan otentikasi CHAP, PAP, MSCHAPv1 dan MSCHAPv2, Radius, dial on demand, modem pool hingga 128 ports.

3. **Bonding**: Mendukung dalam pengkombinasian beberapa antarmuka ethernet ke dalam 1 pipa pada koneksi cepat.
4. **Bridge**: Mendukung fungsi bridge spinning tree, multiple bridge interface, bridging firewalling.
5. **Data Rate Management**: QoS berbasis HTB dengan penggunaan burst, PCQ, RED, SFQ, FIFO queue, CIR, MIR, limit antar peer to peer.
6. **DHCP**: Mendukung DHCP tiap antarmuka; DHCP Relay; DHCP Client, multiple network DHCP; static and dynamic DHCP leases.
7. **Firewall dan NAT**: Mendukung pemfilteran koneksi peer to peer, source NAT dan destination NAT. Mampu memfilter berdasarkan MAC, IP address, range port, protokol IP, pemilihan opsi protokol seperti ICMP, TCP Flags dan MSS.
8. **Hotspot**: Hotspot gateway dengan otentikasi RADIUS. Mendukung limit data rate, SSL, HTTPS.
9. **IPSec**: Protokol AH dan ESP untuk IPSec; MODP Diffie-Hellmann groups 1, 2, 5; MD5 dan algoritma SHA1 hashing; algoritma enkripsi menggunakan DES, 3DES, AES-128, AES-192, AES-256; Perfect Forwarding Secresy (PFS) MODP groups 1, 2,5.
10. **ISDN**: mendukung ISDN dial-in/dial-out. Dengan otentikasi PAP, CHAP, MSCHAPv1 dan MSCHAPv2, Radius. Mendukung 128K bundle, Cisco HDLC, x751, x75ui, x75bui line protokol.
11. **M3P**: MikroTik Protokol Paket Packer untuk wireless links dan ethernet.
12. **MNDP**: MikroTik Discovery Neighbour Protokol, juga mendukung Cisco Discovery Protokol (CDP).

13. **Monitoring / Accounting:** Laporan Traffic IP, log, statistik graph yang dapat diakses melalui HTTP.
14. **NTP:** Network Time Protokol untuk server dan clients; sinkronisasi menggunakan system GPS.
15. **Point to Point Tunneling Protocol:** PPTP, PPPoE dan L2TP Access Concentrator; protokol otentikasi menggunakan PAP, CHAP, MSCHAPv1, MSCHAPv2; otentikasi dan laporan Radius; enkripsi MPPE; kompresi untuk PPOE; limit data rate.
16. **Proxy:** Cache untuk FTP dan HTTP proxy server, HTTPS proxy; transparent proxy untuk DNS dan HTTP; mendukung protokol SOCKS; mendukung parent proxy; static DNS.
17. **Routing:** Routing statik dan dinamik; RIP v1/v2, OSPF v2, BGP v4.
18. **SDSL:** Mendukung Single Line DSL; mode pemutusan jalur koneksi dan jaringan.
19. **Simple Tunnel:** Tunnel IP/IP dan EoIP (Ethernet over IP).
20. **SNMP:** Simple Network Monitoring Protocol mode akses read-only.
21. **Synchronous:** V.35, V.24, E1/T1, X21, DS3 (T3) media ttypes; sync-PPP, Cisco HDLC; Frame Relay line protokol; ANSI-617d (ANDI atau annex D) dan Q933a (CCITT atau annex A); Frame Relay jenis LMI.
22. **Tool:** Ping, Traceroute; bandwidth test; ping flood; telnet; SSH; packet sniffer; Dinamik DNS update.
23. **UPnP:** Mendukung antarmuka Universal Plug and Play.
24. **VLAN:** Mendukung Virtual LAN IEEE 802.1q untuk jaringan ethernet dan wireless; multiple VLAN; VLAN bridging.

25. **VoIP**: Mendukung aplikasi voice over IP.
26. **VRRP**: Mendukung Virtual *Router* Redudant Protocol.
27. **WinBox**: Aplikasi mode GUI untuk meremote dan mengkonfigurasi MikroTik *RouterOS*.

2.2.7.3 Mikrotik *RouterOS* API

RouterOS dari Mikrotik, memiliki API tersendiri agar para pengembang dapat melakukan pengembangan kedepan. Application Programmable Interface (API) memungkinkan pengembang membuat solusi perangkat lunak kustom agar dapat berkomunikasi dengan *RouterOS* untuk mengumpulkan informasi, menyesuaikan konfigurasi dan mengelola *router*. API mengikuti sintaks dari command line interface (CLI). Hal ini dapat digunakan untuk membuat alat konfigurasi yang diterjemahkan atau kustom untuk membantu kemudahan penggunaan yang berjalan dan mengelola *router* dengan *RouterOS* [16].

Protokol

Router Mikrotik memiliki protokol tersendiri dalam melakukan komunikasi, dimana komunikasi ini dilakukan dengan mengirimkan beberapa kalimat dan akan mendapat respon satu atau beberapa kalimat. Tiap kalimat yang dikirim ke *router* melalui API harus mengandung perintah sebagai kata pertama diikuti dengan kata lain, akhir dari kalimat ditandai dengan *zero length word*.

1. *API words*

Kata adalah bagian dari kalimat. Setiap kata harus dikodekan dengan cara tertentu - panjang kata diikuti oleh konten kata. Panjang kata harus diberikan sebagai hitungan byte yang akan dikirim. Panjang kata dikodekan sebagai berikut:

Tabel 2.2 API Word Length

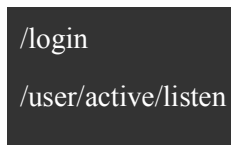
Nilai panjang	# byte	Encoding
$0 \leq \text{len} \leq 0x7F$	1	len, byte terendah
$0x80 \leq \text{len} \leq 0x3FFF$	2	len 0x8000, dua byte lebih rendah
$0x4000 \leq \text{len} \leq 0x1FFFFFF$	3	len 0xC00000, tiga byte lebih rendah
$0x200000 \leq \text{len} \leq 0xFFFFFFFF$	4	len 0xE0000000
$\text{len} > 0x10000000$	5	0xF0 dan len sebagai empat byte

- Setiap kata dikodekan sebagai panjang, diikuti oleh banyak byte konten
- Kata-kata dikelompokkan ke dalam kalimat. Akhir kalimat diakhiri dengan kata nol panjang
- Skema memungkinkan pengkodean panjang hingga **0x7FFFFFFFFF**, hanya empat byte panjang yang didukung
- Byte **len** dikirim terlebih dahulu (network order)

- e. Jika byte pertama kata adalah $\geq 0xF8$, maka itu adalah byte kontrol yang dicadangkan. Setelah menerima klien API byte kontrol yang tidak diketahui tidak dapat melanjutkan, karena tidak dapat mengetahui bagaimana menafsirkan byte berikutnya

2. *Command words*

Kata pertama dalam *command words* harus diikuti dengan kata-kata atribut dan kata nol panjang atau kata penghentian. *Command words* harus dimulai dengan '/'. *Command words* biasanya digunakan untuk controlling *router* via CLI, dengan spasi diganti dengan '/'. Contoh:



```
/login
/user/active/listen
```

Gambar 2.7 Contoh Command Word

3. *Attribute words*

Setiap *command word* memiliki daftar *attribute word* tergantung pada konten. Struktur *attribute word* terdiri dari 5 bagian dalam urutan ini:

- a. panjang encode
- b. Tanda prefix konten yang sama (=)
- c. nama atribut
- d. memisahkan sama dengan tanda (=)
- e. nilai *attribute* jika ada. Ada kemungkinan atribut itu tidak memiliki nilai

4. Query Word

Kalimat dapat memiliki parameter permintaan tambahan yang membatasi ruang lingkungannya. Contoh kalimat menggunakan atribut *query word*:

```
/interface/print
?type=ether
?type=vlan
?#!
```

Gambar 2.8 Contoh Query Word

- a. Kata kueri dimulai dengan '?'.
- b. Saat ini hanya perintah *cetak yang* menangani kata-kata query.

5. Reply word

Ini hanya dikirim oleh *router*. Ini hanya dikirim sebagai respon dari kalimat yang dikirimkan oleh klien.

- a. *Reply word* pertama dimulai dengan '!'.
- b. Setiap *sentence* yang dikirim menghasilkan setidaknya satu *reply* (jika koneksi tidak dihentikan)
- c. *Reply* terakhir untuk setiap kalimat adalah *reply* yang memiliki kata pertama *!done*
- d. *Errors danexception* dimulai dengan *!trap*
- e. Data balasan dimulai dengan *!re*
- f. Jika koneksi API ditutup, *RouterOS* mengirim *!fatal* dengan alasan sebagai balasan dan kemudian menutup koneksi;

Kalimat API

Kalimat API adalah objek komunikasi utama dengan menggunakan API.

- a. Kalimat kosong diabaikan
- b. Kalimat diproses setelah menerima *zero length word*
- c. Ada batas jumlah dan ukuran kalimat yang dapat dikirim klien sebelum masuk
- d. Urutan kata atribut tidak boleh diandalkan. Karena pesanan dan hitungan diubah oleh atribut *.proplist*
- e. Struktur kalimat adalah sebagai berikut:
 - Kata pertama harus mengandung *command word*
 - Harus mengandung *zero length word* untuk mengakhiri kalimat
 - Tidak boleh mengandung beberapa atau beberapa *attribute word*. Tidak ada urutan tertentu pada kata atribut apa yang harus dikirim dalam kalimat, ketertiban tidak penting untuk *attribute words*

Tidak boleh mengandung *query words*. Urutan *query words* dalam kalimat itu penting.

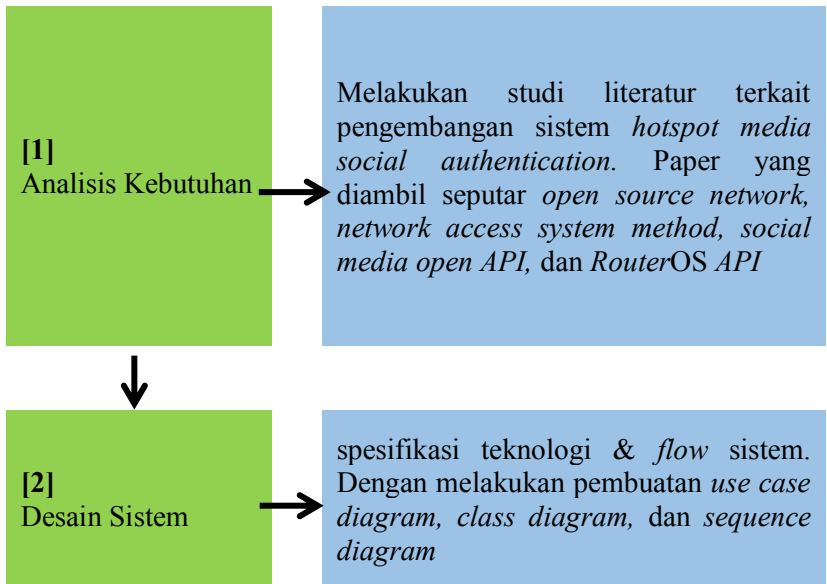
(halaman ini sengaja dikosongkan)

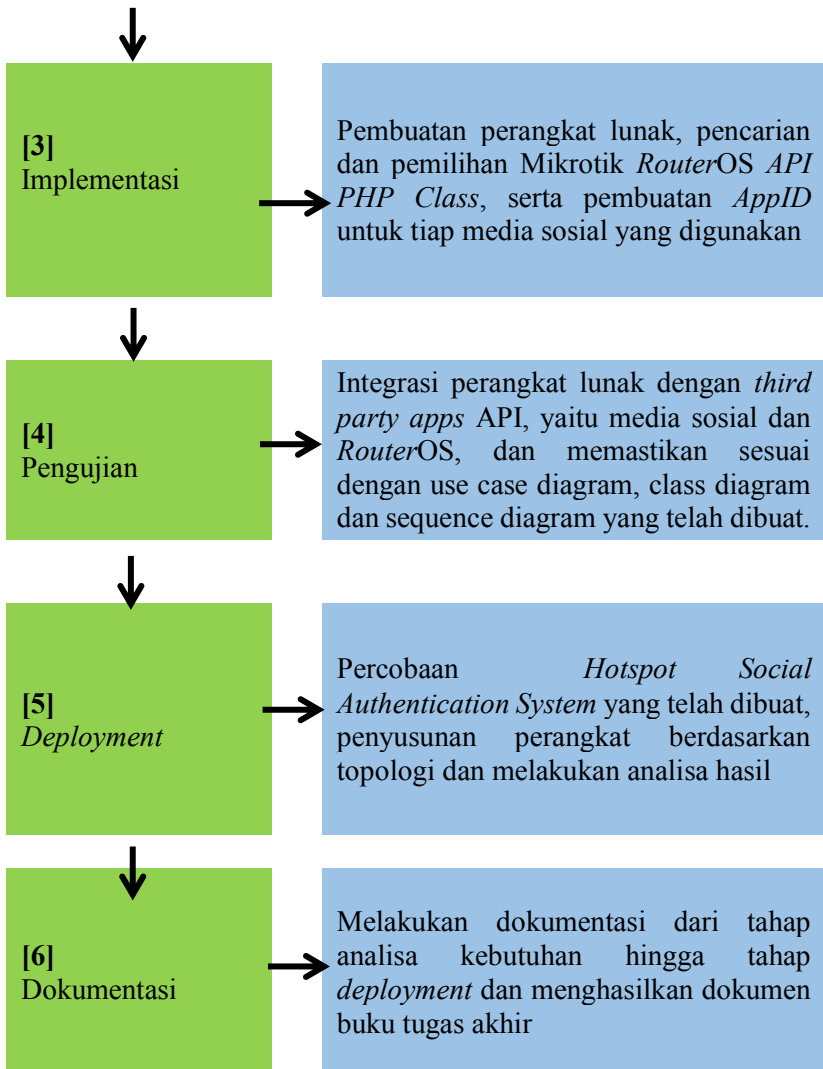
BAB III METODOLOGI

Bab ini akan menjelaskan mengenai alur atau tahapan metode penelitian yang dilakukan oleh penulis dalam pengerjaan penelitian tugas akhir ini. Metode penelitian juga digunakan sebagai pedoman dalam pengerjaan tugas akhir agar mempunyai arah dan terstruktur. Berikut tahapan dari pengerjaan tugas akhir :

3.1 *Flowchart* Metodologi

Dalam pengerjaan penelitian tugas akhir ini terdapat 3 tahapan yaitu tahap persiapan, tahap pengumpulan data, tahap analisis dan pembahasan. *Flowchart* tahap pelaksanaan tugas akhir beserta penjelasannya diilustrasikan pada **Error! Reference source not found.**





Gambar 3.1 Tahapan Pengerjaan Tugas Akhir

3.2 Tahap Pengerjaan Tugas Akhir

Sesuai dengan tahapan pengerjaan tugas akhir pada **Error! Reference source not found.**, berikut penjelasan mendetail pada setiap gambaran langkah-langkah yang digunakan dalam pengerjaan tugas akhir ini untuk setiap tahapannya.

3.2.1 Analisa Kebutuhan

Tahapan ini merupakan fase pertama dalam pengerjaan Tugas Akhir ini. Pada tahap studi literatur dilakukan penggalan kebutuhan pengetahuan terkait dengan studi kasus yang akan diambil. Paper yang dijadikan referensi berasal dari luar dan dalam negeri yang menggunakan metode pendukung penelitian. Paper-paper yang diambil adalah penelitian seputar *open source network*, *network access system method*, *social media authentication*, dan PHP class API untuk *RouterOS*.

3.2.2 Desain Sistem

Pada tahapan ini akan didefinisikan teknologi dan *flow* yang akan digunakan dalam pembuatan sistem. Pemilihan teknologi yang akan digunakan untuk implementasi akan berdasarkan teknologi yang sudah ada dalam bab batasan masalah, yaitu untuk perangkat *router* menggunakan Mikrotik RB series, kemudian mencari open source API untuk Mikrotik *RouterOS* dengan bahasa pemrograman PHP. Selanjutnya dilakukan integrasi antara sistem yang dibuat dengan Mikrotik *Router* melalui API.

3.2.2.1 Desain Topologi dan Arsitektur Sistem

Dalam pembuatan sistem, maka dibutuhkan untuk melakukan perancangan topologi dan juga arsitektur sistem, mulai dari melakukan perincian terhadap perangkat – paerangkat yang dibutuhkan, kemudian menggambar skema topologi dan juga arsitektur sistem agar dapat berjalan dengan maksimal.

3.2.2.2 Pembuatan *Use Case Diagram*

Pada tahapan ini, akan dibuat *use case diagram*, untuk menentukan apa saja yang dapat dilakukan oleh aktor dan batasannya terhadap sistem. Pada pembuatan *use case diagram* akan ada dua aktor, yaitu pengguna dan admin.

3.2.2.3 Pembuatan *Entity Relationship Diagram*

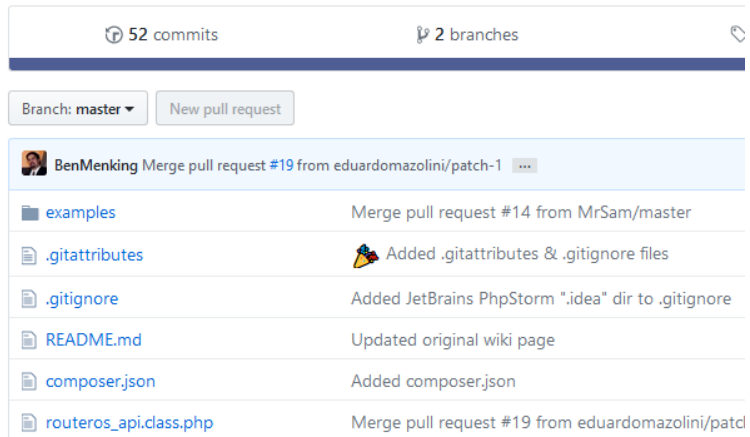
Pada tahapan ini, akan dibuat *entity relationship diagram*, untuk menentukan akan seperti apa hubungan tiap entitas, hal ini dilakukan sebagai pedoman untuk pembuatan database.

3.2.3 Implementasi

3.2.3.1 Pencarian Mikrotik RouterOS API

Pada tahapan ini, akan dilakukan pencarian Mikrotik RouterOS API yang akan digunakan untuk proses integrasi. Mikrotik RouterOS API sudah banyak tersedia di beberapa website *repository* seperti GitHub dll. Namun disini harus dipilah – pilah dan disesuaikan dengan bahasa pemrograman yang akan digunakan untuk implementasi, yaitu PHP.

Client API for RouterOS/Mikrotik



Gambar 3.2 RouterOS API Repository

Gambar diatas merupakan salah satu contoh open Mikrotik *RouterOS API PHP Class* yang telah tersedia pada situs GitHub.

3.2.3.2 Pembangunan Perangkat Lunak

Pada tahapan ini dilakukan pembuatan perangkat lunak website, dengan bahasa PHP menggunakan framework PHP yang bernama *CodeIgniter* versi 3.0.4 *stable release*. Pembuatan perangkat lunak ini mengacu pada *use case diagram* yang telah dibuat dan alur pembuatan sesuai dengan *sequence diagram* yang telah di desain sebelumnya. Pada tahapan ini akan diimplementasikan Mikrotik *RouterOS API* agar perangkat lunak dapat berkomunikasi dengan *router* Mikrotik yang telah disiapkan. Disini juga akan dilakukan pembuatan desain tampilan keseluruhan untuk tiap – tiap aktivitas, dan juga integrasi API untuk media sosial.

3.2.3.3 Pembuatan App ID Social Media

Pada tahapan ini, dilakukan pembuatan App ID untuk tiap – tiap media sosial yang akan digunakan sebagai otentikasi masuk agar mendapatkan *read and write permission* untuk melihat basic info pengguna yang terotentikasi. Untuk media sosial yang dimaksud yaitu, Google, Facebook, dan juga Twitter yang sudah sangat populer di dunia. Pembuatan App ID ini masuk kedalam *developer account* yang didalamnya terdapat *console* khusus pengembang dimana banyak disediakan API yang lain lain, untuk API yang akan digunakan hanya terbatas pada API untuk otentikasi atau biasa disebut dengan *OAuth* API. Berikut adalah contoh developer console yang dimiliki oleh masing – masing media sosial yang dimaksud.

3.2.3.4 Transformasi Perubahan Kuantitas

Dalam tahapan ini dilakukan perubahan kuantitas indikator ke dalam satuan finansial terkait dengan hal tersebut. Kuantitas perubahan yang telah didefinisikan pada tahap sebelumnya digunakan sebagai *input* dalam melakukan analisis biaya. Acuan harga digunakan untuk dapat menkonversi nilai kuantitas perubahan menjadi satuan angka finansial. Keluaran (*output*) yang dihasilkan dalam tahap ini merupakan hasil perhitungan manfaat *intangible* dari implementasi aplikasi layanan *Service Desk* ITS dalam satuan angka finansial.

3.2.4 Pengujian

Dalam tahapan pengujian disini akan dilakukan percobaan untuk sistem yang dibuat, mulai dari penyusunan perangkat – perangkat sesuai dengan topologi yang telah dibuat, untuk perangkat lunak yang telah dibuat, apakah sudah sesuai dengan *use case* dan *sequence diagram* yang telah dibuat, dan parameter HTTP method yang digunakan apakah sudah cocok.

3.2.5 Implementasi

Pada tahapan ini apabila pada tahap pengujian sudah lengkap, maka tahapan terakhir yang dilakukan yaitu *deployment* sistem, dimana disini akan melakukan *setting* environment terhadap sistem, seperti penyusunan perangkat sesuai dengan topologi yang telah dibuat, dan juga *deployment* perangkat lunak kedalam *web server* yang telah disiapkan, dan *media placement* yang memenuhi kondisi dan kriteria yang ditentukan pada bab perancangan

3.2.6 Dokumentasi

Pada tahapan terakhir ini akan dilakukan pembuatan laporan dalam bentuk buku tugas akhir yang disusun sesuai format yang telah ditentukan. Buku ini berisi dokumentasi langkah-langkah pengerjaan tugas akhir secara rinci. Buku ini diharapkan dapat bermanfaat sebagai referensi untuk pengerjaan penelitian lain, serta sebagai acuan untuk pengembangan lebih lanjut terhadap topik penelitian yang serupa.

(halaman ini sengaja dikosongkan)

BAB IV PERANCANGAN

Bab ini akan menjelaskan mengenai rancangan tugas akhir dalam penelitian yang akan dilakukan sebagai panduan untuk melaksanakan penelitian tugas akhir yang terdiri dari analisa kebutuhan, desain sistem, yang didalamnya terdapat perancangan skenario otentikasi, perancangan perangkat lunak, hingga penentuan modul *RouterOS* yang akan digunakan untuk skenario dan topologi yang telah ditentukan.

4.1 Analisa Kebutuhan

Tahap awal dari perancangan dimulai dari mengumpulkan studi literatur dari berbagai sumber yang membahas tentang pemanfaatan Mikrotik *RouterOS* API PHP Class lalu menyesuaikan dengan perangkat lunak yang ingin dikembangkan. Untuk perangkat lunak yang akan dikembangkan sendiri memiliki beberapa kebutuhan baik dari segi fungsional ataupun non fungsional:

1. Fungsional

Tabel 4.1 Tabel Kebutuhan Fungsional

Kode	Deskripsi
F1	Aplikasi dapat tersambung ke <i>router</i> Mikrotik dengan menggunakan PHP Class API
F2	Aplikasi dapat menjembatani akses internet dengan metode otentikasi melalui media sosial
F3	Aplikasi dapat mendeteksi adanya multi session dan memberikan notifikasi dan input kode yang dikirimkan via email
F4	Aplikasi dapat melakukan pencatatan setiap user yang terkoneksi ke <i>router</i> di <i>hotspot</i> users, dan ke database MySQL

2. Non-fungsional

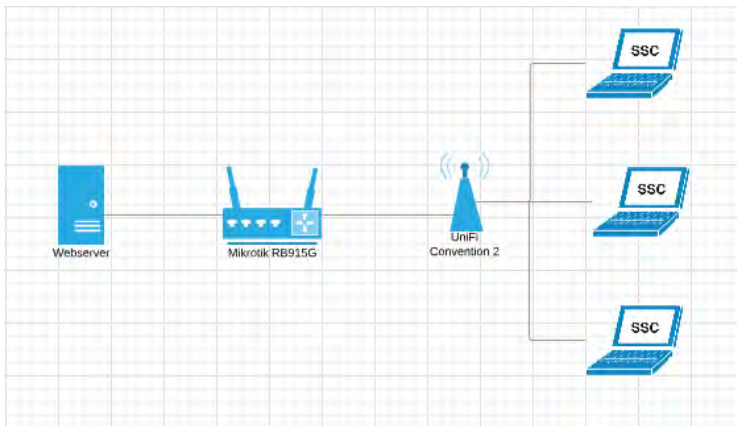
- Aplikasi dapat diakses melalui *web server*
- Aplikasi memiliki tampilan dan visualisasi yang menarik, karena akan diakses oleh banyak pengguna / pengunjung.

4.2 Desain Topologi dan Skenario

Pada tahap kali ini akan membahas tentang pembuatan desain topologi jaringan dan skema otentikasi untuk terkoneksi internet.

4.2.1 Desain Topologi Jaringan

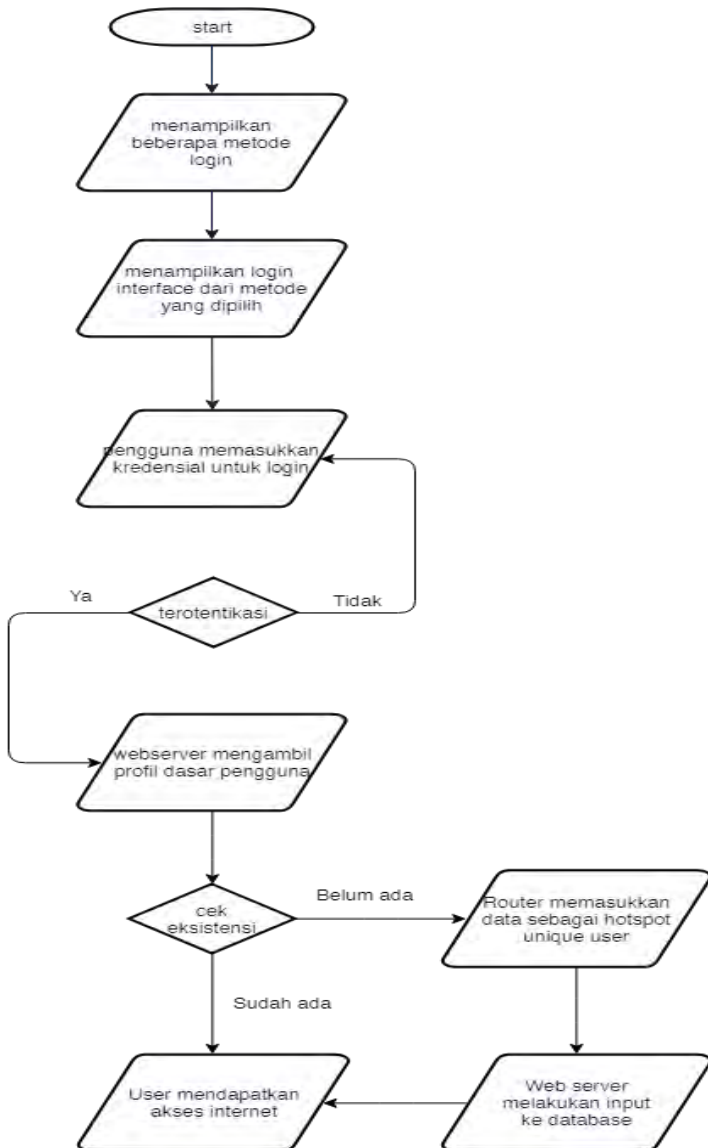
Pada tahap sebelumnya telah diketahui kebutuhan fungsional dan non-fungsional. Sehingga dalam tahap ini akan dirancang pengembangan aplikasi otentikasi *hotspot* menggunakan media sosial.



Gambar 4.1 Desain Topologi

menjelaskan bagaimana topologi jaringan untuk pengembangan aplikasi yang akan dibuat, terdiri dari satu buah *router*, satu buah *web server*, satu buah *access point* dan beberapa *device client* yang digunakan untuk melakukan uji coba perangkat lunak.

4.2.2 Perancangan Skenario Otentikasi *Hotspot*



Gambar 4.2 Skenario *Hotspot Authentication*

Dalam implementasi perangkat lunak ini sendiri dibutuhkan skenario untuk pengguna apabila hendak terkoneksi internet dengan melalui beberapa tahapan atau proses, mulai dari proses otentikasi *social network service provider* hingga pengguna dapat terkoneksi dengan internet.

Dari gambar skenario tersebut, pada awalnya pengguna akan diberi opsi untuk memilih media sosial yang diinginkan untuk proses otentikasi lalu user akan diarahkan menuju halaman masuk dari media sosial yang dipilih, setelah user memasukkan kredensial, apabila berhasil terotentikasi, akan muncul layar persetujuan dimana pengguna bersedia untuk membagikan profil dasar mereka untuk diberikan kepada pihak ketiga. Setelah itu barulah SNSP melakukan *callback* ke *redirect url* yang telah ditentukan, dalam hal ini akan melakukan *callback* ke *web server* dengan membawa parameter profil dasar untuk dilakukan pencocokan apakah profil ini sudah tersedia atau belum, apabila belum, maka data tersebut akan dimasukkan kedalam database dan *router*, barulah setelah itu pengguna dapat terkoneksi ke internet.

4.3 Perancangan Perangkat Lunak

Pada tahap perancangan perangkat lunak, ada beberapa proses yang harus dilakukan, mulai dari pembuatan use case diagram, pembuatan hingga pembuatan sequence diagram.

4.3.1 *Framework* dan *Library*

Pada tahap rancang bangun, perlu adanya beberapa library yang harus digunakan. Library sendiri ini merupakan sekumpulan kelas dan fungsi yang telah dibuat untuk membantu pengembang lebih cepat dalam membangun program. Dalam pembangunan perangkat lunak ini juga akan menggunakan framework PHP bernama CodeIgniter.

1. Facebook OAuth Library

Untuk OAuth library dari Facebook, dibawah ini merupakan sumber library yang digunakan
<https://github.com/facebook/php-graph-sdk>

2. Twitter OAuth Library

Untuk OAuth library dari Twitter, dibawah ini merupakan sumber library yang digunakan
<https://github.com/abraham/twitteroauth>

3. Google OAuth Library

Untuk OAuth library dari Google, dibawah ini merupakan sumber library yang digunakan
<https://github.com/theiphleague/oauth2-google>

4. RouterOS API PHP Class

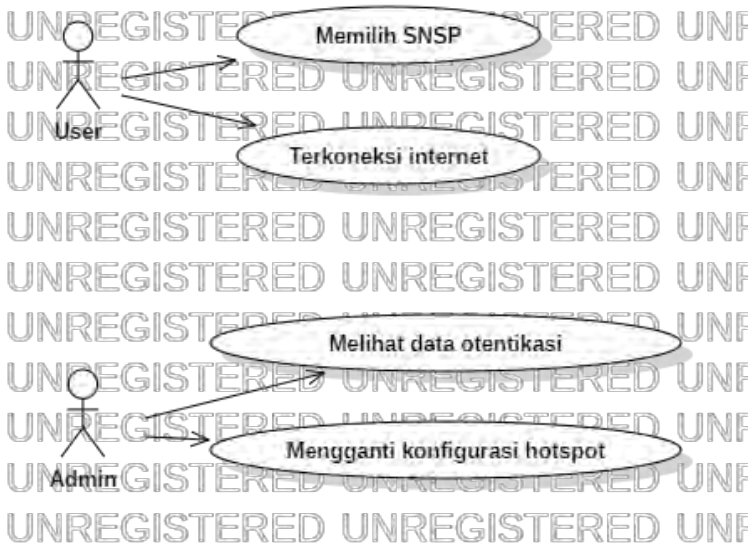
Untuk RouterOS API PHP Class, dibawah ini merupakan sumber library yang digunakan
<https://github.com/BenMenking/routeros-api>

5. CodeIgniter PHP Framework v3.0.8

CodeIgniter adalah framework pemrograman berbasis PHP yang menerapkan pengembangan dengan arsitektur MVC (*Model-View-Controller*). CodeIgniter paling sering dicatat untuk kecepatannya jika dibandingkan dengan framework PHP lainnya. Dari sekian banyak kerangka kerja PHP secara umum, pencipta PHP Rasmus Lerdorf berbicara di frOSCon pada bulan Agustus 2008, mencatat bahwa ia menyukai CodeIgniter "karena lebih cepat, lebih ringan dan paling tidak seperti kerangka kerja."

<https://codeigniter.com/download>

4.3.2 Use Case Diagram



Gambar 4.3 Use Case Diagram

Untuk use case sendiri, disini dibagi menjadi dua aktor, yaitu user / pengguna dan admin, dimana user dapat memilih social network provider yang diinginkan dan bisa terkoneksi dengan internet, dan untuk admin sendiri, dapat melihat data siapa saja yang telah terotentikasi dan mengganti konfigurasi *hotspot* seperti IP, DNS name, user, dan kata sandi agar dapat terkoneksi dengan *router*, karena itulah parameter yang dibutuhkan oleh API agar dapat terkoneksi dengan *router*.

Tabel 4.2 *Use Case Description* Otentikasi Media Sosial

Nama Usecase	UCD.01.04 Otentikasi Media Sosial
Aktor	Pengguna Wi-Fi
Deskripsi	Use case ini bermaksud agar aktor dapat melakukan otentikasi menggunakan social media
Precondition	Aktor telah masuk ke dalam jaringan Wi-Fi dan memilih tiga metode otentikasi media sosial
Pemicu	Aktor ingin melakukan otentikasi
Urutan Kejadian	<ol style="list-style-type: none"> 1. Aktor telah masuk ke dalam jaringan Wi-Fi 2. Aktor memilih salah satu dari tiga metode otentikasi 3. Aktor melakukan otentikasi dan akan di <i>redirect</i> ke laman aplikasi untuk mendapatkan sesi <i>browsing</i>
Langkah Alternatif	-
Kesimpulan	Use case ini berhasil apabila pengguna telah berhasil melakukan otentikasi dengan media sosial dan mendapatkan <i>browsing session</i>
Kondisi Setelah	Aktor dapat menggunakan internet
Aturan Proses	Aktor harus menyetujui adanya pengambilan data pengguna apabila aktor telah melakukan otentikasi
Asumsi	Aktor ingin menggunakan koneksi internet

Tabel 4.3 *Use Case Description* Keluar Media Sosial

Nama Usecase	UCD.01.04 Keluar Media Sosial
Aktor	Pengguna Wi-Fi
Deskripsi	Use case ini bermaksud agar aktor dapat melakukan <i>logout</i> pada media sosial mereka
Precondition	Aktor telah masuk ke dalam jaringan Wi-Fi dan terotentikasi serta mendapat <i>browsing session</i>
Pemicu	Aktor tidak ingin melanjutkan sesi <i>browsing</i>
Urutan Kejadian	<ol style="list-style-type: none"> 1. Aktor telah mendapatkan <i>browsing session</i> 2. Aktor harus mengakses tombol <i>logout</i> untuk keluar 3. Aktor akan diarahkan menuju halaman awal, yaitu pemilihan tiga metode otentikasi media sosial
Langkah Alternatif	-
Kesimpulan	Use case ini berhasil apabila pengguna tersebut tidak memiliki <i>session</i>
Kondisi Setelah	Aktor telah <i>logout</i>
Aturan Proses	Aktor harus sudah terotentikasi terlebih dahulu
Asumsi	Aktor tidak ingin melanjutkan sesi <i>browsing</i>

Tabel 4.4 Use Case Description Log Off Browsing Session

Nama Usecase	UCD.01.04 Log Off Sesi Browsing
Aktor	Pengguna Wi-Fi
Deskripsi	Use case ini bermaksud agar aktor dapat melakukan <i>log off browsing session</i>
Precondition	Aktor telah masuk ke dalam jaringan Wi-Fi dan terotentikasi serta mendapat <i>browsing session</i>
Pemicu	Aktor telah selesai melakukan sesi <i>browsing</i>
Urutan Kejadian	<ol style="list-style-type: none"> 1. Aktor telah mendapatkan <i>browsing session</i> 2. Aktor harus mengakses tombol <i>logoff</i> untuk keluar 3. Aktor akan diarahkan menuju halaman awal, yaitu pemilihan tiga metode otentikasi media sosial
Langkah Alternatif	-
Kesimpulan	Use case ini berhasil apabila pengguna tersebut tidak memiliki <i>session</i>
Kondisi Setelah	Aktor telah <i>logout</i>
Aturan Proses	Aktor harus sudah terotentikasi terlebih dahulu
Asumsi	Aktor telah selesai melakukan sesi <i>browsing</i>

Tabel 4.5 Use Case Description Update Account

Nama Usecase	UCD.01.04 <i>Update Akun</i>
Aktor	Admin
Deskripsi	Use case ini bermaksud agar aktor dapat melakukan pergantian username atau password
Precondition	Aktor telah berhasil <i>login</i>
Pemicu	Aktor ingin ada perubahan konfigurasi pada akun
Urutan Kejadian	<ol style="list-style-type: none"> 1. Aktor telah melakukan <i>login</i> 2. Aktor mengakses menu akun 3. Aktor melakukan pergantian <i>username</i> atau <i>password</i>
Langkah Alternatif	-
Kesimpulan	Use case ini berhasil apabila pengguna telah berhasil mengubah <i>password</i> dan ter <i>update</i> di database
Kondisi Setelah	Aktor di <i>redirect</i> di halaman yang sama
Aturan Proses	Aktor harus sudah terotentikasi terlebih dahulu
Asumsi	Aktor ingin mengganti <i>username</i> dan <i>password</i>

Tabel 4.6 Use Case Diagram Update Router Settings

Nama Usecase	UCD.01.04 Update Konfigurasi Router
Aktor	Admin
Deskripsi	Use case ini bermaksud agar aktor dapat melakukan perubahan konfigurasi pada <i>router</i>
Precondition	Aktor telah berhasil <i>login</i>
Pemicu	Aktor ingin mengganti konfigurasi pada <i>router</i>
Urutan Kejadian	<ol style="list-style-type: none"> 1. Aktor telah melakukan <i>login</i> 2. Aktor mengakses menu konfigurasi <i>router</i> 3. Aktor melakukan pembaruan parameter konfigurasi <i>router</i>
Langkah Alternatif	-
Kesimpulan	Use case ini berhasil apabila pengguna telah berhasil mengubah konfigurasi dan <i>update</i> di database
Kondisi Setelah	Aktor di <i>redirect</i> di halaman yang sama
Aturan Proses	Aktor harus sudah terotentikasi terlebih dahulu
Asumsi	Aktor ingin mengganti konfigurasi <i>router</i>

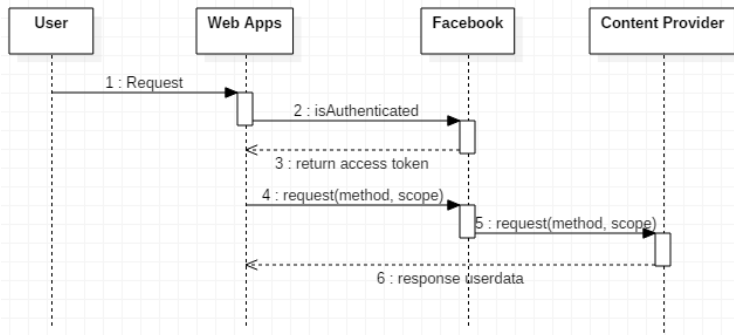
Tabel 4.7 Use Case Diagram View Captured User Data

Nama Usecase	UCD.01.04 <i>View Captured User Data</i>
Aktor	Admin
Deskripsi	Use case ini bermaksud agar aktor dapat melihat hasil data pengguna yang <i>ter-captured</i> dalam database
Precondition	Aktor telah berhasil <i>login</i>
Pemicu	Aktor ingin melihat hasil <i>captured user data</i>
Urutan Kejadian	<ol style="list-style-type: none"> 1. Aktor telah melakukan <i>login</i> 2. Aktor mengakses menu <i>visitor's data</i> 3. Aktor melihat hasil <i>capture data</i> pengguna yang terotentikasi
Langkah Alternatif	-
Kesimpulan	Use case ini berhasil apabila pengguna telah berhasil memperlihatkan hasil <i>captured user data</i>
Kondisi Setelah	Aktor dapat melakukan <i>searching</i> dan <i>sorting data</i>
Aturan Proses	Aktor harus sudah terotentikasi terlebih dahulu
Asumsi	Aktor ingin melihat data pengguna yang telah <i>ter-captured</i>

4.3.3 Sequence Diagram

Dalam tahap pembangunan perangkat lunak, dibutuhkan pembuatan sequence diagram untuk membantu memahami alur dalam melakukan pembuatan kode program nantinya.

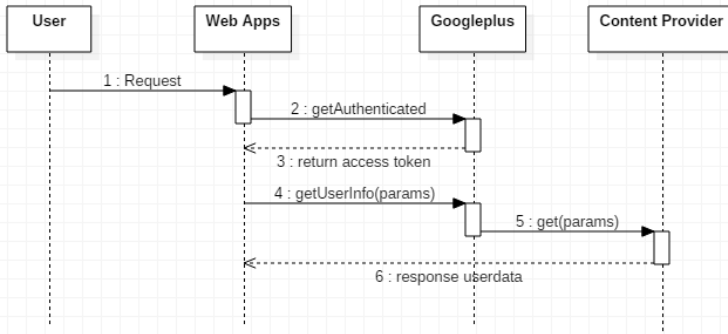
4.3.3.1 Facebook OAuth Sequence Diagram



Gambar 4.4 Facebook OAuth Sequence Diagram

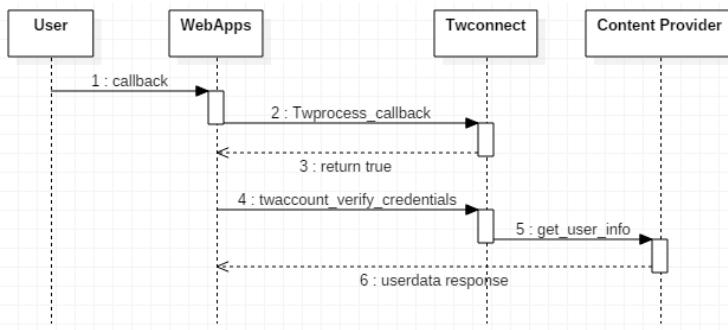
Gambar diatas merupakan sequence diagram yang terjadi apabila pengguna memilih Facebook sebagai SNSP untuk autentikasi media sosial. User melakukan request dan web apps melakukan redirecting ke Get OAuth dialog untuk memunculkan dialog persetujuan, dan pada tahap ini, di proses belakang, apabila pengguna menyetujui kebijakan dan ketentuan yang ditetapkan oleh aplikasi pihak ketiga, maka user akan di redirect kembali sesuai dengan URL yang ditetapkan oleh aplikasi pihak ketiga dengan membawa token akses. Token akses ini yang digunakan untuk melakukan akses beberapa API yang telah ditentukan, dalam hal ini aplikasi pihak ketiga meminta user data dari pengguna yang telah terotentikasi, dan memberikan response JSON yang berisi profil dasar pengguna.

4.3.3.2 Google OAuth Sequence Diagram



Berdasarkan instrumen pengumpulan data telah dirancang, langkah selanjutnya dilakukan pemetaan untuk masing-masing instrumen tersebut kepada target narasumber yang tepat. Pemetaan narasumber dilakukan untuk instrumen protokol wawancara, dimana dengan dilakukannya pemetaan narasumber yang tepat dapat menghasilkan informasi yang akurat dan tepat sasaran.

4.3.3.3 Twitter OAuth Sequence Diagram

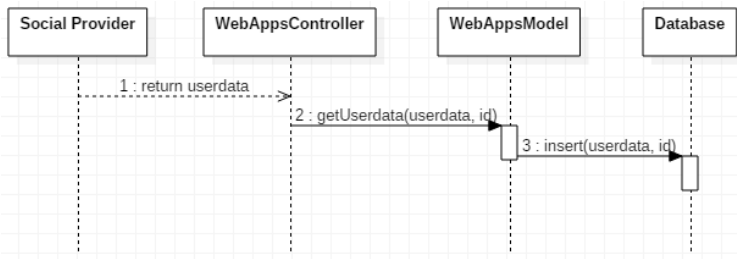


Gambar 4.5 Twitter OAuth Sequence Diagram

Diatas merupakan gambar sequence diagram ketika pengguna menggunakan twitter sebagai otentikasi media sosial. Di awal, pengguna melakukan request ke web apps, web apps melakukan

akses ke twitter OAuth dengan membawa parameter consumer key dan consumer secret yang didapatkan dari database dan dicocokkan dengan consumer id dan secret yang telah diberikan oleh twitter sebelumnya di halaman developer console. Setelah itu, melakukan request token dengan parameter callback url yang diambil dari database dan diinput di twitter developer console. Setelah itu akan muncul dialog otorisasi dan ketika user menyetujui syarat dan ketentuan, akan mendapatkan access token, dimana access token ini berfungsi agar pengembang dapat menggunakan API dari Facebook untuk mengakses resource dalam hal ini mengambil basic profile dari pengguna. Setelah mendapatkan access token, dan telah melakukan request ke twitter content provider, hasilnya adalah JSON user data.

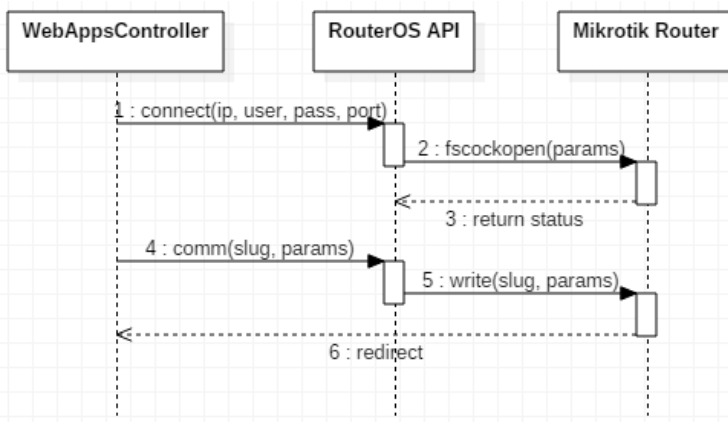
4.3.3.4 Memasukkan Data User



Gambar 4.6 Insert User Data Sequence Diagram

Gambar diatas merupakan sequence diagram setelah user melakukan otentikasi dan telah menerima user data dari social media provider, data pengguna yang telah didapatkan berbentuk JSON, dan pada aplikasi, JSON ini dirubah dalam bentuk array dan dipisahkan berdasarkan key, dan dari tiap key terdapat value. Setelah itu dilakukan pencocokan id di database, apabila id tersebut belum ada, maka akan dilakukan insert ke dalam database.

4.3.3.5 Router API Connection



Gambar 4.7 Router API Connection Sequence Diagram

Gambar diatas merupakan sequence diagram ketika aplikasi hendak berinteraksi dengan *router* melalui RouterOS API. Tujuan dalam melakukan koneksi dengan *router* adalah, untuk memberikan user yang menggunakan *hotspot*, akses internet, dan merekam user data yang dibutuhkan untuk melakukan otentikasi. Pada dasarnya pada saat melakukan koneksi. User data telah didapatkan sebelumnya dari proses otentikasi masing - masing media sosial. User data ini digunakan kembali untuk direkam didalam user database *hotspot router*, yang dibutuhkan hanya email yang dijadikan sebagai username user *hotspot*, dan social id yang bersifat unik, yang digunakan sebagai kata sandi. Untuk prosesnya, RouterOS API membutuhkan empat parameter yang digunakan agar aplikasi dapat terhubung dengan *router*, yaitu IP address, username *router*, kata sandi *router*, dan juga port, dalam hal ini port yang digunakan adalah port API yang secara default memiliki nilai 8728. Apabila telah terkoneksi barulah aplikasi memerintahkan RouterOS API untuk melakukan input data pengguna dalam user *hotspot*

router dengan membawa parameter slug (*ip/hotspot/user/add*), dan *params*(*user data*).

4.4 Penggunaan Modul RouterOS

Mikrotik memiliki banyak modul, seperti yang telah disebutkan pada bab tinjauan pustaka. Namun tidak semua fungsi tersebut akan dipergunakan pada tahap rancang bangun kali ini, berikut adalah beberapa modul yang nanti akan digunakan dalam tahap implementasi.

- 1. DHCP Client**

Fitur ini digunakan untuk memposisikan router sebagai client dari internet source ISP, sehingga router akan mendapatkan IP secara DHCP dari internet source ISP.

- 2. Address List**

Fitur ini digunakan untuk melihat list jaringan yang ada, dan juga bisa digunakan untuk menambahkan alamat IP baru, untuk jaringan yang berada di bawah *router*

- 3. DNS (Domain Name Server)**

Fitur ini digunakan untuk melakukan translasi alamat atau *domain name*.

- 4. NAT (Network Address Translation)**

Fitur ini digunakan agar jaringan local bisa terkoneksi dengan internet menggunakan IP publik yang dimiliki oleh ISP.

- 5. DHCP Server**

Fitur ini digunakan untuk mengatur pembagian IP untuk jaringan yang ada di bawah router, agar perangkat yang terkoneksi mendapatkan IP secara otomatis, sesuai dengan DHCP Pool yang disediakan

- 6. Hotspot**

Fitur ini digunakan untuk memberikan otentikasi pada pengguna

(halaman ini sengaja dikosongkan)

BAB V IMPLEMENTASI

Bab ini akan menjelaskan mengenai implementasi dari perancangan penelitian studi kasus dan hasil dari pengolahan data yang didapatkan dari wawancara kondisi terkini organisasi yang berisi latar belakang implementasi, tujuan implementasi, manfaat yang diperoleh dari implementasi, serta perubahan yang terjadi setelah implementasi aplikasi.

5.1 Lingkungan Implementasi

Aplikasi dikembangkan dengan menggunakan beberapa teknologi seperti editor, database, server, bahasa pemrograman, dan *library* yang disajikan dalam

Tabel 5.1 Daftar Teknologi Yang Digunakan Aplikasi

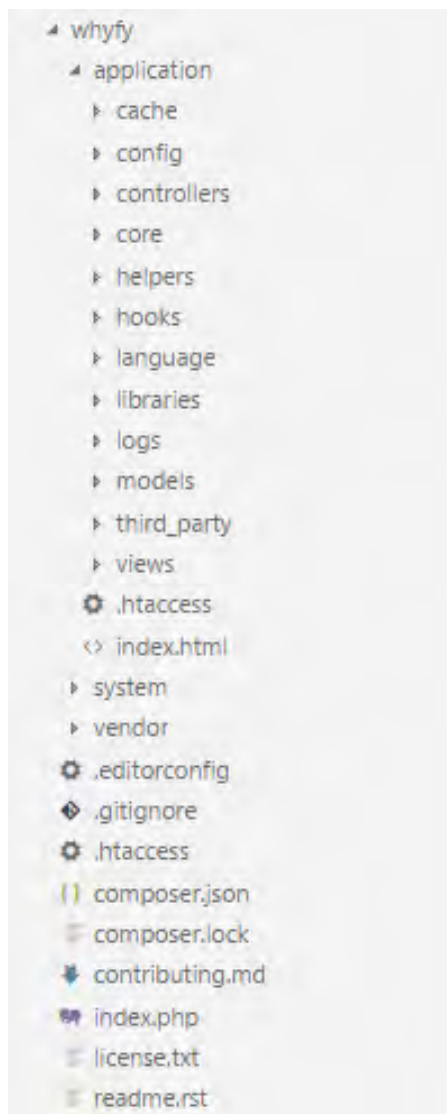
<i>Web server</i>	Apache
Bahasa Pemrograman	PHP
<i>Database</i>	MySQL
<i>Editor (IDE)</i>	Visual Code Editor
<i>Browser</i>	Google Chrome 56

<i>Library</i>	<ul style="list-style-type: none"> • Google OAuth API • Facebook OAuth API • Twitter OAuth API • MySQL • Session • RouterOS PHP Class API
-----------------------	---

5.2 Perancangan Aplikasi

Pada tahapan perancangan aplikasi ini dibuat dengan framework PHP bernama Codeigniter, dengan menggunakan design pattern MVC. Pada struktur folder Codeigniter, yang akan sering digunakan adalah pada folder *model*, *view*, dan *controller*. Masing – masing memiliki fungsi yang berbeda, *controller* berfungsi untuk mengatur logika dari aplikasi, *model* berfungsi sebagai penyedia data untuk *controller*, dalam arti lain, berperan sebagai perantara antara database dan juga *controller*. Lalu *view* berfungsi sebagai tampilan dari aplikasi.

Kemudian folder yang akan digunakan dalam tahapan implementasi ini yaitu *library* dan *config*. Folder *library* sendiri berisi kumpulan kelas – kelas yang didapatkan dari *vendor* lain agar developer dapat mengembangkan aplikasi dengan lebih cepat dan dapat berinteraksi dengan platform yang disediakan oleh *vendor* atau biasa disebut dengan API. Folder *config* berisi konfigurasi untuk framework ini sendiri atau juga bisa berkaitan untuk menyimpan konfigurasi yang digunakan oleh suatu *library* atau *helper*.



Gambar 5.1 Codeigniter *Folder Structure*

5.2.1 Konfigurasi Aplikasi

Pada bagian konfigurasi ini, menekankan pada konfigurasi parameter tiap – tiap social media provider.

5.2.1.1 Konfigurasi Google OAuth *Library*

```
$config['googleplus']['application_name'] = 'WhyFy';
$config['googleplus']['client_id'] =
'563349*****-
1bke1n58sr3ri0skh5rh6utjvbi6ks5c.apps.googleusercontent
.com';
$config['googleplus']['client_secret'] =
'G9xoHVNuy7AZy0Jnqel-****';
$config['googleplus']['redirect_uri'] =
'http://localhost/google';
$config['googleplus']['api_key'] =
'AIzaSyCs0gbDx9o46gSd6jy4eigV-*****';
$config['googleplus']['scopes'] =
array('https://www.googleapis.com/auth/userinfo.email',
'https://www.googleapis.com/auth/userinfo.profile',
'https://www.googleapis.com/auth/plus.login',
'https://www.googleapis.com/auth/plus.me');
```

Gambar 5.2 Google OAuth *Library* Configuration

Diatas merupakan konfigurasi yang diperlukan untuk menggunakan Google OAuth API. Ada beberapa elemen konfigurasi yang diperlukan untuk melakukan sambungan dengan API, yaitu *client_id*, *client_secret*, sebagai identitas dan menunjukkan kepemilikan terhadap aplikasi, *redirect_url* sebagai tujuan ketika melakukan *redirect* saat pengguna telah berhasil terotentikasi. Kunci API yang diperlukan untuk mengakses API yang diperlukan ketika telah mendapatkan token akses, dan yang terakhir adalah *scopes*, yang

menunjukkan otoritas aplikasi terhadap apa saja yang akan diakses dari data pengguna.

5.2.1.2 Konfigurasi Facebook OAuth *Library*

```
$config['facebook_app_id']           =
'456112588093021';
$config['facebook_app_secret']       =
'0c108809501b1f1087e6ff812b5c664a';
$config['facebook_login_type']       = 'web';
$config['facebook_login_redirect_url'] =
'facebookauth';
$config['facebook_logout_redirect_url'] =
'facebookauth/logout';
$config['facebook_permissions']       =
array('public_profile', 'email');
$config['facebook_graph_version']     = 'v2.6';
$config['facebook_auth_on_load']     = TRUE;
```

Gambar 5.3 Facebook OAuth *Library* Configuration

Diatas merupakan konfigurasi yang diperlukan untuk menggunakan Facebook OAuth API. Untuk konfigurasi Facebook API ada beberapa parameter yang diperlukan diantaranya, facebook *app_id* dan *app_secret* yang menunjukkan identitas dan ownership dari aplikasi yang nanti akan diakses oleh pengguna, login type menunjukkan akan diakses via apakah aplikasi ini, *redirect_url* menunjukkan *redirect* setelah pengguna telah melakukan otentikasi, kemudian ada permission, yaitu otoritas aplikasi ini terhadap data apa saja yang akan diakses.

5.2.1.3 Konfigurasi Facebook OAuth Library

```
'consumer_key'    => 'QswaJrtHe7Hqoi2fw1hMVEar2',
'consumer_secret' =>
'6xFychMW2Xtxyuiw1b3thAhIf06FOD48KbucfM6wjxcrp2eZNd',
'oauth_callback' => 'twitter/callback'
```

Gambar 5.4 Twitter OAuth Library Configuration

Diatas merupakan konfigurasi yang diperlukan untuk menggunakan Twitter OAuth API. Untuk konfigurasi Twitter API ada beberapa parameter yang diperlukan diantaranya, Twitter *consumer_key* dan *consumer_secret* yang menunjukkan identitas dan ownership dari aplikasi yang nanti akan diakses oleh pengguna, *oauth_callback* menunjukkan *redirect* setelah pengguna telah melakukan otentikasi.

5.2.2 Pembuatan Model

Model disini dibagi menjadi dua, yaitu *User_models* dan *Admin_models*, disesuaikan dengan use case dan struktur tabel.

5.2.2.1 User Models

- *Get Router Config*

```
public function getSetting(){
    return $this->db->query("SELECT * FROM
social_setting WHERE id='1241v52v54'");
}
```

Gambar 5.5 Get Router Config

Gambar diatas merupakan fungsi untuk mengambil konfigurasi *router* yang diperlukan. Adapun *basic configuration* yang diambil adalah *IP address router*, *port API*, *user name router*, *password router*.

- *Get User*

```
public function getUserList(){
    return $this->db->select('*')-
>from('user')->get();
}
```

Gambar 5.6 *Get Router Config*

Gambar diatas merupakan fungsi untuk melakukan *select* data secara keseluruhan pada tabel *user*.

- *Insert or Update User*

```
public function getUser($id, $data){
    $this->db->where('social_id',$id);
    $q = $this->db->get('user');

    if ( $q->num_rows() > 0 )
    {
        $this->db->where('social_id',$id);
        $data['login'] = date("Y-m-d H:i:s");
        $this->db->update('user',$data);
    } else {
        $data['login'] = date("Y-m-d H:i:s");
        $data['created'] = date("Y/m/d H:i:s");
        $this->db->set('social_id', $id);
        $this->db->insert('user',$data);
    }
}
```

Gambar 5.7 *Update or Insert User*

Gambar diatas merupakan fungsi untuk melakukan *insert* atau *update* pada *row user* sesuai dengan kondisi, apabila *user* tersebut sudah ada, maka akan dilakukan *update* pada *user*, apabila belum ada maka dilakukan *insert*

- **Matching User IP Address**

```
// Mendapatkan IP pengguna
public function getuserip($ip){
    $this->db->where('ip_address',$ip);
    $this->db->limit(1);
    $q = $this->db->get('session_code');

    if ( $q->num_rows() > 0 )
    {
        foreach($q->result() as $row):
            $data[] = $row;
        endforeach;
        return $row->ip_address;
    } else {
        return FALSE;
    }
}
```

Gambar 5.8 Matching IP Address

Function *getuserip()* berfungsi untuk melakukan pengecekan pada tabel *session_code* apakah ada *IP address* dengan *IP address* yang saat ini sedang digunakan oleh pengguna yang didapatkan dari *Controller Start*. Apabila ada *IP address* yang sama, maka akan dilakukan *return IP address* yang terdaftar pada database tabel *session_code*. Apabila tidak ada, maka akan dilakukan *return boolean* dengan nilai *FALSE*.

- **Matching User Session**

```
// Checking session apakah ada user dengan social_id
terkait telah melakukan otentikasi
public function checksession($id){
    $this->db->where('session_id', $id);
    $q = $this->db->get('session_code');

    if ( $q->num_rows() > 0 )
    {
        foreach($q->result() as $row):
            $data[] = $row;
        endforeach;
        if($row->status == 1){
            return TRUE;
        } else {
            return FALSE;
        }
    } else {
        return FALSE;
    }
}
```

Gambar 5.9 Matching User Session

Function checksession() berfungsi untuk melakukan pengecekan pada tabel `session_code` apakah ada `session_id` dengan `session_id` yang saat ini sedang digunakan oleh pengguna yang didapatkan dari *Controller Start*. Apabila ada `session_id` yang sama dan nilai statusnya adalah *TRUE*, maka akan dilakukan return *boolean* dengan nilai *TRUE*. Apabila tidak ada, maka akan dilakukan return *boolean* dengan nilai *FALSE*.

- *User Check Session Code*

```

public function checksessioncode($data){
    $this->db->where('code',$data['session_code']);
    $q = $this->db->get('session_code');

    if ( $q->num_rows() > 0 && $q->session_id ==
    $data['session_id'])
    {
        $this->db-
    >where('code',$data['session_code']);
        $this->db->set('status', 0);
        $this->db->update('session_code');
        return TRUE;
    }
    else {
        return FALSE;
    }
}

```

Gambar 5.10 *Check User Session*

Function *checksessioncode()* berfungsi untuk melakukan pengecekan pada tabel *session_code* apakah ada *session_id* dengan *session_id* yang saat ini sedang digunakan oleh pengguna yang didapatkan dari *Controller Start*. Apabila ada *session_id* yang sama, maka akan dilakukan *set value status* menjadi 0 dan memberikan kembalian *boolean* dengan nilai *TRUE* sedangkan apabila tidak ada, maka akan memberikan kembalian *boolean* dengan nilai *FALSE*.

- **Delete User Session**

```
// Melakukan flag session untuk menandakan session
tidak aktif
public function deletesession($code, $id){
    $this->db->where('code', $code);
    $q = $this->db->get('session_code');

    if ( $q->num_rows() > 0 )
    {
        $this->db->where('code', $code);
        $this->db->set('status', 0);
        $this->db->update('session_code');
    }
}
```

Gambar 5.11 Delete User Session

Function deletesession() berfungsi untuk melakukan pengecekan pada tabel *session_code* apakah ada *session_id* dengan *session_id* yang saat ini sedang digunakan oleh pengguna yang didapatkan dari *Controller Start*. Apabila ada *session_id* yang sama, maka akan dilakukan *set value status* menjadi 0.

5.2.2.2 Admin Models

- **Login Check**

```
public function checkLogin($username, $password){  
  
    $mikpass_en = md5($password);  
    $query_str = "SELECT * FROM admin WHERE username  
= ? AND password = ?";  
    $result = $this->db->query($query_str,  
array($username, $mikpass_en));  
  
    if ($result->num_rows() == 1) {  
        return $result;  
    }  
    else {  
        return false;  
    }  
}
```

Gambar 5.12 Check Login Function

Diatas merupakan fungsi untuk melakukan pengecekan pada database, apakah ada *username* dan *password* yang sesuai dengan parameter yang dimaksud pada tabel admin.

- **Get User List**

```
public function getUserList(){  
    return $this->db->select('*')  
->from('user')  
->get();  
}
```

Gambar 5.13 Get User Function

Gambar diatas merupakan fungsi untuk melakukan *select* data user secara keseluruhan pada database.

- ***Update Admin***

```
public function updateUserSetting($id, $data){  
    $this->db->where('id_mik', $id);  
    $this->db->set($data);  
    $this->db->update('admin');  
}
```

Gambar 5.14 Update Admin

Diatas merupakan fungsi pada *admin model* untuk melakukan pembaruan pada *field* tabel admin, yaitu *username* dan *password* untuk masuk ke *admin panel*.

- **Update Router Setting**

```

public function updateLandingSetting($id, $data){
    $query = $this->db->query("SELECT * FROM
social_setting WHERE id = '$id'");
    $result = $query->result_array();
    $count = count($result);

    if($count == 0){
        return $this->db->
>insert('social_setting', $data);
    }

    elseif ($count == 1) {
        return $this->db->
>replace('social_setting', $data);
    }

    else {

    }
}

```

Gambar 5.15 Update Router Setting

Diatas merupakan fungsi pada *admin model* untuk melakukan pembaruan pada *field* *social_setting*, yang berisi konfigurasi untuk *router* yaitu *IP*, *port*, *username*, dan *password*.

5.2.3 Pembuatan *Controller*

Controller disini berperan sebagai logika dari aplikasi. Untuk pembuatan *controller* sendiri dibagi menjadi lima, tiap – tiap otentikasi media sosial memiliki *controller* tersendiri, yaitu,

Twitter, Google, dan Facebookauth, sedangkan sisanya yaitu *Start Controller* dan *Admin Controller*.

5.2.3.1 Start Controller

Controller ini secara umum berfungsi untuk melakukan pengecekan sesi pengguna, apabila memiliki sesi akan menampilkan data pengguna pada sesi yang tersimpan pada *view landing.php*, dan apabila tidak ada sesi tersimpan, maka akan menampilkan *view auth.php* dimana *view* tersebut berisi tiga metode otentikasi media sosial.

- **Function construct**

```
function __construct () {  
    parent::__construct();  
    $this->load->model('user_models');  
    $this->load->helper('url');  
}
```

Gambar 5.16 Function Construct

Pada *function construct* ini, memanggil model yang sebelumnya telah dibuat yaitu *user_models*. Dan akan melakukan *loading helper* yang bernama *url*.

- **Function index**

```
public function index()
{
    if($this->session->userdata('login') != true){
        $contents['facebook_login_url'] = $this->facebook->login_url();
        $contents['google_login_url'] = $this->googleplus->loginURL();
        $this->load->view('auth', $contents);
    }
    else {
        . . . . .
    }
}
```

Gambar 5.17 Start Controller Function Index

Index function adalah fungsi yang secara default akan dijalankan pertama kali apabila mengakses pada suatu *controller*. Pada fungsi *index*, akan dilakukan pengecekan sesi pengguna, apabila memiliki tidak memiliki sesi, maka akan memanggil Facebook *library* untuk meminta tautan otentikasi Facebook, dan memanggil Google *library* untuk meminta tautan otentikasi Google, kemudian tautan tersebut dimuat pada *view auth.php*. Apabila ada sesi pengguna, maka akan membuat *array user_profile* yang berisi sesi pengguna, dan dimuat pada *view landing.php*.

```
$userdata = $this->session->userdata('user_profile');
$id = $userdata['social_id'];
$session = $this->user_models->checksession($id);
$ip = $this->getip();
$getip = $this->user_models->getuserip($ip);
$code = $this->random();
$data = array(
    'code' => $code,
);

if ($session == TRUE && $ip != $getip){
    $this->session->set_userdata('session_code',
$data);
    $this->user_models->postcode($id, $ip, $code);
    $this->sendcodemail($userdata, $code);
    redirect('start/inputcode');
} else {
...
}
```

Gambar 5.18 *Checking Session & Get IP Address*

Pada tahapan ini dilakukan pengecekan *social_id* dengan *userdata* pada *database* dan juga pengecekan *IP address* pengguna saat ini dengan yang ada pada *database*, apabila berbeda, maka sistem akan mengirim kode akses, yang menunjukkan bahwa ada *session* baru yang masuk menggunakan akun media sosial pengguna tersebut dengan alamat *IP* yang berbeda. Oleh karena itu, sistem ingin memastikan apakah itu benar dilakukan oleh pengguna, untuk lebih memberikan keamanan pada akun media sosial pengguna yang menggunakan akses internet.

- **Function Send Mail Code**

```

public function sendcodemail($userdata, $code){
    $config['protocol']    = 'smtp';
    $config['smtp_host']   =
'ssl://smtp.gmail.com';
    $config['smtp_port']   = '465';
    $config['smtp_timeout'] = '7';
    $config['smtp_user']   =
'codematters.dev@gmail.com';
    $config['smtp_pass']   = '*****';
    $config['charset']     = 'utf-8';
    $config['newline']     = "\r\n";
    $config['mailtype']    = 'text'; // or html
    $config['validation'] = TRUE; // bool whether
to validate email or not

    $this->email->initialize($config);
    $this->email->from('codematters.dev@gmail.com',
'CodeMatters');
    $this->email->to($userdata['email']);
    $this->email->subject('Hotspot Code');
    $this->email->message($code);
    $this->email->send();
    echo $this->email->print_debugger();
}

```

Gambar 5.19 Send Access Code via Email

Baris kode diatas merupakan fungsi untuk mengirim kode ke alamat email pengguna yang memiliki *double session* dan diakses dengan IP *address* yang berbeda.

- ***Mikrotik RouterOS Connect***

```
foreach ($this->user_models->getSetting()-
>result_array() as $row) {

if ($row != NULL) {
    $ip = $row['mik_ip'];
    $port_mik = $row['mik_port'];
    $user = $row['mik_username'];
    $password = $row['mik_password'];
    $user_profile = $row['user_profile'];
    }
}
```

Gambar 5.20 *Get Router Config*

Pada tahap selanjutnya, akan melakukan pengambilan data dari database dengan method `getSetting()` ke `user_model` yang berisi IP *router*, *port router*, *username router*, dan kata sandi dari *router*, beserta *user_profile*.

```

$API = new RouterosAPI();
$API->debug = true;
if ($API->connect($ip, $port_mik, $user, $password)) {
    $insert_user_router = $API-
>comm("/ip/hotspot/user/add", array(
        "name" => (string) 'g-'. $email,
        "profile" => (string) $user_profile,
        "comment" => (string) $name,
        "password" => (string) $code,
    ));
    if($insert_user_router){
        $contents['user_profile'] = $this-
>session->userdata('user_profile');
        $this->load->view('landing',$contents);
    }else{
        echo 'Error Save User';
    }
}
$API->disconnect();
}else{
    echo 'Error RouterosAPI';
}

```

Gambar 5.21 Start Controller Insert User Data to Mikrotik Hotspot

Setelah melakukan pengecekan user data ke database, melakukan koneksi ke *router* melalui RouterOS API PHP *Class* yang diinisiasi. Dalam melakukan koneksi ke *router*, dibutuhkan empat parameter, yaitu IP *address* dari *router*, *username router*, kata sandi *router*, dan *port router*.

Setelah *router* berhasil terhubung dengan aplikasi, selanjutnya adalah memasukkan data pengguna yang terotentikasi dalam *user hotspot* database yang ada pada Mikrotik. Ada beberapa

parameter dalam melakukan *insert user* pada Mikrotik, yaitu name, user_profile, comment, dan kata sandi. Untuk field name, mengambil dari email yang telah dimasukkan kedalam session, profile mengambil dari `getsettings()`, profile merupakan user profile yang menentukan otorisasi untuk tiap – tiap user yang telah dibahas pada sub bab sebelumnya, kemudian ada comment, hanya sebagai keterangan, namun disini, comment diambil dari nama yang didapatkan dari user data pengguna, dan yang terakhir adalah kata sandi, kata sandi ini diambil dari social id dari user data pengguna, karena social_id tidak mungkin sama untuk user satu dengan user yang lain.

- ***Function Get IP***

```
public function getip(){
    if(!empty($_SERVER['HTTP_CLIENT_IP'])){
        //ip from share internet
        $ip = $_SERVER['HTTP_CLIENT_IP'];
    }elseif(!empty($_SERVER['HTTP_X_FORWARDED_FOR']))
){
        //ip pass from proxy
        $ip = $_SERVER['HTTP_X_FORWARDED_FOR'];
    }else{
        $ip = $_SERVER['REMOTE_ADDR'];
    }
    return $ip;
}
```

Gambar 5.22 *Get Device IP Address*

Fungsi diatas digunakan untuk mengambil IP *address* local dari suatu device yang terkoneksi dengan jaringan, untuk kemudian akan dilakukan pengecekan pada proses *checking session* dan *checking IP*.

- **Function Random Number**

```
public function random(){
    $result = "";
    $chars =
"abcdefghijklmnopqrstuvwxyz0123456789$11";
    $len = 6;
    $charArray = str_split($chars);
    for($i = 0; $i < $len; $i++){
        $randItem = array_rand($charArray);
        $result .= "".$charArray[$randItem];
    }
    return $result;
}
```

Gambar 5.23 Generate Random Character

Fungsi diatas digunakan untuk melakukan *generate random character* yang akan digunakan sebagai unique code yang akan dikirimkan di *email* pengguna yang memiliki *multi session* dengan *IP address* yang berbeda beda.

- *Function Input Code*

```

public function inputcode(){
if($this->session->userdata('login') != true){
    redirect('start');
} else {
    if(isset($_POST['sessioncode'])){
        $userdata = $this->session-
>userdata('user_profile');
        $id = $userdata['social_id'];
        $data = array(
            'session_code' => $this->input-
>post('sessioncode'),
            'session_id' => $id
        );
        $checkcode = $this->user_models-
>checksessioncode($data);
        if($checkcode == TRUE){
            redirect('start');
        } else {
            // return view show error
            $this->session-
>set_flashdata('msg', 'Kode salah,
silahkan masukkan kode yang telah kami
 kirim via email');
            $this->load->view('input_code');
        }
    } else {
        $this->load->view('input_code');
    }
}
}

```

Gambar 5.24 *Function Input Unique Code*

Fungsi diatas berfungsi untuk melakukan pengecekan, apakah ada *input post 'sessioncode'* yang diterima dari pengiriman form pada *view*, apabila ada, maka akan melakukan pengecekan *sessioncode* yang diterima, kemudian dicocokkan ke database bersamaan dengan *session_id* yang ada pada user, apabila ada *sessioncode* dengan *session_id* yang sama dengan user, maka *sessioncode* tersebut benar dan akan di redirect ke controller start untuk kemudian mendapatkan user hotspot Mikrotik agar dapat melakukan akses internet.

5.2.3.2 Google Controller

- **Function index**

```
if($this->session->userdata('login') == true){  
    redirect('start');  
}
```

Gambar 5.25 Google Controller Function Index Check Session

Kode diatas berfungsi untuk melakukan pengecekan sesi, apabila memiliki sesi akan langsung diarahkan menuju *controller start*.

```

$this->googleplus->getAuthenticate();
$user = $this->googleplus->getuserinfo();
$id      = $user['id'];
$name    = $user['given_name'].' '.$user['family_name'];
$birthday = date("Y-m-d");

if (!isset($user['gender'])) {
    $gender = 'undefined';
} else {
    $gender = $user['gender'];
}

if (!isset($user['link'])) {
    $link = 'empty';
} else {
    $link = $user['link'];
}
$email    = $user['email'];
$picture  = $user['picture'];
$social_type = 'google-plus';

```

Gambar 5.26 Google Controller Authenticate and Get User Data

Setelah melakukan pengecekan, apabila tidak memiliki sesi akan dilanjutkan ke proses kode pada Gambar 5.26 yang mengarahkan pengguna ke halaman otentikasi Google kemudian variabel \$user akan menampung hasil response dari user data yang didapatkan dalam bentuk array.

Array tersebut kemudian ditampung ke masing – masing variable berdasarkan array key yang didapat. Terdapat beberapa pengecualian, yaitu pada *gender* dan *link* Google, karena kedua hal tersebut bukan merupakan required parameter yang harus diisi oleh pengguna dan terkadang kosong, sehingga apabila tidak terisi maka akan muncul error karena index key yang dimaksud tidak ditemukan pada array response yang

didapatkan, sehingga harus diberi kondisional untuk melakukan pengecekan array key tersebut.

```

$data = array(
    'social_id' => $id,
    'name' => $name,
    'email' => $email,
    'link' => $link,
    'gender' => $gender,
    'birthday' => $birthday,
    'picture' => $picture,
    'social_type' => $social_type,
);

$this->session->set_userdata('login',true);
$this->session->set_userdata('user_profile', $data);
$this->user_models->getUser($id, $data);
redirect('start');

```

Gambar 5.27 Google Controller Set User Data Session and Database Check

Kemudian, variabel dari masing – masing *key* ditampung kembali menjadi satu array, yaitu *social_id*, nama, link profile social media, alamat email, jenis kelamin, kemudian gambar / avatar. Setelah itu, akan dilakukan pengecekan pada database apakah ada *social_id* yang terdaftar, apabila tidak ada maka akan dilakukan insert, apabila sudah ada, maka tidak dilakukan tindakan lebih lanjut.

Kemudian tahap selanjutnya adalah menambahkan pengguna yang telah terotentikasikan tersebut kedalam session, dengan data – data seperti pada cuplikan barisan kode diatas dan kemudian sistem akan melakukan redirect ke *controllers start..*

- **Function logout**

```
public function logout(){
    $this->session->sess_destroy();
    $this->googleplus->revokeToken();
    redirect('start');
}
```

Gambar 5.28 Google Controller Logout Function

Function ini adalah function yang ada pada Google *controller*, gunanya yaitu untuk melakukan logout pada *session*, *unset* user data dari *session*, kemudian melakukan *revoke* Google token, dan melakukan *redirecting* kembali ke main *controller*.

5.2.3.3 Facebookauth Controller

- **Function index**

```
if($this->session->userdata('login') == true){
    redirect('start');
}
```

Gambar 5.29 Facebook Controller Function Index Check Session

Kode diatas berfungsi untuk melakukan pengecekan sesi, apabila memiliki sesi akan langsung diarahkan menuju *controller start*.

```

if ($this->facebook->is_authenticated())
{
$user = $this->facebook->request('get',
'/me?fields=name,first_name,last_name,email,link,gender
,locale,picture');
    if (!isset($user['error']))
    {
        $id          = $user['id'];
        $name        = $user['first_name'].'
        '.$user['last_name'];
        if (!isset($user['birthday'])) {
            $birthday = 'undefined';
        } else {
            $birthday = date("Y-m-d",
            strtotime($user['birthday']));
        }
        if (!isset($user['gender'])) {
            $gender    = 'undefined';
        } else {
            $gender = $user['gender'];
        }
        if (!isset($user['link'])) {
            $link      = 'empty';
        } else {
            $link = $user['link'];
        }
$email          = $user['email'];
$picture        = $user['picture']['data']['url'];
$social_type    = 'facebook';
}

```

Gambar 5.30 Facebook Controller Authenticate and Get User Data

Setelah melakukan pengecekan, apabila tidak memiliki sesi akan dilanjutkan ke proses kode pada Gambar 5.30 yang mengarahkan pengguna ke halaman otentikasi Facebook kemudian variabel \$user akan menampung hasil response dari user data yang didapatkan dalam bentuk array.

Array tersebut kemudian ditampung ke masing – masing variable berdasarkan array key yang didapat. Terdapat beberapa pengecualian, yaitu pada *gender*, *link*, dan *birthday* Facebook, karena ketiga hal tersebut bukan merupakan required parameter yang harus diisi oleh pengguna dan terkadang kosong, sehingga apabila tidak terisi maka akan muncul error karena index key yang dimaksud tidak ditemukan pada array response yang didapatkan, sehingga harus diberi kondisional untuk melakukan pengecekan array key tersebut.

```
$data = array(
    'social_id' => $id,
    'name' => $name,
    'email' => $email,
    'link' => $link,
    'gender' => $gender,
    'birthday' => $birthday,
    'picture' => $picture,
    'social_type' => $social_type,
);

$this->session->set_userdata('login', true);
$this->session->set_userdata('user_profile', $data);
$this->user_models->getUser($id, $data);
redirect('start');
```

Gambar 5.31 Facebook Controller Set User Data Session and Database Check

Kemudian, variabel dari masing – masing *key* ditampung kembali menjadi satu array, yaitu *social_id*, *nama*, *link profile social media*, *alamat email*, *jenis kelamin*, kemudian *gambar / avatar*. Setelah itu, akan dilakukan pengecekan pada database apakah ada *social_id* yang terdaftar, apabila tidak ada maka akan dilakukan insert, apabila sudah ada, maka tidak dilakukan tindakan lebih lanjut dan kemudian sistem akan melakukan redirect ke *controllers start..*

- **Function logout**

```
public function logout(){
    $this->session->sess_destroy();
    redirect('start');
}
```

Gambar 5.32 Twitter Controller Logout Function

Function ini adalah function yang ada pada *Twitter controller*, gunanya yaitu untuk melakukan logout pada *session*, *unset* user data dari *session*, kemudian melakukan *redirecting* kembali ke main *controller*.

5.2.3.4 Twitter Controller

- **Function index**

```
public function index()
{
    if($this->session->userdata('login') == true){
        redirect('start');
    }
    $this->load->view('auth');
}
```

Gambar 5.33 Twitter Controller Function Index Check Session

Kode diatas berfungsi untuk melakukan pengecekan sesi, apabila memiliki sesi akan langsung diarahkan menuju *controller start*.

- ***Function callback***

```
public function callback() {  
  
if($this->session->userdata('login') == true){  
    redirect('start');  
}  
  
$ok = $this->twconnect->twprocess_callback();  
    if ( $ok ) {  
        redirect('twitter/success');  
    }  
    else redirect ('twitter/failure');  
}
```

Gambar 5.34 Twitter Controller Callback Function

Callback function pada Twitter ini berfungsi untuk melakukan koneksi dengan API Twitter dengan *consumer id* dan *consumer secret* yang telah didapat untuk mendapatkan akses token sehingga bisa mengambil user data dari pengguna yang terotentikasi. Apabila berhasil terkoneksi dengan API Twitter, maka akan dilakukan pengalihan ke Twitter *controller* dengan fungsi *success* dan apabila gagal terkoneksi, akan dialihkan ke Twitter *controller* fungsi *failure*.

- **Function success**

```

$this->twconnect->twaccount_verify_credentials();
$user = $this->twconnect->tw_user_info;
$id = $user->id;
$name      = $user->name;
if (!isset($user->birthday)) {
    $birthday = 'undefined';
} else {
    $birthday = date("Y/m/d H:i:s");
}

if (!isset($user->gender)) {
    $gender = 'undefined';
} else {
    $gender = $user['gender'];
}

if (!isset($user->link)) {
    $link = 'https://twitter.com/' . $user->screen_name;
} else {
    $link = $user['link'];
}

$email      = 'undefined';
$picture    = $user->profile_image_url;
$social_type = 'twitter';

```

Gambar 5.35 Function Success Twitter Auth Get User Data

Setelah melakukan pengecekan, apabila proses *callback* berhasil, maka pengguna akan dialihkan ke halaman otentikasi

Twitter, kemudian variabel \$user akan menampung hasil response dari user data yang didapatkan dalam bentuk array.

Array tersebut kemudian ditampung ke masing – masing variable berdasarkan array key yang didapat. Terdapat beberapa pengecualian, yaitu pada *gender*, *link*, dan *birthday* Twitter, karena ketiga hal tersebut bukan merupakan required parameter yang harus diisi oleh pengguna dan terkadang kosong, sehingga apabila tidak terisi maka akan muncul error karena index key yang dimaksud tidak ditemukan pada array response yang didapatkan, sehingga harus diberi kondisional untuk melakukan pengecekan array key tersebut, untuk menyamakan dengan field yang ada pada database.

```

$data = array(
    'social_id' => $id,
    'name' => $name,
    'email' => $email,
    'link' => $link,
    'gender' => $gender,
    'birthday' => $birthday,
    'picture' => $picture,
    'social_type' => $social_type,
);

$this->session->set_userdata('login',true);
$this->session->set_userdata('user_profile', $data);
$this->user_models->getUser($id, $data);
redirect('start');

```

Gambar 5.36 Twitter Controller Set User Data Session and Database Check

Kemudian, variabel dari masing – masing *key* ditampung kembali menjadi satu array, yaitu *social_id*, *nama*, *link profile*

social media, alamat email, jenis kelamin, kemudian gambar / avatar. Setelah itu, akan dilakukan pengecekan pada database apakah ada `social_id` yang terdaftar, apabila tidak ada maka akan dilakukan insert, apabila sudah ada, maka tidak dilakukan tindakan lebih lanjut.

Kemudian tahap selanjutnya adalah menambahkan pengguna yang telah terotentikasitersebut kedalam session, dengan data – data seperti pada cuplikan barisan kode diatas dan kemudian sistem akan melakukan redirect ke *controllers start*.

- ***Function failure***

```
public function failure() {
    if($this->session->userdata('login') == true){
        redirect('start');
    }

    echo '<p>Twitter connect failed</p>';
    echo '<p><a href="' . base_url() .
    'twitter/logout">Try again!</a></p>';
}
```

Gambar 5.37 Twitter Controller Failure Function

Baris kode diatas merupakan fungsi *failure* yang melakukan handling apabila tidak berhasil terkoneksi dengan API Twitter.

- **Function logout**

```
public function logout(){
    $this->session->sess_destroy();
    $this->googleplus->revokeToken();
    redirect('start');
}
```

Gambar 5.38 Twitter Controller Logout Function

Function ini adalah function yang ada pada Twitter *controller*, gunanya yaitu untuk melakukan logout pada *session*, *unset* user data dari *session*, kemudian melakukan *redirecting* kembali ke main *controller*.

5.2.3.5 Admin Controller

- **Function construct**

```
function __construct () {
    parent::__construct();
    $this->load->helper('url');
    $this->load->model('admin_model');
}
```

Gambar 5.39 Admin Controller Function Construct

Diatas merupakan baris kode function construct, dimana pada baris fungsi diatas, melakukan load *Admin_models* karena pada *Admin Controller* ini akan mengatur konfigurasi tentang router dan untuk melihat visualisasi dari user data yang telah *ter-captured* dan *url helper*.

- **Function index**

```

public function index(){

    if(!$this->session->userdata('is_logged_in')) {
        $this->load->view('admin/login');
    }
    else {
        $this->load->view('admin/header');
        $data['count_fb'] = $this->admin_model-
>getFBUser();
        $data['count_g'] = $this->admin_model-
>getGuser();
        $data['count_tw'] = $this->admin_model-
>getTWuser();

        $this->load->view('admin/index', $data);
        $this->load->view('admin/footer');
    }
}

```

Gambar 5.40 Admin Controller Index Function

Index function pada gambar diatas melakukan pengecekan *session* di awal, apabila belum terotentikasi maka akan diarahkan ke *admin login view*, apabila telah terotentikasi, maka akan diarahkan ke view *admin/index.php* yang berisi cuplikan jumlah data pengguna yang terotentikasi secara keseluruhan dari masing – masing *social media provider*, yang sebelumnya telah melakukan pemanggilan data terlebih dahulu melalui *admin_models* dan di *passing* ke view *index*.

- **Function login**

```

public function loginAccount() {
    $username = $this->input->post('username');
    $password = $this->input->post('password');
    $login = $this->admin_model-
>checkLogin($username, $password);

    if (!$login) {
        redirect('Admin/index');
    }

    else {
        foreach ($login->result() as $sess) {
            $sess_data['is_logged_in'] = TRUE;
            $sess_data['username'] = $sess-
>mik_username;
            $this->session-
>set_userdata($sess_data);
        }
        redirect('Admin/index');
    }
}

```

Gambar 5.41 Admin Controller Login Function

Login function diatas berfungsi untuk melakukan pengecekan terhadap pengguna yang hendak melakukan otentikasi. Apabila *username* dan *password* yang dimasukkan ada pada database, maka akan dilakukan *redirect* ke *function* Admin/index, apabila tidak cocok maka juga akan dilakukan *redirect* ke Admin/index karena pada *function* Admin/index sudah memiliki *handling* untuk menangani dua *case* tersebut

- ***Function account***

```
public function account() {  
  
    if(!$this->session->userdata('is_logged_in')){  
        redirect('Admin/login');  
    }  
  
    else{  
        $this->load->view('admin/header');  
        $data['mik'] = $this->admin_model-  
>getUserSetting()->result_array();  
        $this->load->view('admin/accountform',  
$data);  
        $this->load->view('admin/footer');  
    }  
}
```

Gambar 5.42 Admin Controller Account Function

Fungsi diatas merupakan fungsi untuk menampilkan view untuk melakukan *update* pada akun, seperti penggantian *username* atau *password*. Data yang ditampilkan diambil dari tabel admin yang dipanggil melalui *admin_model* dengan function *getUserSetting()*.

- *Function update account*

```
public function updateAccount() {  
  
    if(!$this->session->userdata('is_logged_in')) {  
  
        redirect('Admin/login');  
    }  
  
    else {  
  
        $id = $this->input->post('id');  
        $username = $this->input->post('username');  
        $password = md5($this->input->  
>post('password'));  
        $data = array(  
            'id_mik' => $id,  
            'username' => $username,  
            'password' => $password  
        );  
        $this->admin_model->updateUserSetting($id,  
$data);  
        redirect('Admin/account');  
    }  
}
```

Gambar 5.43 Admin Controller Update Account Function

Fungsi diatas merupakan fungsi untuk melakukan *update* pada akun, dengan *field id, username, dan password* yang diberikan enkripsi *md5*. Kemudian ditampung dalam sebuah *\$data* array yang di *passing* ke *admin_model->updateUserSetting()* dengan parameter *\$id* dan *\$data*. Apabila sudah akan di *redirect* ke *Admin/account*.

- **Function settings**

```
public function settings() {  
  
    if(!$this->session->userdata('is_logged_in')){  
        redirect('Admin/login');  
    }  
  
    else{  
        $this->load->view('admin/header');  
        $data['setting'] = $this->admin_model->  
>getLandingSetting()->result_array();  
        $this->load->view('admin/form', $data);  
        $this->load->view('admin/footer');  
    }  
}
```

Gambar 5.44 Admin Controller Settings Function

Fungsi diatas digunakan untuk menampilkan halaman/form untuk konfigurasi *router*. Konfigurasi *router* ditampung dalam variabel array `$data['setting']` yang didapatkan dari `admin_model getLandingSetting()`, kemudian hasilnya akan di *passing* ke *view* `admin/form.php`.

- *Function update settings*

```
public function updateSettings() {  
  
    if(!$this->session->userdata('is_logged_in')) {  
        redirect('Admin/login');  
    }  
  
    else {  
        $id = $this->input->post('id');  
        $mikip = $this->input->post('ip');  
        $mikuser = $this->input->post('mik_user');  
        $mikpass = $this->input->post('mik_pass');  
        $dns = $this->input->post('dns');  
  
        $data = array(  
            'id' => $id,  
            'mik_ip' => $mikip,  
            'mik_username' => $mikuser,  
            'mik_password' => $mikpass,  
            'dns_name' => $dns  
        );  
  
        $this->admin_model->  
updateLandingSetting($id, $data);  
  
        redirect('Admin/settings');  
    }  
}
```

Gambar 5.45 Admin Controller Update Setting Function

Fungsi diatas merupakan fungsi untuk melakukan *update* pada konfigurasi *router*, dengan *field id, mik_ip, mik_user, mik_password, dan dns_name*. Kemudian ditampung dalam sebuah *\$data* array yang di *passing* ke *admin_model->updateLandingSetting()* dengan parameter *\$id* dan *\$data*. Apabila sudah akan di *redirect* ke *Admin/settings*.

- **Function visitor data**

```
public function visitor_data() {

    $this->load->view('admin/header');

    $data['user_list'] = $this->admin_model-
>getUserList()->result_array();
    $data['vdata'] = $this;

    $this->load->view('admin/datatables', $data);
    $this->load->view('admin/footer');
}
```

Gambar 5.46 Admin Controller View Visitor Function

Fungsi diatas digunakan untuk menampilkan halaman untuk melihat daftar pengguna yang telah terotentikasi. Data pengguna ditampung dalam variabel array *\$data['user_list']* yang didapatkan dari *admin_model getUserList()*, kemudian hasilnya akan di *passing* ke *view admin/datatables.php*.

- **Function logout**

```
public function logoutAccount(){  
  
    $this->session->sess_destroy();  
    redirect('Admin');  
}
```

Gambar 5.47 Admin Controller Logout Function

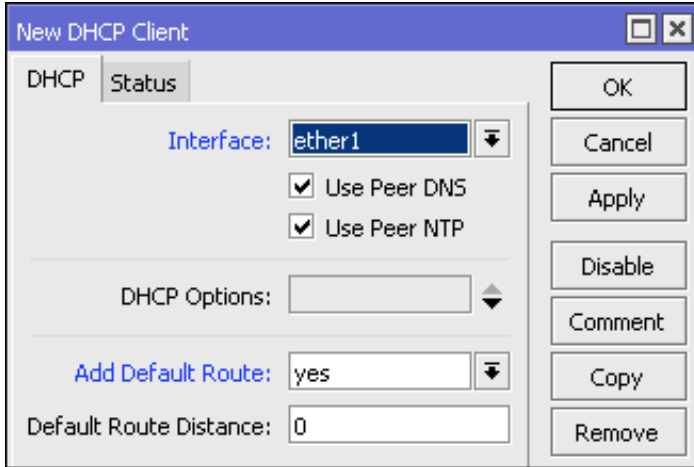
Function ini adalah function yang ada pada *Admin controller*, gunanya yaitu untuk melakukan logout pada *session*, *unset* user data dari *session*, kemudian melakukan *redirecting* kembali ke *Admin controller*.

5.3 Konfigurasi Router

Berikut merupakan konfigurasi untuk tiap – tiap modul pada *router* yang telah didefinisikan pada BAB 4.

5.3.1 Konfigurasi DHCP Client

Dalam percobaan kali ini, untuk mendapatkan IP Address secara otomatis dari ISP (Network Source), fitur yang digunakan adalah DHCP Client, untuk menggunakan fitur ini sendiri, dapat dilakukan pada menu **IP -> DHCP Client -> Add** pada Winbox.

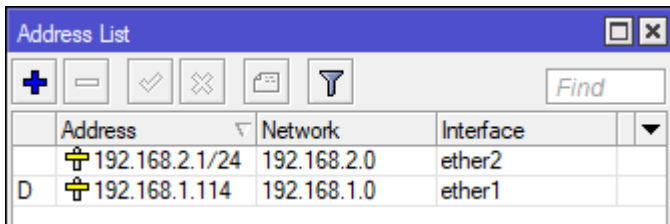


Gambar 5.48 Konfigurasi DHCP Client

Untuk pengaktifkan DHCP Client, definisikan parameter interface dengan interface yang terhubung ke DHCP Server, atau dalam kasus ini adalah interface yang terhubung ke ISP.

Karena ingin semua traffic ke internet menggunakan jalur koneksi dari ISP, maka Use-Peer-DNS=yes dan Add-Default-Route=yes.

5.3.2 Konfigurasi Address List



Gambar 5.49 Address List Configuration

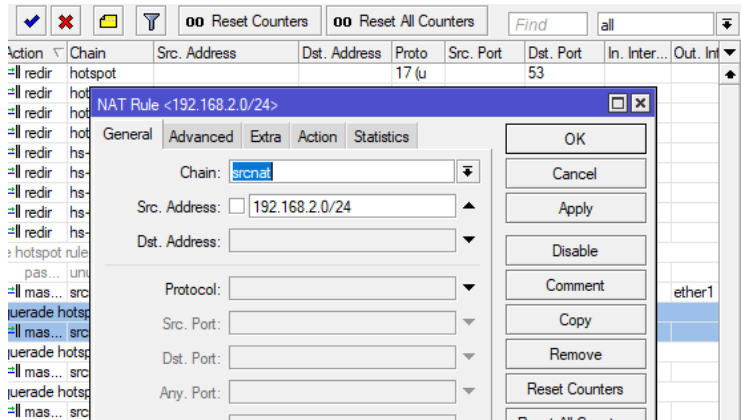
Gambar diatas merupakan konfigurasi Address list, dimana address list satu didapatkan dari hasil router sebagai DHCP

client dari modem ISP, sedangkan untuk ether dua adalah jaringan buatan yang ada di bawah router untuk pengguna nantinya. IP ether1 didapatkan dari DHCP client.

5.3.3 Konfigurasi NAT (*Network Address Translation*)

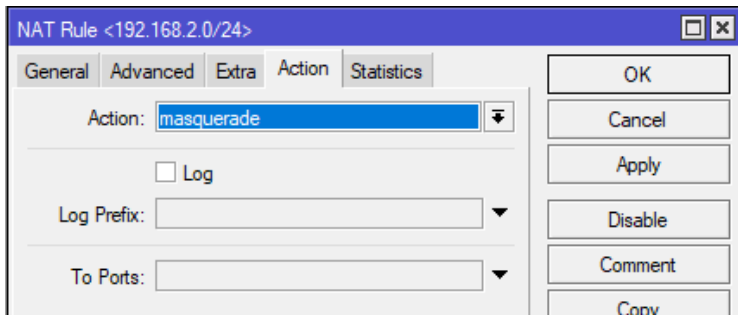
NAT(Network address translation) adalah suatu protokol yang digunakan mikrotik untuk mentranslasikan IP publik ke IP privat agar ip privat dapat tersambung dengan ip publik dalam penggunaan internet. dan lakukan hanya pada interface yang mengarah ke public. Tujuan NAT sendiri berfungsi dimana router Mikrotik yang tersambung dengan modem atau PC sumber internet yang akan mendapatkan IP publik dari internet serta pada PC guest/client yang tersambung dengan Mikrotik menggunakan IP Privat akan dapat mengakses internet. Dalam melakukan konfigurasi NAT dapat dilakukan pada menu **IP > Firewall > NAT**

1. Sekarang setting NAT, dengan cara buka **IP - Firewall - pada tab NAT** pilih tanda (+) untuk menambah konfigurasi NAT
2. Pilih chain **srcNAT** dan out interface pilih **ether1**, srcNAT merupakan source NAT, dimana network mendapatkan sumber internet dalam hal ini ethernet1 yang terkoneksi ke modem ISP



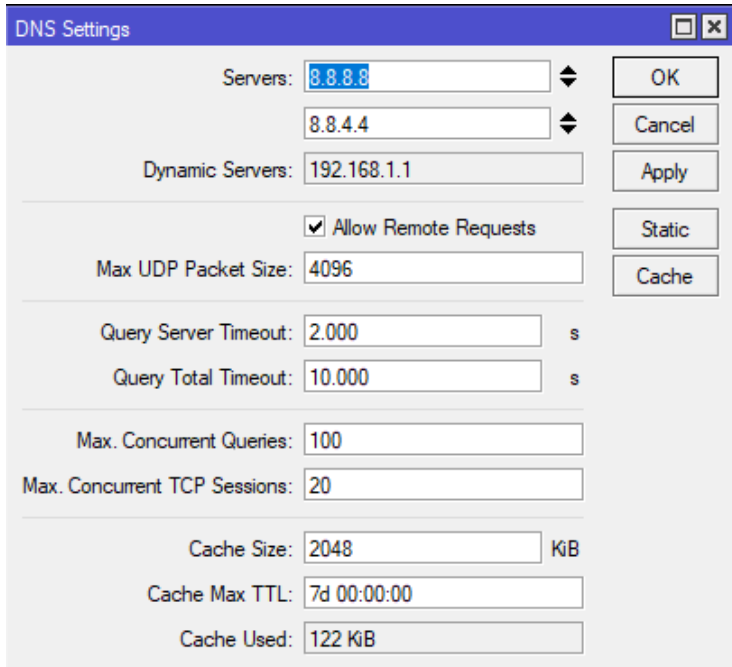
Gambar 5.50 NAT Configuration

3. Pada Tab Action pilih **masquerade** lalu **OK**. Masquerade ini sendiri berarti konfigurasi untuk "menyembunyikan" LAN pribadi 192.168.0.0/24 "di belakang" satu alamat 10.5.8.109 diberikan kepada client oleh ISP, maka dari itu harus menggunakan fitur source NAT (masquerading).



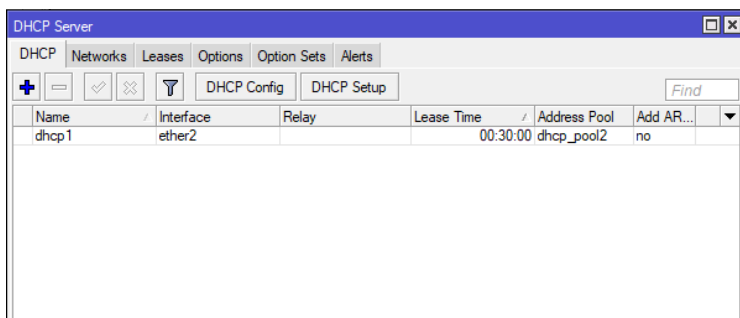
Gambar 5.51 NAT Rules

4. Setting DNS dengan pilih IP>DNS masukan IP dari server modem/ pc yang tersambung internet dan DNS google 8.8.8.8



Gambar 5.52 NAT DNS Setup

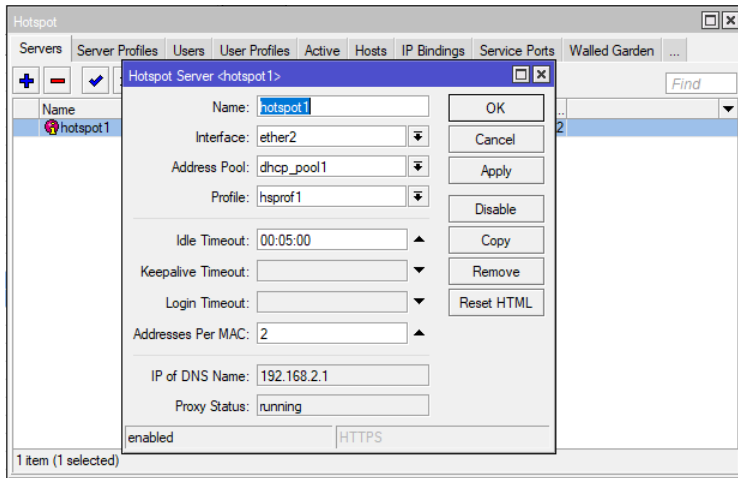
5.3.4 Konfigurasi DHCP Server



Gambar 5.53 DHCP Server Configuration

Untuk interface pilih ether2, karena ether2 yang terkoneksi ke jaringan lokal, kemudian untuk DHCP address akan secara default memilih address list yang telah ditetapkan pada ether2. Untuk gateway juga otomatis menggunakan gateway yang digunakan pada ether2 pada address list. Untuk dhcp pool akan memberikan address mulai dari 192.169.1.2 sampai dengan 192.168.1.254. Berarti ada 253 IP client yang tersedia. Untuk DNS server akan langsung otomatis terisi DNS yang telah ditentukan sebelumnya pada menu DNS yaitu 8.8.8.8. Lease time set menjadi 3d, artinya IP tersebut tersedia sampai waktu 3 hari apabila tidak digunakan.

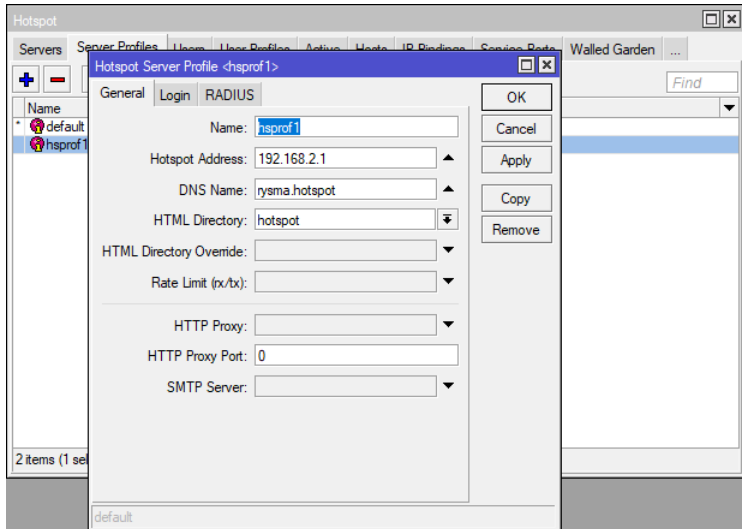
5.3.5 Konfigurasi Hotspot



Gambar 5.54 Hotspot Server Configuration

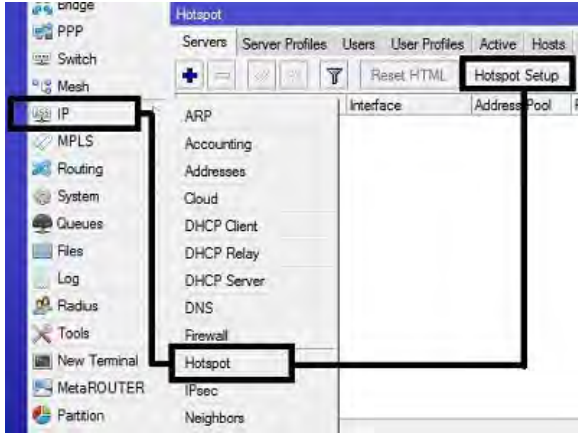
Dalam konfigurasi hotspot, pilih hotspot setup yang berada pada menu hotspot. Langkah pertama adalah menentukan interface yang akan digunakan sebagai hotspot, dalam case ini adalah wlan1. Kemudian langkah berikutnya adalah menentukan address yang akan digunakan, pada step ini akan

langsung terisi adalah address yang ada pada ether2, juga sama dengan address pool, akan menggunakan address list yang ada pada ether2, karena ether2 telah ditentukan sebagai wlan.



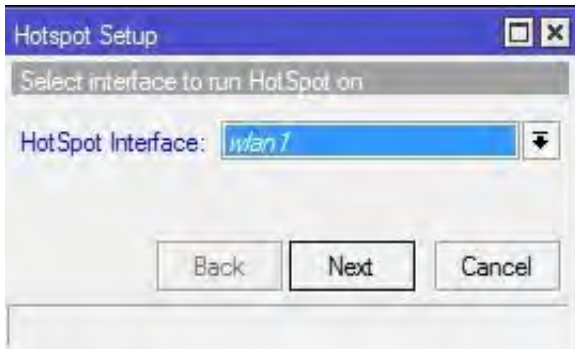
Gambar 5.55 Hotspot Server Profile Configuration

Buka di menu IP > Hotspot > Hotspot Setup.



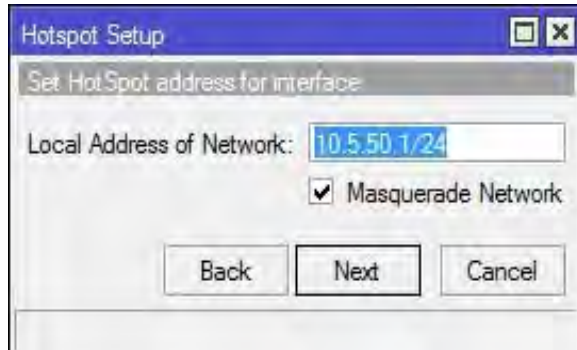
Gambar 5.56 Hotspot Configuration Wizard

Hotspot setup akan memunculkan wizard dialog yang menuntun dan memudahkan konfigurasi hotspot, terdapat beberapa langkah dalam wizard hotspot setup.



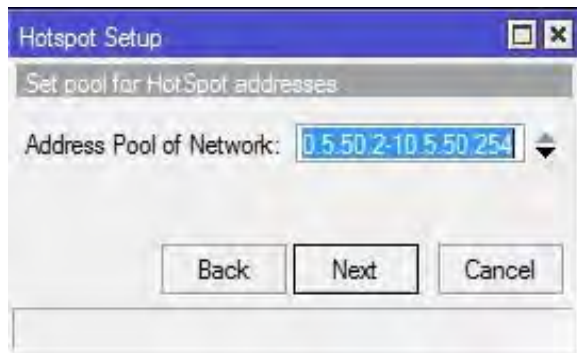
Gambar 5.57 Hotspot Interface Configuration

Langkah pertama, diminta untuk menentukan interface mana Hotspot akan diaktifkan. Pada kasus kali ini, Hotspot diaktifkan pada wlan1, dimana wlan1 sudah set sebagai access point (ap-bridge). Selanjutnya klik Next.



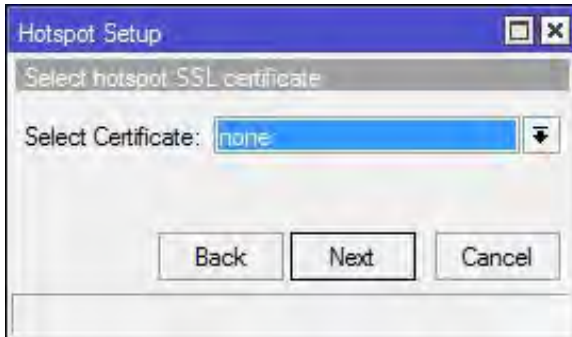
Gambar 5.58 Hotspot Local Address Configuration

Jika di interface wlan1 sudah terdapat IP, maka pada langkah kedua ini, secara otomatis terisi IP Address yang ada di wlan1. Tetapi jika belum terpasang IP, maka bisa menentukan IP nya di langkah ini. Kemudian klik Next.



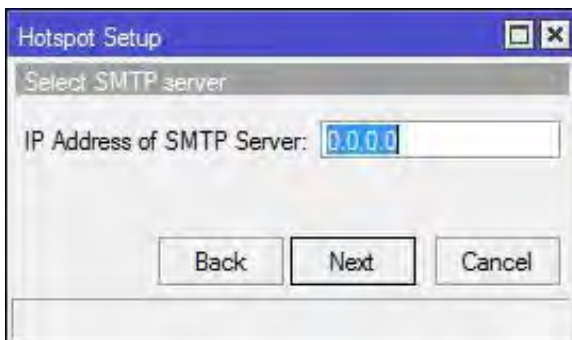
Gambar 5.59 Hotspot Address Pool Setup

Langkah ketiga, tentukan range IP Address yang akan diberikan ke user (DHCP Server). Secara default, router otomatis memberikan range IP sesuai dengan prefix/subnet IP yang ada di interface. Tetapi bisa merubahnya jika dibutuhkan. Lalu klik Next.



Gambar 5.60 Hotspot Certificate Configuration

Langkah selanjutnya, menentukan SSL Certificate jika akan menggunakan HTTPS untuk halaman loginnnya. Tetapi jika tidak memiliki sertifikat SSL, pilihl none, kemudian klik Next



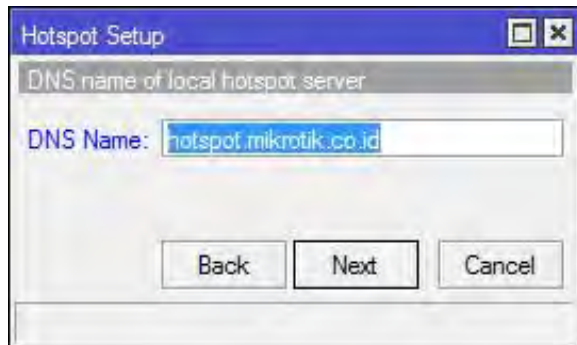
Gambar 5.61 Hotspot SMTP Server

Jika diperlukan SMTP Server khusus untuk server hotspot bisa ditentukan, sehingga setiap request SMTP client diredirect ke SMTP yang tentukan. Karena tidak disediakan smtp server, IP 0.0.0.0 maka biarkan default. Kemudian klik Next.



Gambar 5.62 Hotspot DNS Configuration

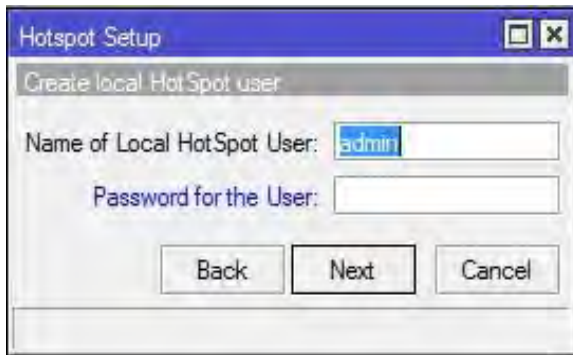
Di langkah ini, meentukan alamat DNS Server. DNS Server bisa isi dengan DNS yang diberikan oleh ISP atau dengan open DNS. Sebagai contoh, menggunakan DNS Server Google. Lalu klik Next.



Gambar 5.63 Hotspot DNS Name

Selanjutnya diminta memasukkan nama DNS untuk local hotspot server. Jika diisikan, nantinya setiap user yang belum melakukan login dan akan akses ke internet, maka browser akan di redirect ke halaman login ini. Disini DNS name sebaiknya menggunakan format FQDN yang benar. Jika tidak diisikan maka di halaman login akan menggunakan url IP address dari

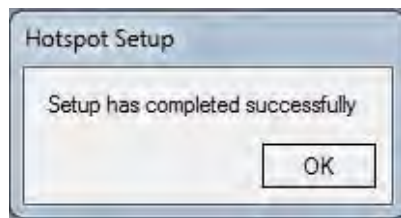
wlan1. Pada kasus ini, nama DNS-nya diisi "hotspot.mikrotik.co.id". Lalu klik Next.



Gambar 5.64 Hotspot User Configuration

Langkah terakhir, tentukan username dan password untuk login ke jaringan hotspot. Ini adalah username yang akan digunakan untuk mencoba jaringan hotspot .

Sampai pada langkah ini, jika di klik Next maka akan muncul pesan yang menyatakan bahwa *setting* Hotspot telah selesai.



Gambar 5.65 Hotspot Setup Wizard Finish

Selanjutnya akan mencoba melakukan koneksi pada laptop ke wifi hotspot yang sudah dibuat. Kemudian buka browser dan akses web apapun dan pastikan mengakses web yang

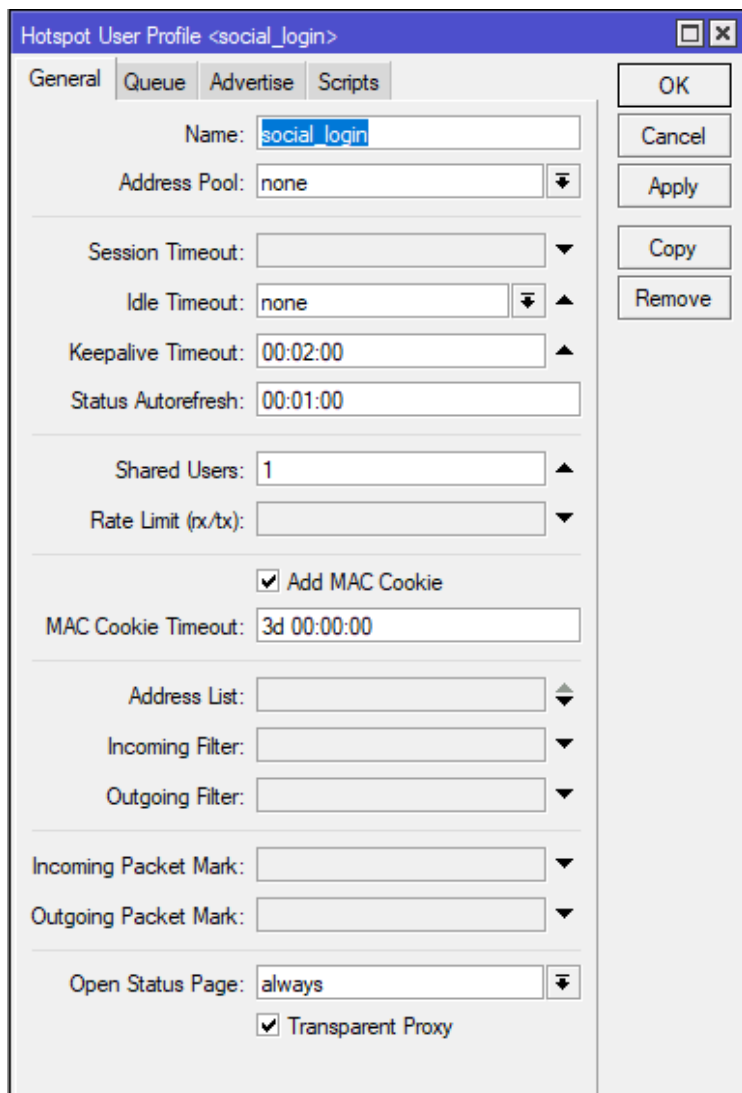
menggunakan protokol http, karena hotspot mikrotik belum mendukung untuk redirect web yang menggunakan https, maka akan dialihkan ke halaman login hotspot seperti pada gambar berikut ini:



Gambar 5.66 Default Login Page Hotspot Mikrotik

Jika berhasil login maka akan membuka halaman web yang diminta dan membuka popup halaman status Hotspot. Namun, hal ini masih belum seperti yang diinginkan, karena seharusnya terdapat redirect page tersendiri agar dapat melakukan autentikasi menggunakan media sosial, maka akan ada beberapa tahapan lagi yang harus dilakukan agar hotspot melakukan redirect ke landing page yang telah disiapkan untuk melakukan autentikasi via media sosial.

5.3.6 Konfigurasi *Hotspot User Profile*



The image shows a configuration window titled "Hotspot User Profile <social_login>". The window has a blue title bar with standard window controls (minimize, maximize, close). Below the title bar, there are four tabs: "General" (selected), "Queue", "Advertise", and "Scripts".

The "General" tab contains the following fields and controls:

- Name:** A text input field containing "social_login".
- Address Pool:** A dropdown menu set to "none".
- Session Timeout:** An empty dropdown menu.
- Idle Timeout:** A dropdown menu set to "none".
- Keepalive Timeout:** A text input field containing "00:02:00".
- Status Autorefresh:** A text input field containing "00:01:00".
- Shared Users:** A text input field containing "1".
- Rate Limit (x/tx):** An empty dropdown menu.
- Add MAC Cookie:** A checked checkbox.
- MAC Cookie Timeout:** A text input field containing "3d 00:00:00".
- Address List:** An empty dropdown menu.
- Incoming Filter:** An empty dropdown menu.
- Outgoing Filter:** An empty dropdown menu.
- Incoming Packet Mark:** An empty dropdown menu.
- Outgoing Packet Mark:** An empty dropdown menu.
- Open Status Page:** A dropdown menu set to "always".
- Transparent Proxy:** A checked checkbox.

On the right side of the window, there is a vertical stack of buttons: "OK", "Cancel", "Apply", "Copy", and "Remove".

Gambar 5.67 *Hotspot User Profile*

Langkah berikutnya adalah membuat user profile, user profile ini berguna untuk mengatur otorisasi pada pengguna yang memiliki profile tertentu. Untuk user profile akan diberi nama `social_login`, untuk seluruh pengguna yang melakukan autentikasi melalui media sosial. Dalam implementasinya, waktu untuk *browsing session* dibatasi maksimal 20 menit, apabila lebih dari itu, pengguna akan di *disconnect* dari jaringan dan router akan memunculkan *captive portal* dari Mikrotik untuk melakukan autentikasi kembali.

5.3.7 Konfigurasi *Hotspot Walled garden*

Dst. Host	Dst. Port	Hits	
*facebook.com		40	
*facebook.net		1	
*fbcdn.net		8	
*google.com		156	
*googleapis.com		50	
*gstatic.com		9	
*googleusercontent.com		0	
*google.co.id		106	
*googleapis.co.id		0	
*gstatic.co.id		0	
*googleusercontent.com		0	
*twitter.com		0	
*twimg.com		0	
*abs.twimg.com		0	
*localhost		0	
*its.ac.id		4	
*bootstrapcdn.com		97	
*jquery.com		26	
*akamaihd.net		0	

Gambar 5.68 *Hotspot Walled Garden Configuration*

Langkah berikutnya adalah memasukkan alamat yang ada pada list sebelumnya kedalam Mikrotik. Diatas merupakan list walled garden untuk tiap social media, tetapi dalam percobaan

ini hanya menggunakan Google, Facebook, dan Twitter sebagai metode otentikasi.

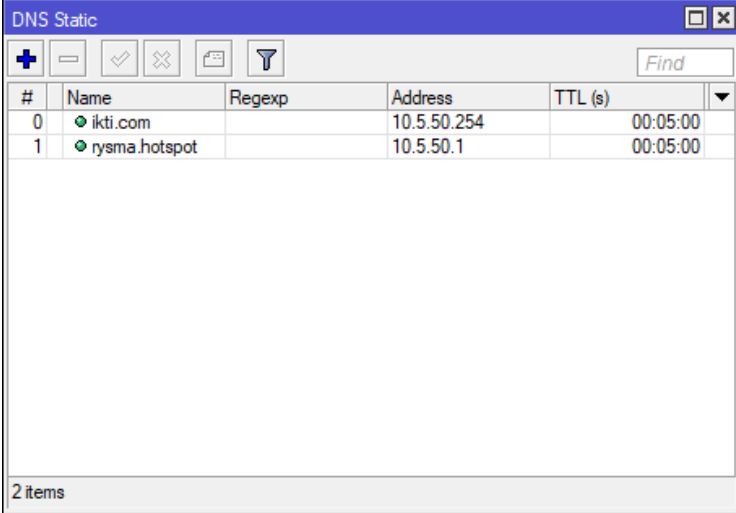
5.3.8 Konfigurasi *Hotspot Redirect*

```
<html>
<head>
<title>...</title>
<meta http-equiv="refresh" content="0; url=http://localhost">
<meta http-equiv="pragma" content="no-cache">
<meta http-equiv="expires" content="-1">
</head>
<body>
</body>
</html>
```

Gambar 5.69 Mikrotik *Hotspot Web Server Redirect*

Diatas merupakan tampilan login page yang telah dimodifikasi, ketika pengguna telah masuk kedalam captive portal mikrotik, maka akan langsung dilakukan redirect ke localhost, dan memberikan opsi untuk melakukan otentikasi dengan tiga metode.

5.3.9 Konfigurasi *Static DNS*



#	Name	Regexp	Address	TTL (s)
0	• ikti.com		10.5.50.254	00:05:00
1	• rysma.hotspot		10.5.50.1	00:05:00

2 items

Gambar 5.70 Konfigurasi *Static DNS*

Gambar diatas merupakan konfigurasi static DNS yang berfungsi untuk melakukan redirecting pada jaringan local apabila ada yang mengakses suatu DNS, maka akan dilakukan redirect ke *local server* jaringan tersebut. Hal ini bertujuan agar pengguna tidak melihat alamat IP pada saat masuk kedalam *captive portal* tetapi alamat IP tersebut sudah ditranslasikan menjadi alamat web.

5.4 Media Implementasi

Dalam melakukan implementasi sistem otentikasi hotspot ini, maka dibutuhkan tempat yang cocok sebagai media penerapan, dimana media ini terdiri dari beberapa kategori yang masing – masing memiliki karakteristik dan kebutuhan yang berbeda. Dalam melakukan implementasi, yang menjadi parameter utama media implementasi adalah luas area dan juga jumlah pengguna yang akan mengakses internet pada area tersebut,

semakin luas area dan semakin banyak pengguna, maka akan semakin banyak biaya yang dikeluarkan, dan juga sebaliknya. Oleh karena itu, pada tahap ini, akan dilakukan justifikasi untuk masing – masing media implementasi yang dibagi menjadi empat kategori.

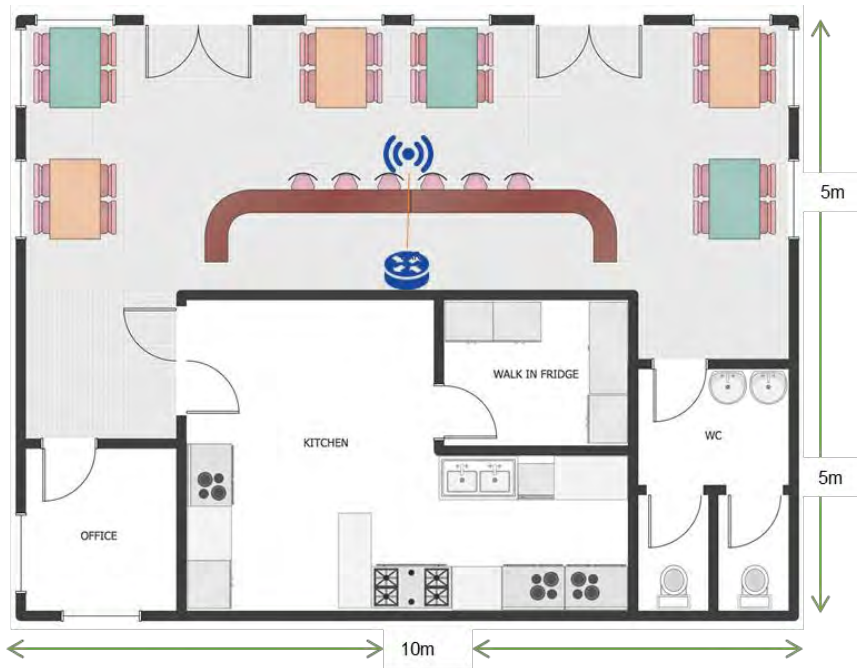
Tabel 5.2 Justifikasi Media Implementasi

Luas/User	0 - 100m ²	100 - 500m ²	500 - 5000m ²
1--25	Kategori 1	Kategori 1	
26 - 50	Kategori 2	Kategori 2	
51 - 100		Kategori 3	Kategori 3
100 - 200			Kategori 4

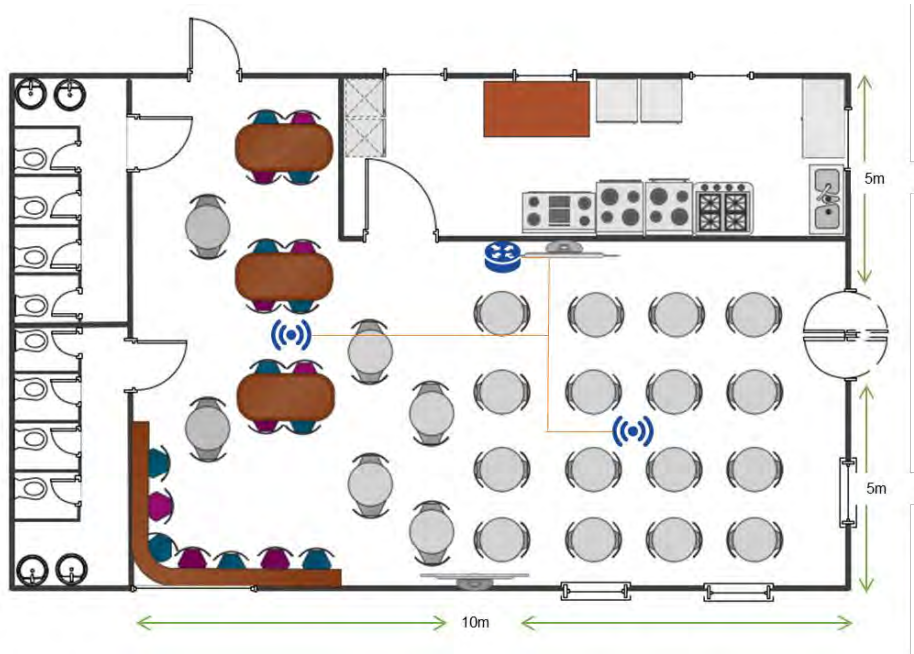
Dari tabel diatas, justifikasi yang dilakukan ini bertujuan untuk melakukan kategorisasi media implementasi dan membuat layouting topologi sesuai dengan lokasi lokasi umum untuk tiap – tiap kategori berdasarkan luas wilayah yang telah dipetakan, yaitu ada tiga range luas, mulai dari nol sampai dengan seribu meter persegi.

5.4.1 Layout Media Implementasi

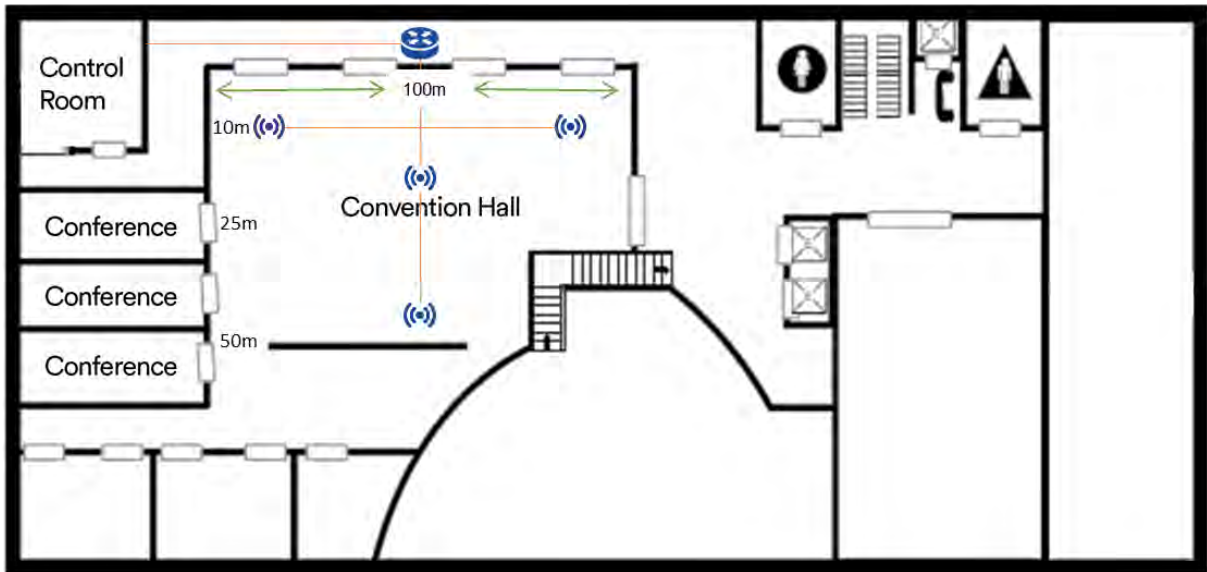
Masing – masing kategori memiliki range luas area 100 hingga 5000 meter persegi dengan jumlah pengguna kurang lebih sebanyak 25 – 200 orang. Lokasi yang cocok dengan kategori ini adalah café, restoran, co-working space untuk kategori satu dan dua, sedangkan untuk kategori tiga dan empat, yang cocok adalah auditorium dan juga convention hall.



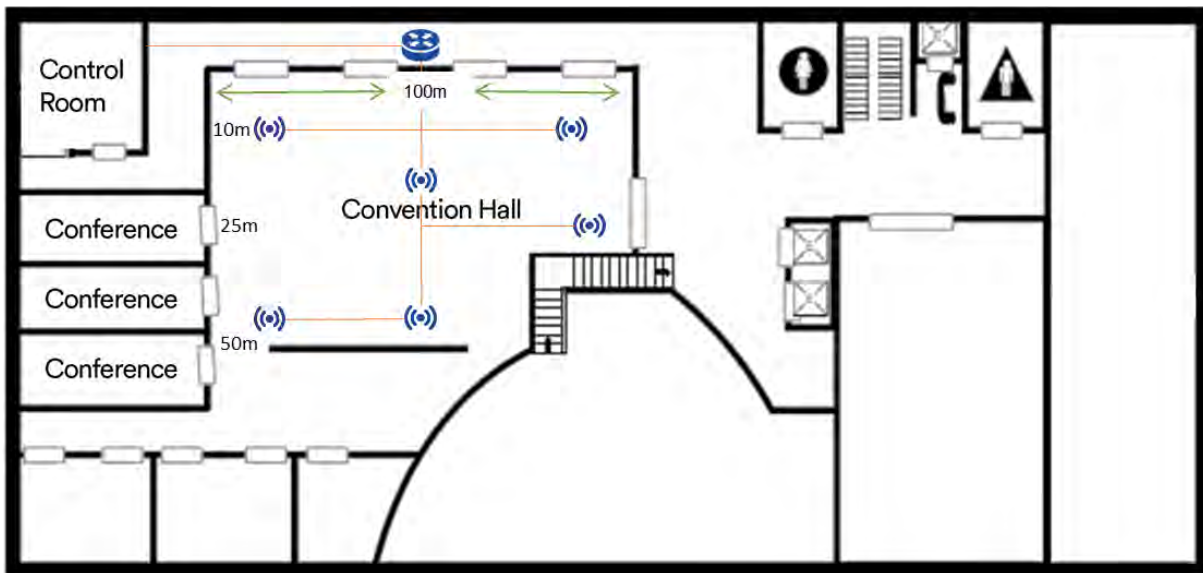
Gambar 5.71 Layout Small Cafe Kategori 1



Gambar 5.72 Layout Large Cafe Kategori 2



Gambar 5.73 Layout Convention Hall Kategori 3



Gambar 5.74 Layout Convention Hall Kategori 4

5.4.2 Kebutuhan Media Implementasi

Masing – masing kategori memiliki range luas area 100 hingga 5000 meter persegi dengan jumlah pengguna kurang lebih sebanyak 25 – 200 orang. Lokasi yang cocok dengan kategori ini adalah café, restoran, co-working space untuk kategori satu dan dua, sedangkan untuk kategori tiga dan empat, yang cocok adalah auditorium dan juga convention hall.

Tabel 5.3 Perincian Harga Kebutuhan

Barang	Satuan	Harga
Router Mikrotik RB 750	Pcs	Rp 530,000
Router Mikrotik RB2011iL	Pcs	Rp 1,350,000
Access Point Tenda AP4	Pcs	Rp 230,000
Access Point UniFi AP AC LR	Pcs	Rp 1,200,000
Kabel UTP Cat 5	m	Rp 1,300
RJ-45	Pcs	Rp 1,000
Pemasangan	Point	Rp 125,000

Diatas merupakan tabel estimasi harga untuk tiap – tiap perangkat yang dibutuhkan untuk implementasi aplikasi otentikasi hotspot menggunakan media sosial. Ada beberapa tipe item yang berbeda, seperti contohnya router pada tabel estimasi memiliki dua jenis, hal tersebut dikarenakan akan digunakan untuk kebutuhan dan medan area yang berbeda pula.

Tabel 5.4 Tabel Perencanaan Kebutuhan per Kategori

	Paket 1	Paket 2	Paket 3	Paket 4
Router	Mikrotik RB 750	Mikrotik RB 750	Mikrotik RB2011iL	Mikrotik RB2011iL
Access Point	Tenda AP4	Tenda AP4	UniFi UAP AC LR	UniFi UAP AC LR
Kabel UTP	20m	100m	400m	600m
RJ-45	4	8	16	24
Jumlah Access Point	1	2	4	6

Diatas merupakan tabel estimasi kebutuhan untuk tiap – tiap kategori berdasarkan layouting yang telah dibuat, perencanaan ini bisa berubah sewaktu waktu dan sangat fleksibel mengingat layout tempat untuk penerapan nantinya akan berbeda beda.

	Paket 1	Paket 2	Paket 3	Paket 4
Panjang	5 m	5 m	100 m	100 m
Lebar	10 m	10 m	50 m	50 m
Tinggi	3 m	3 m	10 m	10 m

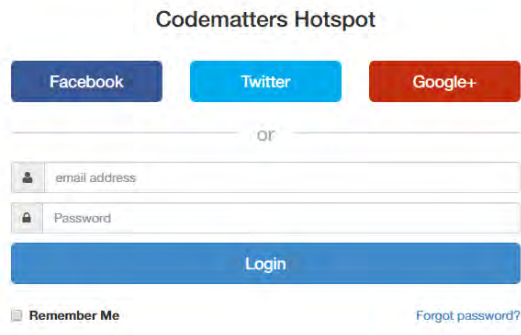
Pada gambar Gambar 5.71 diambil dari sumber [17]. Pada Gambar 5.72 diambil dari sumber [18] . Pada Gambar 5.73 dan Gambar 5.74 diambil dari sumber [19].

BAB VI HASIL DAN ANALISIS

Bab ini akan menjelaskan mengenai hasil dan pembahasan dari keseluruhan implementasi penelitian studi kasus yang ada pada bab sebelumnya. Berikut hasil analisa dan pembahasannya:

6.1 Aplikasi *Social Hotspot Authentication*

Berikut adalah hasil dari pembuatan aplikasi. Pada *splash screen* di awal, ketika pengguna terkoneksi dengan hotspot, akan diarahkan ke *web server* dengan tampilan seperti di bawah ini dengan opsi tiga metode otentikasi, yaitu Facebook, Google, dan juga Twitter.



Codematters Hotspot

Facebook Twitter Google+

or

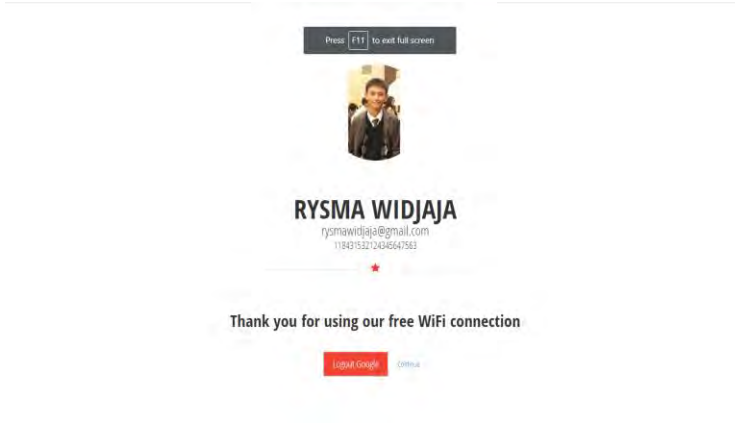
email address

Password

Login

Remember Me [Forgot password?](#)

Gambar 6.1 *Landing Page Web Server*



Gambar 6.2 Pengguna Terotentikasi

Setelah pengguna terotentikasi, maka akan muncul tampilan seperti diatas dengan menyertakan foto dan profil pengguna, kemudian akan diberikan dua opsi, yaitu untuk melanjutkan, dan juga logout. Apabila melanjutkan, maka pengguna tersebut akan terkoneksi dengan internet.

Welcome g-rysmawidjaja@gmail.com!

IP address:	10.5.50.254
bytes up/down:	5.8 KiB / 7.1 KiB
connected:	3m5s
status refresh:	1m

log off

Gambar 6.3 Authenticated Users

The screenshot shows the Mikrotik Hotspot interface with the 'Active' tab selected. The table below displays the active user information.

Server	User	Domain	Address	Uptime	Idle Time	Session Time ...	Rx Rq
::: Rysma Widjaja	hotspot1	f-rysmaaditya...	192.168.2.254	00:23:05	00:00:01		45.2 kb

Gambar 6.4 Active User Hotspot Mikrotik

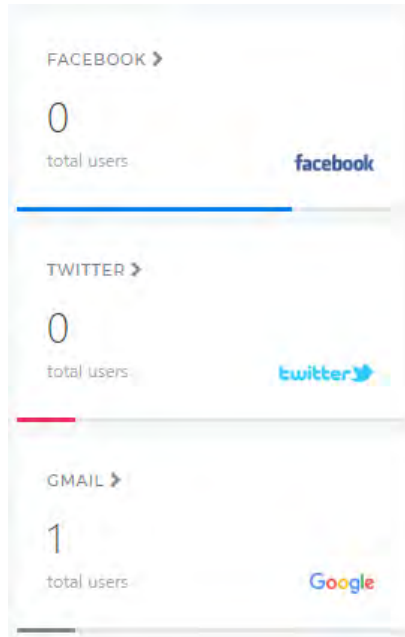
Diatas merupakan pengguna yang telah terotentikasi dan telah terdaftar ke dalam database user hotspot Mikrotik, dan ada pada tab active yang menunjukkan bahwa pengguna tersebut telah terkoneksi dengan internet.

The screenshot shows the Mikrotik Hotspot interface with the 'Users' tab selected. The table below displays the list of users.

Server	Name	Address	MAC Address	Profile	Uptime
::: counters and limits for trial users					
	admin			default	00:00:00
::: g-aditya14@mhs.its.ac.id	RYSMA ADITYA WIDJAJA			social_login	2d 01:36:24
::: okay					00:00:00
::: Rysma Widjaja	g-rysmawidjaja@gmail.co			social_login	00:00:00
::: Rysma Widjaja	g-rysmawidjaja@gmail.com			social_login	00:00:00
::: Rysma Widjaja	f-rysmaaditya@yahoo.com			social_login	00:02:23
::: Widjaja	t-undefined			social_login	00:00:00
	mimin			social_login	00:01:14

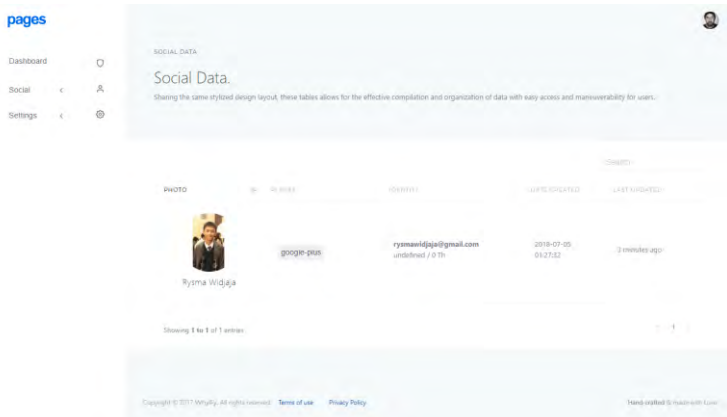
Gambar 6.5 Mikrotik Hotspot User List

Gambar diatas merupakan hasil dari pengguna yang telah terotentikasi, terdapat beberapa field yang menjadi informasi, yaitu server yang menunjukkan hotspot server, kemudian username yang berisi email pengguna yang telah diambil dan dimasukkan kedalam user hotspot Mikrotik. Lalu ada user_profile untuk tiap – tiap pengguna dan juga uptime.



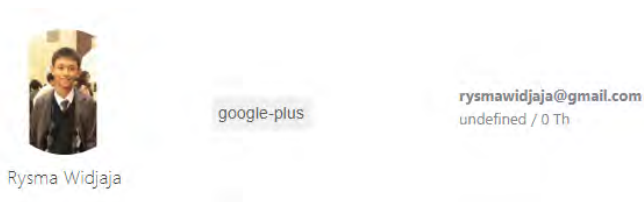
Gambar 6.6 Dashboard Total Pengguna per Media Sosial

Gambar diatas merupakan cuplikan dashboard yang berisi tentang banyaknya pengguna yang terotentikasi pada masing – masing media sosial. Terdapat tiga jenis media sosial yang digunakan untuk melakukan otentikasi yaitu Facebook, Google, dan Twitter.



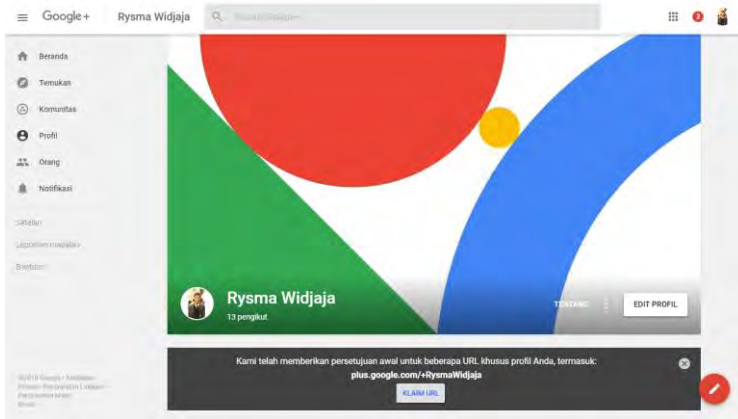
Gambar 6.7 Datatables Social User Data

Gambar diatas merupakan cuplikan dashboard yang berisi tentang *datatables* pengguna yang berhasil terotentikasi.



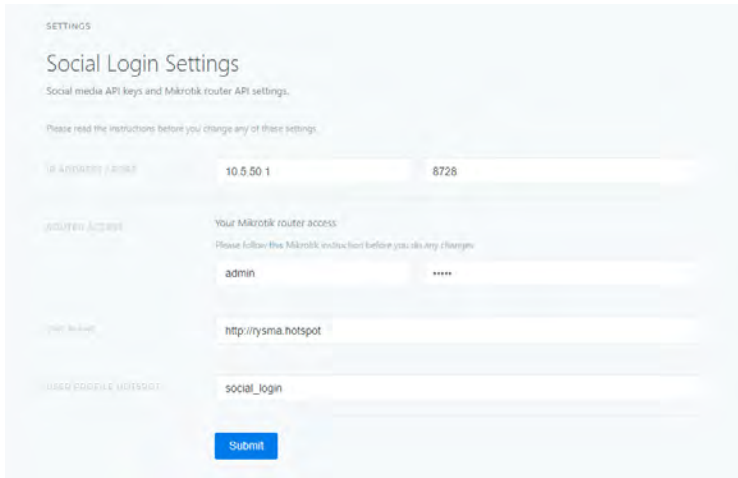
Gambar 6.8 Contoh User Data

Gambar diatas merupakan cuplikan pengguna yang telah terotentikasi dan telah masuk ke dalam database. Terdapat beberapa parameter yang bisa didapat, yaitu, nama, email, foto profil, dan tautan akun. Disitu terlihat gender yang bertuliskan *undefined* dan usia 0^{th} , itu berarti pengguna tersebut tidak menyertakan *gender* pada profilnya dan tidak menyantumkan tanggal lahir atau bisa juga karena masalah privasi yang diterapkan oleh pengguna, sehingga kedua data tersebut tidak bisa didapatkan. Link dari profil terdapat pada button yang bertuliskan ‘google-plus’, dan halaman tersebut akan menuju ke laman profil pengguna seperti pada gambar di bawah.



Gambar 6.9 Tautan Laman Profil Pengguna

Diatas merupakan contoh halaman profil pengguna yang tersimpan dalam tautan dari hasil pengambilan data pengguna yang terotentikasi.



Gambar 6.10 Update Konfigurasi Router

Gambar diatas merupakan gambar untuk melakukan update konfigurasi pada *router*. Konfigurasi yang di update adalah IP

address, username, password, DNS name, dan user_profile, hal – hal yang diperlukan sebagai parameter agar dapat terkoneksi dengan RouterOS API.

Your user access

Please follow [this Mikrotik instruction](#) before you do any changes

admin Password

Submit

Gambar 6.11 Update Konfigurasi Akun

Gambar diatas merupakan menu dashboard akun, dimana Admin sebagai aktor dapat melakukan pergantian username dan password.

social_id	name	email	1
91509	Rostanto	undefined	
58083	Widjaja	undefined	
20307	Sondra Launardo	sondra.launardo@il.com	
11843	Rysma Widjaja	rysmawidjaja@il.com	
24389	Rysma Widjaja	rysmawidjaja@il.com	

Gambar 6.12 Hasil User Hotspot Database

6.2 Pengujian Perangkat Lunak

Tabel 6.1 Hasil Pengujian Metode Otentikasi Pada Pengguna

No	Test Case	Expected Output	Hasil	Kesimpulan
1	Masuk ke laman otentikasi Google	Redirect ke url:/Start <i>profile view</i>	Redirect ke url:/Start dengan menampilkan profil Gplus pengguna	Berhasil
2	Masuk ke laman otentikasi Facebook	Redirect ke url:/Start <i>profile view</i>	Redirect ke url:/Start dengan menampilkan profil Facebook pengguna	Berhasil
3	Masuk ke laman otentikasi Twitter	Redirect ke url:/Start <i>profile view</i>	Redirect ke url:/Start dengan menampilkan profil Twitter pengguna	Berhasil
4	Logout otentikasi Google, Twitter, Facebook	Redirect ke url:/Start/ <i>auth view</i>	Redirect ke url:/Start halaman pemilihan metode otentikasi	Berhasil

Tabel 6.2 Hasil Pengujian Koneksi API ke Router

No	Test Case	Expected Output	Hasil	Kesimpulan
1	Koneksi dengan router menggunakan API url:/ikti.com/test dengan parameter yang benar	Muncul tulisan 'Connected' pada debug API	Muncul tulisan 'Connected' pada debug API	Berhasil
2	Koneksi dengan router menggunakan API url:/ikti.com/test/salah dengan parameter yang salah	Muncul tulisan 'Error RouterOS API'	Muncul tulisan 'Error RouterOS API'	Berhasil
3	Memasukkan data pengguna menggunakan API url:/ikti.com/test	Muncul tulisan 'Berhasil'	Muncul tulisan 'Berhasil'	Berhasil
4	Otentikasi hotspot via url dengan akun yang terdaftar yang benar url: / http://ryasma.hotspot/login?username=admin&password=admin	Redirect url:/ryasma.hotspot/status	Redirect url:/ryasma.hotspot/status	Berhasil
4	Otentikasi hotspot via url dengan akun yang terdaftar yang salah url: / http://ryasma.hotspot/login?username=mimin&password=momon	Redirect url:/ryasma.hotspot/login	Redirect url:/ryasma.hotspot/login	Berhasil

Tabel 6.3 Hasil Pengujian Otentikasi Halaman Admin

No	Test Case	Expected Output	Hasil	Kesimpulan
1	Masuk ke laman admin dengan <i>credentials</i> yang salah	Redirect url:/ikti.com/Admin login view	Redirect url:/ikti.com/Admin login view	Berhasil
2	Masuk ke laman admin dengan <i>credentials</i> yang benar	Redirect url:/ikti.com/Admin dashbord view	Redirect url:/ikti.com/Admin dashboard view	Berhasil

Tabel 6.4 Hasil Pengujian Form Update Konfigurasi Router dan Akun

No	Test Case	Expected Output	Hasil	Kesimpulan
1	Mengubah data konfigurasi router url:/ikti.com/Admin/settings	Redirect url:/ikti.com/Admin/settings dengan form value baru	Redirect url:/ikti.com/Admin/settings dengan form value baru	Berhasil
2	Mengubah data konfigurasi akun url:/ikti.com/Admin/account	Redirect url:/ikti.com/Admin/account dengan form value yang baru	Redirect url:/ikti.com/Admin/account dengan form value baru	Berhasil

6.3 Pembahasan

Aplikasi otentikasi hotspot menggunakan media sosial ini dibuat sesuai dengan skenario metode otentikasi pada bab desain skenario. Aplikasi ini secara umum merupakan modifikasi dari metode otentikasi hotspot standar, dengan menambahkan otentikasi media sosial di tengah prosesnya.

Aplikasi ini tidak hanya sekedar metode otentikasi biasa, namun juga memberikan nilai lebih bagi penyedia layanan *Wi-Fi* guna melakukan monitoring penggunaan dari layanan yang disediakan, yaitu terkait data pengguna yang *ter-captured* secara berkala yang dapat dimanfaatkan untuk menunjang keperluan bisnis, promosi misalnya.

Aplikasi ini juga sudah dilengkapi dengan dashboard untuk melihat data – data pengguna *Wi-Fi* beserta demografi nya.

6.4 Analisa Bisnis

6.4.1 Analisa Paket Bisnis Implementasi

Tabel 6.5 Bisnis Paket Implementasi Aplikasi

Luas/User	0 - 100m ²	100 - 500m ²	500 - 5000m ²
1--25	Paket 1	Paket 1	
26 - 50	Paket 2	Paket2	
51 - 100		Paket 3	Paket 3
100 - 200			paket 4

Tabel diatas merupakan paket – paket bisnis untuk implementasi aplikasi ini berdasarkan luas area dan juga jumlah pengguna. Dari segi luas area, membandingkan dengan luas *convention hall* rata – rata, yaitu kisaran 500 – 1000m², dengan kapasitas *full stand party* biasanya berkisar antara 500 – 800 orang, dengan asumsi yang akan menggunakan Wi-Fi tidak lebih dari 200 orang per koneksi. Dari analisa paket bisnis

diatas, kemudian dibuat tabel *pricing* dengan detail sebagai berikut.

Tabel 6.6 Perincian Harga Kebutuhan

Barang	Satuan	Harga
Router Mikrotik RB 750	Pcs	Rp 530,000
Router Mikrotik RB2011iL	Pcs	Rp 1,350,000
Access Point Tenda AP4	Pcs	Rp 230,000
Access Point UniFi AP AC LR	Pcs	Rp 1,200,000
Kabel UTP Cat 5	m	Rp 1,300
RJ-45	Pcs	Rp 1,000
Pemasangan	Point	Rp 125,000

Diatas merupakan tabel estimasi harga untuk tiap – tiap perangkat yang dibutuhkan untuk implementasi aplikasi otentikasi hotspot menggunakan media sosial. Ada beberapa tipe item yang berbeda, seperti contohnya router pada tabel estimasi memiliki dua jenis, hal tersebut dikarenakan akan digunakan untuk kebutuhan dan medan area yang berbeda pula.

6.4.2 Analisa Biaya Awal

Tabel 6.7 Tabel Perencanaan Kebutuhan per Paket

	Paket 1	Paket 2	Paket 3	Paket 4
Router	Mikrotik RB 750	Mikrotik RB 750	Mikrotik RB2011iL	Mikrotik RB2011iL
Access Point	Tenda AP4	Tenda AP4	UniFi UAP AC LR	UniFi UAP AC LR
Kabel UTP	20m	100m	400m	600m
RJ-45	4	8	16	24
Jumlah Access Point	1	2	4	6

Diatas merupakan daftar kebutuhan untuk tiap – tiap paket, tabel diatas merupakan estimasi best case untuk sebuah implementasi aplikasi ini. Terdapat perbedaan yang mendasar dari paket satu dan dua dengan paket tiga dan empat, yaitu luas area yang *di-cover* ditambah dengan pengguna yang dapat ditampung, paket satu dan dua cocok untuk diterapkan pada café ataupun selasar kampus yang luas area tidak melebihi 500m² dengan jumlah pengguna yang masih terhitung sedikit, sedangkan paket tiga dan empat cocok diterapkan pada *exhibition hall*, auditorium yang memiliki kapastas besar dan juga luas area yang besar pula.

Tabel 6.8 Tabel Estimasi Harga per Paket

	Paket 1	Paket 2	Paket 3	Paket 4
Router	530000	530000	1350000	1350000
Access Point	230000	230000	4800000	7200000
Kabel UTP	26000	130000	520000	780000
RJ-45	4000	8000	16000	24000
Biaya Pemasaan	125000	250000	500000	750000
Harga	915,000	1148,000	7,186,000	10,356,000

Diatas merupakan daftar total perencanaan biaya untuk masing – masing paket. Dari harga tersebut nantinya akan disesuaikan juga dengan kebutuhan *real* pada saat melakukan survey di lapangan. Dari estimasi harga tersebut, akan dihitung *cost benefir analysis* yang akan diterapkan dengan sistem *annual pay* per tahun.

6.4.3 Analisa Biaya Tahunan

Dalam melakukan implementasi perangkat lunak ini, ada beberapa kebutuhan lain untuk operasional, untuk daftar kebutuhan operasional beserta *cost* yang dikeluarkan ada pada tabel dibawah

Tabel 6.9 Biaya Kebutuhan per Tahun

Biaya Bulanan	Satuan	Harga
VPS	1	Rp 600,000.00
Domain	1	Rp 240,000.00
IP Public	1	Rp 2,880,000.00
Total Biaya Tahunan		Rp 3,720,000.00

Asumsi Parameter	
Usia/lifetime	4
Suku bunga	0.07
Tingkat depresiasi	0.25
Inflasi	0.03
harga per kWh	Rp1,630.00

Tabel 6.10 Total Biaya Operasional per Tahun

Barang	Satuan	Harga	Depresiasi	Watt	Biaya Listrik
Router Mikrotik RB 750	Pcs	Rp530,000	Rp132,500	6	Rp62,200.80
Router Mikrotik RB2011iL	Pcs	Rp1,350,000	Rp337,500	6	Rp62,200.80
Access Point Tenda AP4	Pcs	Rp230,000	Rp57,500	5	Rp51,834.00
Access Point UniFi AP AC LR	Pcs	Rp1,200,000	Rp300,000	24	Rp248,803.20
Kabel UTP Cat 5	m	Rp1,300	Rp325		
Lan Extender	Pcs	Rp50,000	Rp12,500		
RJ-45	Pcs	Rp1,000	Rp250		

Tabel 6.11 Total Operasional Paket 1

Paket 1	Qty	Depresiasi	Listrik
Router Mikrotik RB 750	1	Rp132,500	Rp62,201
Access Point Tenda AP4	1	Rp57,500	Rp51,834
Kabel UTP Cat 5	20	Rp6,500	Rp0
RJ-45	4	Rp1,000	Rp0
Subtotal		Rp197,500	Rp114,035
Total Operational Cost			Rp311,535

Tabel 6.12 Total Operasional Paket 2

Paket 2	Qty	Depresiasi	Listrik
Router Mikrotik RB 750	1	Rp132,500	Rp62,201
Access Point Tenda AP4	2	Rp115,000	Rp103,668
Kabel UTP Cat 5	100	Rp32,500	Rp0
RJ-45	4	Rp1,000	Rp0
Subtotal		Rp281,000	Rp165,869
Total Operational Cost			Rp446,869

Tabel 6.13 Total Operasional Paket 3

Paket 3	Qty	Depresiasi	Listrik
Router Mikrotik RB2011iL	1	Rp337,500	Rp62,201
Access Point UniFi AP AC LR	4	Rp1,200,000	Rp207,336
Kabel UTP Cat 5	100	Rp32,500	Rp0
RJ-45	4	Rp1,000	Rp0
Subtotal		Rp1,571,000	Rp269,537
Total Operasional Cost			Rp1,840,537

Tabel 6.14 Total Operasional Paket 4

Paket 4	Qty	Depresiasi	Listrik
Router Mikrotik RB2011iL	1	Rp337,500	Rp62,201
Access Point UniFi AP AC LR	6	Rp1,800,000	Rp311,004
Kabel UTP Cat 5	100	Rp32,500	Rp0
RJ-45	4	Rp1,000	Rp0
Subtotal		Rp2,171,000	Rp373,205
Total Operasional Cost			Rp2,544,205

6.4.4 Analisa Keuntungan, ROI, dan *Payback Period*

Analisa keuntungan atau profit pertahunnya pada usaha ini didapatkan dari pemasukan kotor setiap bulannya, dikurangi dengan pengeluaran. Hasil pengurangan tersebut dikalikan 12 bulan agar menjadi keuntungan pertahunnya. Ditambah, ada beberapa elemen atau variabel yang menjadi parameter yang dapat mempengaruhi perhitungan keuntungan tiap tahunnya, yaitu usia / lifetime perangkat, suku bunga, tingkat depresiasi, dan juga inflasi.

Asumsi Parameter	
Usia/lifetime	4
Suku bunga	0.07
Tingkat depresiasi	0.25
Inflasi	0.03

Penentuan usia / lifetime dari perangkat yang digunakan ini sendiri mengacu pada peraturan UU Pajak Penghasilan, dalam penetapan tersebut dibagi menjadi dua golongan, yaitu kelompok barang yang berwujud dan tak berwujud, untuk perangkat – perangkat IT masuk kedalam kategori barang berwujud dalam kelompok 1 dengan lifetime 4 tahun dan tingkat depresiasi 25%. Untuk inflasi sendiri mengacu pada sumber <http://bi.go.id> dimana inflasi ada pada kisaran 3%.

Sehingga dalam penghitungan analisa keuntungan per tahun, ditambah dengan nilai depresiasi aset per tahun, masing – masing aset yang digunakan pada tiap – tiap paket bisnis juga mengalami penurunan nilai tiap tahunnya, dan kenaikan keuntungan tiap tahunnya diakibatkan oleh inflasi. Dalam

analisa keuntungan per tahun, juga dilakukan perhitungan ROI yaitu (*Return of Investment*), juga akan dilakukan estimasi *payback period* yaitu lama waktu modal investasi akan kembali.

$$ROI = \frac{Profit}{Investment\ Cost}$$

Gambar 6.13 Rumus ROI

$$Payback\ Period = \frac{Initial\ Investment}{Periodic\ Cash\ Flow}$$

Gambar 6.14 Rumus *Payback Period*

Dalam melakukan perhitungan *benefit* yang didapat pemilik usaha dalam menyewa platform ini, ada beberapa keuntungan yang bisa didapatkan, beberapa diantaranya yaitu

- Promosi usaha dalam bentuk postingan Facebook dan Twitter dari pengunjung yang terotentikasi menggunakan Facebook dan Twitter
- CRM dalam bentuk mailing platform dari email yang didapatkan ketika pengguna terotentikasi menggunakan Gmail dan Facebook
- Analitik okupansi pemakaian Wi-Fi dalam periode waktu tertentu

Dalam menentukan perhitungan *benefit* ini dilakukan pendekatan dengan *potential benefit* yaitu *benefit* maksimal yang akan didapatkan apabila memenuhi beberapa kondisi tertentu dan ada parameter yang ada. Parameter tersebut adalah justifikasi dan perkiraan.

Tabel 6.15 Parameter Potential *Benefit*

Parameter	Satuan	Nilai
Socmed Followers	per account	500
Reach	percents	80%
Facebook Account	percents	40%
Twitter Account	percents	9%
Gmail Account	percents	51%
Total Promo	per year	24
Jangka waktu promo	day	7
Advertising Cost	Rp/1000 reach	4000
Mailing CRM	Rp/month/5k mail	82500

Tiap pengunjung diasumsikan memiliki jumlah followers pada akun Facebook atau Twitter nya sebanyak 500. Dengan reach yang akan didapatkan masing – masing dari total followers sebesar 80%. Dengan total pengunjung harian, diasumsikan total pengguna yang terotentikasi menggunakan Facebook, Gmail, dan Twitter masing – masing sebesar 40%, 51%, dan 9% dari data sebelumnya yang telah didapatkan. Dengan total promo yang dikeluarkan oleh pemilik usaha sebanyak 24 kali dalam satu tahun dan jangka waktu promo dalam waktu satu minggu. Advertising cost ini mengacu pada harga Facebook ads yang memiliki cost sebesar 4000

rupiah tiap 1000 reach, dimana ini berlaku untuk pengguna yang terotentikasi melalui Twitter dan Facebook. Ditambah dengan mailing CRM sebesar 82500 rupiah per bulan tiap 5000 kontak email, harga ini didapatkan dari harga rata – rata untuk software as service yang bergerak di bidang mailing CRM.

Tabel 6.16 Total Reach Justification

Visitors	Paket 1	Paket 2	Paket 3	Paket 4
Daily Visitor	30	60	200	300
Available Day	350	350	156	156
Total Visitors	10500	21000	31200	46800
Facebook	4200	8400	12480	18720
Twitter	945	1890	2808	4212
Gmail	5355	10710	15912	23868

Promotional Advertisements FB	Paket 1	Paket 2	Paket 3	Paket 4
Reach	33600	67200	224000	336000

Promotional Advertisements Tw	Paket 1	Paket 2	Paket 3	Paket 4
Reach	7560	15120	50400	75600

Mail Advertisements G	Paket 1	Paket 2	Paket 3	Paket 4
Reach	4284	8568	12730	19094

Pada Tabel 6.16 dilakukan penghitungan justifikasi total reach yang akan didapatkan dari masing – masing kanal media sosial, contohnya untuk paket 1 yang memiliki luas area 100m2 dan okupansi pemakaian Wi-Fi 25, maka diasumsikan tiap hari ada 30 pengunjung, apabila available day 350, maka *potential visitors* yang bisa didapatkan sebanyak 10 ribu pengunjung dengan jumlah akun Gmail, Facebook, dan Twitter sebanyak 5355, 4200, dan 945 tiap tahunnya. Dengan asumsi tersebut, apabila ada pengunjung harian 30 melakukan otentikasi. Maka akan didapatkan total reach promotion pada Facebook dan Twitter sebanyak 33600 dan 7560, hal ini didapatkan dari kalkulasi pengunjung harian * persentase pengguna media sosial * jumlah followers * jangka waktu promo * potential reach dari hasil posting. Sehingga didapatkan hasil promotional advertisement melalui Facebook dan Twitter untuk paket 1 sebesar 33600 dan 7560. Sedangkan untuk akun Gmail, tidak bisa melakukan promotional advertisement yang akan di post pada feed pengguna Facebook dan Twitter, tetapi lebih kearah mailing CRM dengan reach yang didapat per tahun, sebesar 4284 reach per mailing.

Tabel 6.17 Potential Benefit

Potential Benefit	Paket 1	Paket 2	Paket 3	Paket 4
Facebook	Rp134,400	Rp268,800	Rp896,000	Rp1,344,000
Twitter	Rp30,240	Rp60,480	Rp201,600	Rp302,400
Mail	Rp82,500	Rp165,000	Rp122,571	Rp183,857
Total Cost per Ads Period	Rp247,140	Rp494,280	Rp1,220,171	Rp1,830,257
24 Promo Ads/year	Rp5,931,360	Rp11,862,720	Rp29,284,114	Rp43,926,171

Dari hasil potential *benefit* diatas, angka tersebut didapatkan dari hasil kali antara total potential reach yang dimiliki tiap – tiap media sosial Facebook dan Twitter dikalikan dengan harga reach per 1000 view sebesar 4000 rupiah. Untuk harga mail, didapatkan dari hasil perkalian total mail dikalikan dengan harga mailing CRM tiap 5000 email

Tabel 6.18 ROI & *Benefit* Paket 1

Paket 1 ROI	Tahun 0	Tahun 0	Tahun 2	Tahun 3
Revenue	Rp5,931,360	Rp6,109,301	Rp6,292,580	Rp6,481,357
Tarif Sewa/mo	Rp2,500,000	Rp2,575,000	Rp2,652,250	Rp2,731,818
Operational Cost	Rp311,535	Rp311,535	Rp311,535	Rp311,535
<i>Benefit/year</i>	Rp3,119,825	Rp6,342,591	Rp9,671,386	Rp13,109,391
ROI	125%	246%	365%	480%
Payback Period (tahun)				0.90

Tabel 6.19 ROI & *Benefit* Paket 2

Paket 2 ROI	Tahun 0	Tahun 0	Tahun 2	Tahun 3
Revenue	Rp11,862,720	Rp12,218,602	Rp12,585,160	Rp12,962,714
Tarif Sewa/mo	Rp5,000,000	Rp5,150,000	Rp5,304,500	Rp5,463,635
Operational Cost	Rp446,869	Rp446,869	Rp446,869	Rp446,869
<i>Benefit/year</i>	Rp6,415,851	Rp13,037,584	Rp19,871,375	Rp26,923,585
ROI	128%	253%	375%	493%
Payback Period (tahun)				0.85

Tabel 6.20 ROI & Benefit Paket 3

Paket 3 ROI	Tahun 0	Tahun 0	Tahun 2	Tahun 3
Revenue	Rp29,284,114	Rp30,162,638	Rp31,067,517	Rp31,999,542
Tarif Sewa/mo	Rp12,000,000	Rp12,360,000	Rp12,730,800	Rp13,112,724
Operational Cost	Rp1,840,537	Rp1,840,537	Rp1,840,537	Rp1,840,537
Benefit/year	Rp15,443,577	Rp31,405,678	Rp47,901,858	Rp64,948,140
ROI	129%	254%	376%	495%
Payback Period (tahun)				0.90

Tabel 6.21 ROI & Benefit Paket 4

Paket 4 ROI	Tahun 0	Tahun 0	Tahun 2	Tahun 3
Revenue	Rp43,926,171	Rp45,243,957	Rp46,601,275	Rp47,999,314
Tarif Sewa/mo	Rp18,000,000	Rp18,540,000	Rp19,096,200	Rp19,669,086
Operational Cost	Rp2,544,205	Rp2,544,205	Rp2,544,205	Rp2,544,205
Benefit/year	Rp23,381,967	Rp47,541,718	Rp72,502,589	Rp98,288,612
ROI	130%	256%	380%	500%
Payback Period (tahun)				0.88

Dari hasil perhitungan ROI dan *benefit* diatas didapatkan hasil dari semua paket, ROI melebihi 100% dan payback period kurang dari satu tahun. *Benefit* yang didapatkan dari hasil diatas merupakan hasil dari potential *benefit* yang didapatkan dari hasil advertisement yang dilakukan postingan orang yang terotentikasi, untuk Facebook dan Twitter. Sedangkan ada dampak lain berupa material yang bisa didapatkan, yaitu peningkatan jumlah pengunjung yang datang, namun masih belum ada data yang valid mengenai percobaan untuk itu, oleh karena itu, perhitungan *benefit* hanya dilakukan dalam hal menghitung potential *benefit* / biaya yang harusnya dikeluarkan untuk melakukan *advertisement* namun sudah ter-cover oleh penggunaan *platform* ini.

6.4.5 Analisa Biaya NPV (*Net Present Value*)

1. Net Present Value

NPV merupakan selisih antara pengeluaran dan pemasukan yang telah didiskon dengan menggunakan social opportunity cost of capital sebagai diskon faktor, atau dengan kata lain merupakan arus kas yang diperkirakan pada masa yang akan datang yang didiskonkan pada saat ini. Untuk menghitung NPV diperlukan data tentang perkiraan biaya investasi, biaya operasi, dan pemeliharaan serta perkiraan manfaat/*benefit* dari proyek yang direncanakan. NPV dapat dihitung dengan menggunakan rumus:

$$\text{Net Present Value} = \frac{B_0 - C_0}{(1+r)^0} + \frac{B_1 - C_1}{(1+r)^1} + \frac{B_2 - C_2}{(1+r)^2} \dots \frac{B_n - C_n}{(1+r)^n}$$

Apabila $NPV > 0$ maka layak untuk dilakukan investasi, sebaliknya apabila $NPV < 0$ maka investasi tidak layak untuk dilakukan. Dari justifikasi tersebut didapatkan hasil sebagai berikut.

Tabel 6.22 Perhitungan NPV Paket 1

Paket 1 NPV	Tahun 0	Tahun 0	Tahun 2	Tahun 3
Cashflow Setahun	Rp5,931,360	Rp6,109,301	Rp6,292,580	Rp6,481,357
NPV	Rp20,960,643			

Tabel 6.23 Perhitungan NPV Paket 2

Paket 2 NPV	Tahun 0	Tahun 0	Tahun 2	Tahun 3
Cashflow Setahun	Rp11,862,720	Rp12,218,602	Rp12,585,160	Rp12,962,714
NPV	Rp49,629,196			

Tabel 6.24 Perhitungan NPV Paket 3

Paket 3 NPV	Tahun 0	Tahun 0	Tahun 2	Tahun 3
Cashflow Setahun	Rp29,284,114	Rp30,162,638	Rp31,067,517	Rp31,999,542
NPV	Rp122,513,811			

Tabel 6.25 Perhitungan NPV Paket 4

Paket 4 NPV	Tahun 0	Tahun 0	Tahun 2	Tahun 3
Cashflow Setahun	Rp43,926,171	Rp45,243,957	Rp46,601,275	Rp47,999,314
NPV	Rp183,770,717			

Dalam melakukan penghitungan NPV, dilakukan tiap paket, karena analisa manfaat tiap aset terlalu sulit untuk dijustifikasi, karena aset yang ada pada tiap paket merupakan satu kesatuan, sehingga pertimbangan penghitungan NPV dilakukan pada tiap paket. Hasil yang didapatkan NPV pada tiap - tiap paket untuk tiga tahun kedepan bernilai positif, ini artinya layak untuk dilakukan investasi.

(Halaman ini sengaja dikosongkan)

BAB VII KESIMPULAN DAN SARAN

Bab ini akan menjelaskan mengenai kesimpulan dan saran dari hasil keseluruhan pengerjaan penelitian studi kasus ini dan juga akan menjelaskan beberapa keterbatasan penelitian sehingga dapat dijadikan pertimbangan untuk penelitian selanjutnya.

7.1 Kesimpulan

Pada bagian ini akan menjelaskan mengenai kesimpulan dari penelitian. Kesimpulan yang didapatkan berasal dari jawaban atas semua rumusan masalah penelitian. Aplikasi ini menitik beratkan pada komunikasi perangkat lunak dengan router menggunakan RouterOS API dan juga otentikasi pada tiga media sosial yaitu Google, Facebook, dan Twitter. Dalam melakukan komunikasi antara perangkat lunak yang dibuat dengan router Mikrotik, menggunakan RouterOS API yang telah disediakan oleh komunitas pengembang.

Dalam bab analisa bisnis, dijelaskan tentang *cost structure* dan juga lingkungan implementasi yang cocok, perangkat ini cocok untuk diterapkan untuk tempat – tempat open space, café, ataupun pusat perbelanjaan dimana tiap – tiap tempat memiliki kondisional implementasi yang berbeda – beda.

Apabila dibandingkan margin dari tiap *cost structure*, maka akan terlihat adanya gap yang besar antara paket satu dan dua dengan paket tiga dan empat, hal itu dikarenakan adanya perbedaan segmen pasar dan alokasi kebutuhan yang berbeda sesuai dengan kondisi lapangan.

Untuk ROI dan *payback period* yang paling tinggi adalah paket 3, kemudian paket 4, disusul dengan paket 1 dan 3, untuk *payback period* dari masing – masing paket, secara rata – rata modal akan kembali dalam waktu 3.5 bulan. Kemudian dari penghitungan NPV untuk tiap – tiap paket dalam kurun waktu

tiga tahun, bernilai positif. Sehingga memunculkan kesimpulan bahwa investasi layak untuk dilakukan.

7.2 Saran

Berdasarkan pelaksanaan penelitian studi kasus ini, adapun saran yang dapat diberikan agar bisa dijadikan rekomendasi untuk penelitian selanjutnya, yaitu:

1. *Webview fix* untuk otentikasi Google

Ada beberapa temuan, dimana ada beberapa perangkat *smartphone* tidak bisa melakukan otentikasi pada Google dikarenakan pada saat membuka captive portat Mikrotik menggunakan webview (embedded view) sehingga menyalahi aturan dari Google, bahwa otentikasi harus melalui site for desktop dan tidak semua perangkat *smartphone* dapat melakukan hal tersebut.

2. Fungsi masih kurang terkait dengan aktivitas CRM

Aplikasi ini sebenarnya juga menekankan pada aktivitas CRM, namun belum terdapat aktivitas yang berkaitan tentang CRM pada implementasi praktis pada pengembangan aplikasi kali ini. Contohnya adalah fungsi mailing, hal hal tersebut dapat dikembangkan lebih lanjut untuk memperkaya fitur dari produk, namun tentu saja harus melakukan analisa kebutuhan terlebih dahulu.

DAFTAR PUSTAKA

- [1] Janrain and Blue, "The Value of Social Login," 2013.
- [2] Asosiasi Penyelenggara Jasa Internet Indonesia, "Survey Penetrasi & Perilaku Pengguna Internet Indonesia," Jakarta, 2016.
- [3] C. Franklin, "How Routers Work | HowStuffWorks." [Online]. Available: <https://computer.howstuffworks.com/router.htm/printable>. [Accessed: 07-Mar-2018].
- [4] I. G. Lembaga Penerbangan dan Antariksa Nasional (Indonesia), A. S. M., and - Hendrik H. S., *Jurnal analisis dan informasi kedirgantaraan = Journal of aerospace analysis and information.*, vol. 5, no. 1. Lembaga Penerbangan dan Antariksa Nasional, 2003.
- [5] "Measuring API Usability | Dr Dobb's." [Online]. Available: <http://www.drdoobs.com/windows/measuring-api-usability/184405654>. [Accessed: 07-Mar-2018].
- [6] Z. Fang, "How To: Get Started with the Open Graph - Facebook untuk Pengembang," 2011. [Online]. Available: <https://developers.facebook.com/blog/post/564/>. [Accessed: 07-Mar-2018].
- [7] "API Graf." [Online]. Available: <https://developers.facebook.com/docs/graph-api>. [Accessed: 07-Mar-2018].
- [8] "Using OAuth 2.0 to Access Google APIs | Google Identity Platform | Google Developers." [Online]. Available: <https://developers.google.com/identity/protocols/OAuth2>. [Accessed: 07-Mar-2018].
- [9] "OAuth 2.0 for Client-side Web Applications | Google

- Identity Platform | Google Developers.” [Online]. Available:
<https://developers.google.com/identity/protocols/OAuth2UserAgent>. [Accessed: 07-Mar-2018].
- [10] “Using OAuth 2.0 for Web Server Applications | Google Identity Platform | Google Developers.” [Online]. Available:
<https://developers.google.com/identity/protocols/OAuth2WebServer>. [Accessed: 07-Mar-2018].
- [11] “Using Oauth — Twitter Developers.” [Online]. Available:
<https://developer.twitter.com/en/docs/basics/authentication/overview/using-oauth>. [Accessed: 08-Mar-2018].
- [12] “Authorizing a request — Twitter Developers.” [Online]. Available:
<https://developer.twitter.com/en/docs/basics/authentication/guides/authorizing-a-request>. [Accessed: 08-Mar-2018].
- [13] “Access tokens from apps.twitter.com — Twitter Developers.” [Online]. Available:
<https://developer.twitter.com/en/docs/basics/authentication/guides/access-tokens>. [Accessed: 08-Mar-2018].
- [14] Wikipedia, “MikroTik User Meeting Venice 2014,” 2014.
- [15] “Mikrotik.ID: Berbagai Level Router OS dan Kemampuannya.” [Online]. Available:
http://www.mikrotik.co.id/artikel_lihat.php?id=7. [Accessed: 07-Mar-2018].
- [16] “Manual:API - MikroTik Wiki.” [Online]. Available:
<https://wiki.mikrotik.com/wiki/Manual:API>. [Accessed: 07-Mar-2018].
- [17] “Convention Hall Floor Plan.” [Online]. Available:

<https://www.smartdraw.com/hotel-floor-plan/examples/hotel-floor-plan/>. [Accessed: 16-Jul-2018].

- [18] “Cafe and Restaurant Floor Plan Solution | ConceptDraw.com | Restaurant Furniture Layout.” [Online]. Available: <https://www.conceptdraw.com/solution-park/building-cafe-restaurant-plans>. [Accessed: 16-Jul-2018].
- [19] “50 Coffee Shop Floor Plan Awesome.” [Online]. Available: <http://vidmaster.us/coffee-shop-floor-plan/>. [Accessed: 16-Jul-2018].

(halaman ini sengaja dikosongkan)

BIODATA PENULIS



Penulis bernama Rysma Aditya Widjaja yang lahir di Kota Surabaya pada tanggal 19 Maret 1996. Penulis merupakan anak pertama dari dua bersaudara. Penulis telah menamatkan pendidikan di SD Muhammadiyah 11 Surabaya, SMPN 2 Surabaya, SMAN 2 Surabaya, dan sekarang sedang menempuh pendidikan perkuliahan di Departemen Sistem Informasi ITS, Surabaya. Penulis masuk ITS menjadi angkatan 2014

melalui jalur tes tulis atau SBMPTN dengan nomor induk (NRP) 05211440000117. Selama masa perkuliahan, penulis aktif di berbagai kepanitiaan, antara lain menjadi staff sie publikasi dan dokumentasi Information Systems Expo 2015, kemudian menjadi *steering commitee* Information Systems Expo 2016. Selain itu, penulis juga mengikuti pelatihan antara lain LKMM Pra-TD untuk pelatihan kemampuan diri. Dan di akhir masa perkuliahan, penulis memilih topik Tugas Akhir pada bidang minat lab IKTI (Infrastruktur dan Keamanan Teknologi Informasi). Jika terdapat pertanyaan mengenai Tugas Akhir ini, penulis dapat dihubungi melalui *email* rysmawidjaja@gmail.com

