

**RANCANG BANGUN SISTEM MANAJEMEN USER
HOTSPOT MENGGUNAKAN MIKROTIK PHP API
BERBASIS WEB DI PONDOK PESANTREN
AL-LUQMANYAH**

SKRIPSI



Oleh:
KHOIRUL ANWAR
NPM. 10111100134

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PGRI YOGYAKARTA
2017**

**RANCANG BANGUN SISTEM MANAJEMEN USER
HOTSPOT MENGGUNAKAN MIKROTIK PHP API
BERBASIS WEB DI PONDOK PESANTREN
AL-LUQMANNIYYAH**

SKRIPSI



Oleh :

KHOIRUL ANWAR
NPM. 10111100134

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PGRI YOGYAKARTA
2017**

PENGESAHAN PEMBIMBING

SKRIPSI

RANCANG BANGUN SISTEM MANAJEMEN USER HOTSPOT

MENGGUNAKAN MIKROTIK PHP API BERBASIS WEB

DI PONDOK PESANTREN AL-LUQMANYAH



Menyetujui

Pembimbing I

Pembimbing II

Muhammad Fairuzabadi, M. Kom.
NIS. 19740926 200204 1 004

Marti Widya Sari, M. Eng.
NIS. 19790327 201201 2 009

PENGESAHAN DEWAN PENGUJI SKRIPSI

**RANCANG BANGUN SISTEM MANAJEMEN USER HOTSPOT
MENGUNAKAN MIKROTIK PHP API BERBASIS WEB
DI PONDOK PESANTREN AL-LUQMANNIYAH**

Oleh:

KHOIRUL ANWAR

NPM. 10111100134

Telah dipertahankan di depan Dewan Penguji

Pada tanggal 10 Mei 2017

dan dinyatakan telah memenuhi syarat

Susunan Dewan Penguji

	Nama	Tanda tangan	Tanggal
Ketua	: Wibawa, S.Si, M.Kom
Sekretaris	: Hasti Hasanati, M.T
Anggota	: 1. Meilany Nonsi Tentua, S.T, M.Eng
	: 2. Muhammad Fairuzabadi, M.Kom

Yogyakarta, Mei 2017

Dekan Fakultas Teknik

Universitas PGRI Yogyakarta

M. Fairuzabadi, M.Kom

NIS. 19740926 200204 1 004

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini :

Nama : Khoirul Anwar

NPM : 10111100134

Program Studi : Teknik Informatika

Fakultas : Teknik

Judul Skripsi : Rancang Bangun Sistem Manajemen *User Hotspot*
Menggunakan Mikrotik PHP API Berbasis Web Di Pondok
Pesantren Al-Luqmaniyyah

menyatakan dengan sesungguhnya bahwa skripsi yang saya tulis ini benar-benar merupakan pekerjaan saya sendiri, bukan merupakan pengambil alihan tulisan atau pikiran orang lain yang saya aku sebagai hasil tulisan atau hasil pemikiran saya sendiri.

Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, saya bersedia menerima sanksi atas perbuatan tersebut.

Yogyakarta, Mei 2017

Khoirul Anwar

MOTTO DAN PERSEMBAHAN

MOTTO:

“... ini hanyalah belokan, bukan jalan buntu!!!”

-noname.

“Talk is Cheap. Show Me Your Code”

- Linus Torvalds.

PERSEMBAHAN

Skripsi ini kupersembahkan untuk:

Kedua orang tuaku

Adikku

Keluarga besarku

Sahabat-sahabatku

Almamaterku

KATA PENGANTAR

Alhamdulillah, Puji syukur penulis panjatkan kepada Allah SWT karena dengan restu-Nya pelaksanaan dan penyusunan skripsi yang berjudul “Rancang Bangun Sistem Manajemen *User Hotspot* Menggunakan Mikrotik PHP API Berbasis Web Di Pondok Pesantren Al-Luqmaniyyah” dapat diselesaikan sebagai persyaratan menyelesaikan Sarjana Strata Satu (S1) Jurusan Teknik Informatika, Fakultas Teknik Universitas PGRI Yogyakarta..

Penulisan skripsi ini tidak terlepas dari bantuan dan dukungan pelbagai pihak. Oleh karena itu, ucapan terima kasih penulis sampaikan kepada :

1. Bapak Sarwi dan Emak Sukijah tercinta, yang senantiasa mendoakan, memberikan motivasi dan semangat kepada penulis.
2. Bapak M. Fairuzabadi, M. Kom., selaku Dekan Fakultas Teknik Universitas PGRI Yogyakarta dan Dosen Pembimbing I yang telah membantu, membimbing dan mengarahkan dalam skripsi ini.
3. Ibu Marti Widya Sari, M. Eng, selaku Ketua Program Studi Teknik Informatika dan selaku Dosen Pembimbing II yang telah memberikan arahan dan dukungan sehingga penulisan skripsi ini terselesaikan.
4. Semua dosen dan guru yang telah memberikan ilmu, arahan, dan dukungan kepada penulis selama ini.
5. Sedulur “Angon & Ngajozz”, terimakasih telah berbagi ilmu maupun support dari segi spiritual dan material.
6. Keluarga Besar Ponpes. Al-Luqmaniyyah Yogyakarta, yang berkenan menerima saya dalam memberi ruang dalam penelitian skripsi saya.

7. Teman-teman Angkatan 2010, terima kasih untuk semua saat belajar mengenai apapun yang kita bahas selama kuliah di Universitas PGRI Yogyakarta.
8. Semua pihak yang tidak bisa sebutkan satu per satu, terima kasih atas segala bantuannya.

Semoga Allah SWT memberikan balasan yang berlipat ganda dari segala kebaikan yang telah dilakukan. Penulis menyadari bahwa skripsi ini masih belum sempurna. Oleh karena itu, kritik dan saran yang membangun penulis harapkan sebagai masukan ke arah yang lebih baik lagi.

Demikian penulis berharap agar laporan penelitian ini dapat berguna untuk kita semua.

Yogyakarta, Mei 2017

Penulis

**RANCANG BANGUN SISTEM MANAJEMEN USER HOTSPOT
MENGUNAKAN MIKROTIK PHP API BERBASIS WEB DI PONDOK
PESANTREN AL-LUQMANIYYAH**

Khoirul Anwar
10111100134
email: dev@ipnudi.or.id

ABSTRAK

Tujuan dari penelitian ini yaitu membangun Sistem Manajemen User Hotspot yang ada di Pondok Pesantren Al-Luqmaniyyah. Pondok pesantren tersebut menggunakan mikrotik routerboard sebagai *routing* jaringan lokal dan sekaligus sebagai *gateway* internet *hotspot*. Dalam penggunaan fitur *user manager hotspot* tidak maksimal jika menggunakan lisensi mikrotik level 5 kebawah. Berdasarkan masalah tersebut, penelitian ini bertujuan untuk membuat aplikasi alternatif yang bisa mempermudah pengguna dalam memanajemen *user hotspot* di Pondok Pesantren Al-Luqmaniyyah.

Penelitian ini menggunakan metode studi pustaka, studi lapangan dan eksplorasi internet digunakan untuk pengumpulan data dalam pembuatan sistem. Rancangan sistem ini menggunakan bahasa pemrograman PHP (PHP: *Hypertext Preprocessor*) dan MySQL dan memanfaatkan Mikrotik API (*Application Programming Interface*). Sistem yang dibuat telah dilakukan pengujian menggunakan metode *blackbox test* dan *alpha test*.

Berdasarkan hasil pengujian, sistem manajemen *user hotspot* yang dibangun dapat dijadikan solusi pengganti *user manager*. Sistem manajemen *user hotspot* yang dibangun telah terintegrasi dengan baik.

Kata Kunci: Mikrotik, *Hotspot*, PHP, API (*Application Program Interface*)

HOTSPOT USER MANAGEMENT SYSTEM DESIGN USING A WEB-BASED MIKROTIK PHP API IN AL-LUQMANIYYAH BOARDING SCHOOL

Khoirul Anwar
10111100134
PGRI University of Yogyakarta
email: dev@ipnudi.or.id

ABSTRACT

The aim of the research is to build a management system of Hotspot User in Al-Luqmaniyyah Boarding School. The boarding school used mikrotik routerboard as local network routing as well as gateway internet hotspots. The use of user manager hotspot feature use was not maximal if using mikrotik license below level 5. Based on this issues, this research aims to set an alternative application which can ease users in managing user hotspot in Al-Luqmaniyyah Boarding School.

This research using the method of the study of the literature, the study of the field and the exploration of the internet. They were used for data collection in manufacturing systems. The design of these systems used the programming language PHP (PHP: Hypertext Processor) and MySQL and using Mikrotik API (Application Programming Interface).

The system created has done testing method using blackbox test and the alpha test. Based in the result of test, the user hotspot user hotspot management system built can be the solution as user manager substitution. Hotspot user management system built has been properly integrated.

Keyword: Mikrotik, Hotspot, PHP, API (Application Program Interface)

DAFTAR ISI

HALAMAN JUDUL	i
PENGESAHAN PEMBIMBING	ii
PENGESAHAN DEWAN PENGUJI SKRIPSI	iii
PERNYATAAN KEASLIAN TULISAN	iv
MOTTO DAN PERSEMBAHAN.....	v
KATA PENGANTAR.....	vi
ABSTRAK	viii
ABSTRACT	ix
DAFTAR ISI.....	x
DAFTAR TABEL	xiii
DAFTAR GAMBAR.....	xiv
DAFTAR MODUL.....	xvii
BAB I PENDAHULUAN	1
A. Latar Belakang Masalah	1
B. Identifikasi Masalah	3
C. Perumusan Masalah	3
D. Batasan Masalah	4
E. Tujuan Penelitian	4
F. Manfaat Penelitian	5
G. Jadwal Penelitian	5
H. Sistematika Penulisan	6
BAB II LANDASAN TEORI	8
A. Tinjauan Pustaka	8
B. Landasan Teori	9
1. Menenal Router	9
2. Pengertian Sistem Operasi.....	11
3. Pengertian Mikrotik.....	12
4. PHP (<i>HyperText Preprocessor</i>).....	14
5. MySQL	16

6. API (<i>Application Program Interface</i>)	18
7. Sejarah Singkat Pondok Pesantren Al-Luqmaniyyah.....	19
BAB III METODE PENELITIAN	31
A. Obyek Penelitian	31
B. Metode Pengumpulan Data	31
1. Studi Pustaka	31
2. Observasi	31
3. Studi dan <i>Interview</i> (Wawancara)	32
C. Alat-alat Penelitian	32
1. Perangkat Keras (<i>Hardware</i>).....	32
2. Perangkat Lunak (<i>Software</i>)	32
D. Perancangan Sistem.....	33
1. DFD (Data Flow Diagram).....	33
2. Data Flow Diagram Level 1	34
3. Data Flow Diagram Level 2 Proses 1	35
4. Data Flow Diagram Level 2 Proses 2	36
5. Data Flow Diagram Level 2 Proses 3	37
6. Data Flow Diagram Level 2 Proses 4	38
7. Data Flow Diagram Level 2 Proses 5	39
8. Data Flow Diagram Level 2 Proses 6	40
9. Data Flow Diagram Level 2 Proses 7	41
10. Data Flow Diagram Level 2 Proses 8	42
11. Entity Relationship Diagram	42
E. Perancangan Tabel Basis Data	43
1. Tabel Admin	43
2. Tabel Member	44
3. Tabel Konfigurasi	44
4. Tabel Pesan	45
5. Tabel Paket	45
6. Tabel Billing	46
F. Perancangan Antar Muka Sistem	46

1. Perancangan Struktur Menu	47
2. Perancangan Antarmuka Sistem.....	49
BAB IV IMPLEMENTASI DAN PEMBAHASAN	59
A. Implementasi	59
1. Konfigurasi Router Mikrotik.....	59
2. Tampilan Halaman Admin	62
3. Tampilan Halaman <i>Member</i>	81
B. Pembahasan	84
1. Hasil Pengujian.....	84
2. Kelebihan.....	88
3. Kekurangan.....	88
BAB V PENUTUP	90
A. Kesimpulan.....	90
B. Saran	90
DAFTAR PUSTAKA	92
LAMPIRAN.....	93
Lampiran 1: Script Program.....	93
Lampiran 2: Hasil Kuisisioner.....	108
DAFTAR RIWAYAT HIDUP	139

DAFTAR TABEL

Tabel 1.1. Perencanaan Jadwal Skripsi	5
Tabel 3.1. Tabel Admin	44
Tabel 3.2. Tabel Member	44
Tabel 3.3. Tabel Konfigurasi	45
Tabel 3.4. Tabel Pesan	45
Tabel 3.5. Tabel Paket.....	46
Tabel 3.6. Tabel Billing	46
Tabel 4.1. Pengujian Usabilitas Aplikasi	85
Tabel 4.2. Pengujian Fungsionalitas Aplikasi.....	86

DAFTAR GAMBAR

Gambar 2.1. Struktur Pengurus Pondok Pesantren Al-Luqmaniyyah	21
Gambar 2.2. Pengurus Pusat Pondok Pesantren Al-Luqmaniyyah	22
Gambar 2.3. Struktur Kepengurusan Komplek.....	22
Gambar 3.1. Data Flow Diagram Level 0	33
Gambar 3.2. DFD Level 1 Sistem Manajemen User Hotspot.....	34
Gambar 3.3. DFD Level 2 Proses Login Admin.....	35
Gambar 3.4. DFD Level 2 Proses Konfigurasi	36
Gambar 3.5. DFD Level 2 Proses Pesan	37
Gambar 3.6. DFD Level 2 Proses Paket	38
Gambar 3.7. DFD Level 2 Proses Billing	39
Gambar 3.8. DFD Level 2 Proses Profil	40
Gambar 3.9. DFD Level 2 Proses Pengolahan Member	41
Gambar 3.10. DFD Level 2 Proses Login Member	42
Gambar 3.11. Diagram ERD Sistem Manajemen User Hotspot.....	43
Gambar 3.12. Rancangan Struktur Menu Admin.....	47
Gambar 3.13. Rancangan Struktur Menu Member	48
Gambar 3.14. Rancangan Halaman Login System	49
Gambar 3.15. Rancangan Halaman Dashboard Admin	50
Gambar 3.16. Rancangan Halaman Pesan Admin	50
Gambar 3.17. Rancangan Halaman Pesan Baru.....	51
Gambar 3.18. Rancangan Halaman Lihat Pesan	52
Gambar 3.19. Rancangan Halaman Pesan Keluar	52

Gambar 3.20. Rancangan Halaman Paket Admin.....	53
Gambar 3.21. Rancangan Halaman Tambah Paket.....	53
Gambar 3.22. Rancangan Halaman Billing Admin	54
Gambar 3.23. Rancangan Halaman Daftar Member	54
Gambar 3.24. Rancangan Halaman Tambah Member	55
Gambar 3.25. Rancangan Halaman Aktivasi	55
Gambar 3.26. Rancangan Halaman Konfigurasi.....	56
Gambar 3.27. Rancangan Halaman Profil.....	56
Gambar 3.28. Rancangan Halaman Dashboard Member.....	57
Gambar 3.29. Rancangan Halaman Paket Member	57
Gambar 3.30. Rancangan Halaman Billing Member	58
Gambar 4.1. Konfigurasi IP Address Statis	60
Gambar 4.2. Pengaturan DNS pada Mikrotik	60
Gambar 4.3. Konfigurasi Service pada Mikrotik	61
Gambar 4.4. Konfigurasi Grup User API Pada Mikrotik	61
Gambar 4.5. Membuat User API Pada Mikrotik	62
Gambar 4.6. Tampilan Halaman Login	63
Gambar 4.7. Halaman Dashboard Admin	64
Gambar 4.8. Halaman Pesan	65
Gambar 4.9. Halaman Kirim Pesan.....	66
Gambar 4.10. Halaman Lihat Pesan.....	67
Gambar 4.11. Halaman Pesan Keluar	69
Gambar 4.12. Halaman Daftar Member.....	70

Gambar 4.13. Halaman Tambah Member.....	71
Gambar 4.14. Halaman Aktivasi Member	72
Gambar 4.15. Halaman Daftar Paket	73
Gambar 4.16. Halaman Tambah Paket	74
Gambar 4.17. Halaman Daftar Member.....	76
Gambar 4.18. Halaman Konfigurasi	77
Gambar 4.19. Halaman Profil	79
Gambar 4.20. Halaman Dashboard Member.....	81
Gambar 4.21. Halaman Pesan Member.....	82
Gambar 4.22. Halaman Paket Member	82
Gambar 4.23. Halaman Billing Member.....	83
Gambar 4.24. Halaman Lihat Billing.....	83
Gambar 4.25. Halaman Profil Member.....	84

DAFTAR MODUL

Modul 1. Script Proses Login	64
Modul 2. Script Menampilkan Info di Dashboard	65
Modul 3. Script Query Menampilkan Pesan Admin.....	66
Modul 4. Script Proses Mengirim Pesan.....	67
Modul 5. Script Menampilkan Pesan	68
Modul 6. Script Membalas Pesan	68
Modul 7. Script Menampilkan Pesan Keluar	69
Modul 8. Script Menampilkan Daftar Member.....	70
Modul 9. Script Menambah Member	72
Modul 10. Script Aktivasi member.....	73
Modul 11. Script Menampilkan Daftar Paket	74
Modul 12. Script Proses Tambah Paket	76
Modul 13. Script Menampilkan Daftar Billing	77
Modul 14. Script Proses Konfigurasi	78
Modul 15. Script Menampilkan Data Pengguna	79
Modul 16. Script Edit Profil.....	80

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Perkembangan dan kemajuan dunia teknologi saat ini memang tidak bisa dibantahkan lagi. Hampir semua lini kehidupan tidak bisa dilepaskan dari teknologi. Kebutuhan internet di institusi pemerintah, lembaga pendidikan maupun perumahan sebagai media informasi dan komunikasi saat ini dirasa sangat penting. Maka dari itu, tidak heran apabila berbagai institusi maupun lembaga tersebut menyediakan fasilitas *hotspot* agar lebih mudah dalam mengakses informasi.

Pondok Pesantren Al-Luqmaniyyah yang berada di Umbulharjo merupakan pondok pesantren salaf yang memanfaatkan teknologi informasi sebagai media administrasi pondok pesantren, dimana komputer antar kompleks saling terhubung untuk mempermudah penyimpanan data secara terpusat.

Pondok Pesantren Al-Luqmaniyyah menggunakan jaringan Mikrotik sebagai *routing* dan manajemen pada jaringannya. Mikrotik merupakan salah satu sistem operasi yang dirancang khusus sebagai router jaringan yang handal. Selain konfigurasinya yang tidak terlalu sulit, Mikrotik juga memiliki berbagai fitur lengkap untuk jaringan kabel dan *wireless*. Jaringan *wireless* atau yang biasa disebut *hotspot* ini merupakan salah satu fitur andalan Mikrotik, yang mana fitur tersebut mampu memanajemen *bandwidth* dan mudah untuk digunakan dalam mengatur pengguna.

Untuk mendukung kemudahan dalam manajemen *hotspot*, saat ini banyak penyedia layanan *hotspot* yang menggunakan fitur user manager (userman) atau biasa disebut sebagai Radius pada Mikrotik. Hal itu adalah sebagai sistem manajemen *hotspot* mereka, dimana fitur tersebut mampu menghandle segala kebutuhan tentang manajemen *hotspot*. Diantara yang bisa dihandle adalah *generate user*, limitasi bandwidth, multi customer, dan masih banyak lagi. Akan tetapi pemanfaatan fitur tersebut dibatasi oleh lisensi Mikrotik itu sendiri dan fitur ini bisa digunakan secara maksimal apabila menggunakan lisensi Mikrotik level 6 dengan fasilitas *unlimited user max concurrent*. Berbeda dengan lisensi Mikrotik level 5, penggunaan fitur user manager ini masih sangat terbatas, yaitu hanya mampu digunakan 50 *user max concurrent*. Sedangkan untuk lisensi Mikrotik level 4 hanya bisa digunakan 20 *user max concurrent*, dan lisensi Mikrotik level 3 bisa digunakan 10 *user max concurrent*.

Bagi pengguna jaringan Mikrotik, boleh dibilang untuk memperoleh Mikrotik dengan lisensi level 6 pada saat ini masih cukup mahal jika dibandingkan dengan lisensi yang lebih rendah. Sementara apabila penggunaan fitur user manager ini tetap dipaksakan dan penggunaan *hotspot* melebihi dari batas maksimal, maka akibatnya akan berpengaruh fatal. Pasalnya ketika *user* telah melebihi batas maksimal akan mengalami *trouble*, yaitu ditandai dengan tidak dapat tersambung nya *user* dengan *hotspot gateway*. Tentunya hal ini menjadi sebuah masalah yang besar bagi penyedia layanan internet dengan pengguna banyak.

Berdasarkan latar belakang permasalahan tersebut, pada penelitian kali ini peneliti tertarik untuk membuat aplikasi pihak ketiga untuk manajemen *user hotspot* pada Mikrotik dengan mengangkat judul ***“Rancang Bangun Sistem Manajemen User Hotspot Menggunakan Mikrotik PHP API Berbasis Web di Pondok Pesantren Al-Luqmaniyyah”***. Dimana rancangan aplikasi ini mampu mempermudah petugas untuk mengatur dan manajemen user hotspot pada Mikrotik secara efisien dan bersifat *user-friendly*. Aplikasi ini juga mampu bekerja seperti fitur *User Manager*. Selain itu juga tidak memakan biaya mahal sehingga sangat efisien untuk digunakan untuk sistem manajemen user hotspot yang ada di Pondok Pesantren Al-Luqmaniyyah, tanpa perlu membeli lisensi Mikrotik level 6.

B. Identifikasi Masalah

Berdasarkan permasalahan dari latar belakang di atas dapat ditarik beberapa kesimpulan untuk identifikasi masalah yaitu:

1. Penggunaan fitur *user manager* yang kurang efisien pada MikroTik level 5 kebawah.
2. Pengelolaan user hotspot pada winbox yang kurang efisien untuk pengelolaan pengguna yang lebih detail pada MikroTik.
3. Minimnya aplikasi pihak ketiga untuk mengatur dan manajemen *user hotspot* sebagai pengganti user manager yang efektif.

C. Perumusan Masalah

Berdasarkan uraian pada latar belakang dan identifikasi masalah dapat dibuat rumusan masalah sebagai berikut:

1. Bagaimana membuat aplikasi pihak ketiga Mikrotik dengan memanfaatkan Mikrotik PHP API?
2. Bagaimana aplikasi tersebut mampu mengatur dan manajemen *user hotspot* tanpa menggunakan fitur user manager pada Mikrotik?
3. Bagaimana aplikasi tersebut bisa diterapkan dalam jaringan di Pondok Pesantren Al-Luqmaniyyah?

D. Batasan Masalah

Karena besarnya permasalahan dan keterbatasan waktu serta pengetahuan penulis, agar pembahasan tidak menyimpang dari tujuan, maka penulis membuat pembatasan masalah sebagai berikut:

1. Penelitian dilakukan di Pondok Pesantren Al-Luqmaniyyah.
2. Bahasa pemrograman yang digunakan adalah bahasa PHP.
3. Penelitian menggunakan perangkat Mikrotik RouterOS level 4.
4. Tidak membahas tentang sistem jaringan internet yang digunakan.
5. Aplikasi *web* yang dibangun untuk *user* dan administrator.
6. Tidak membahas detail tentang sistem keamanan *website* dan *database* lokal yang dibuat.

E. Tujuan Penelitian

Tujuan yang ingin dicapai dari penulisan skripsi ini adalah:

1. Untuk mempermudah pengaturan *user hotspot* dalam router MikroTik di Pondok Pesantren Al-Luqmaniyyah.
2. Untuk mengoptimalkan penggunaan fitur yang ada dalam Mikrotik.

3. Membuat aplikasi pihak ketiga dengan memanfaatkan fitur PHP API yang disediakan oleh Mikrotik.

F. Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini adalah:

1. Mengetahui perkembangan ilmu pengetahuan dan teknologi yang berkembang saat ini khususnya dalam perkembangan teknologi router.
2. Menjadikan aplikasi ini sebagai solusi untuk mempermudah pengaturan dan manajemen *user hotspot* pada Mikrotik.
3. Mengetahui bagaimana proses perancangan *web* dalam pemanfaatan fitur PHP API pada MikroTik.

G. Jadwal Penelitian

Adapun Perencanaan jadwal skripsi yang akan dilaksanakan berdasarkan metodologi yang diambil sebagai berikut:

Tabel 1. 1. Perencanaan Jadwal Skripsi

No	Kegiatan	Bulan (2016)															
		September				Oktober				November				Desember			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Pengumpulan Data																
2	Penulisan Proposal																
3	Perancangan Sistem																
4	Pembuatan Sistem																
5	Implementasi Sistem																
6	Pengujian Sistem																
7	Penulisan Laporan Akhir																

H. Sistematika Penulisan

Sistematika penulisan dalam penelitian ini berupa penjelasan tentang pokok-pokok bahasan pada setiap bab yang meliputi:

BAB I PENDAHULUAN

Bab ini menjelaskan tentang latar belakang masalah, identifikasi masalah, batasan masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB II KAJIAN PUSTAKA

Bab ini berisi tinjauan pustaka yang berkaitan dengan penelitian-penelitian sebelumnya yang masih berhubungan dengan penelitian ini serta landasan teori yang berisi tentang teori-teori yang mendasari penulisan tugas akhir serta menjelaskan beberapa pengertian, konsep dasar serta hal lain yang berhubungan dengan judul yang penulis angkat.

BAB III METODOLOGI PENELITIAN

Bab ini membahas tentang objek penelitian, tempat dan waktu penelitian, metode pengumpulan data, metode analisis data, dan metode pengembangan sistem.

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

Bab ini membahas tentang hasil penelitian (implementasi sistem) dan evaluasi pembahasan hubungan antar entitas, hubungan antar aplikasi, deskripsi kemampuan dan kelemahan sistem yang telah dibuat.

BAB V PENUTUP

Bab ini berisi kesimpulan dari penelitian yang dilakukan dan disertai saran yang dapat diberikan sebagai pertimbangan untuk penelitian-penelitian berikutnya.

BAB II

LANDASAN TEORI

A. Tinjauan Pustaka

Berdasarkan referensi sebelumnya telah ada beberapa penelitian sejenis yang mengambil bahasan tentang Mikrotik, baik itu manajemen user hotspot pada Mikrotik maupun penggunaan API untuk mempermudah dalam melakukan penelitian.

Pada penelitian yaitu Pemanfaatan API pada Perangkat *Hotspot Gateway* dalam Pembuatan Sistem Akses Internet Berdasarkan *Volume Based* dan *Time Based Access* Berbasis Web, dimana aplikasi tersebut hanya dirancang untuk manajemen pelanggan warnet yang menggunakan *volume based* dan *time based access* yang artinya pelanggan warnet diberi dua pilihan, yaitu berdasarkan batas kuota dan batas waktu dalam menggunakan layanan internet (Okto, 2012).

Pada penelitian yaitu Perancangan Dan Implementasi *Billing Hotspot* Dengan Menggunakan PHP dan API pada Mikrotik di Cybercity Networks menyebutkan bahwa penggunaan API Mikrotik sangat efektif dalam memberi kemudahan kepada *administrator* dalam manajemen dan pemantauan terhadap *user hotspot* (Setiawan, 2015).

Pada penelitian yaitu Sistem Pemesanan Menu Berbasis Web Memanfaatkan Mikrotik API (Studi Kasus: Miaw Shake Cat Cafe), disebutkan bahwa memanfaatkan Mikrotik API sangat menguntungkan dan sangat fleksibel digunakan dalam berbagai hal (Mardiana, 2015).

B. Landasan Teori

1. Mengenal Router

Jaringan komputer sangat diperlukan dalam melakukan proses pengiriman data dari satu tempat ke tempat yang lain. Untuk menyambungkan jaringan komputer secara luas dibutuhkan peralatan tambahan agar proses komunikasi data tidak terhambat.

Router adalah sebuah alat jaringan komputer yang mengirimkan paket data melalui sebuah jaringan atau internet menuju tujuannya, melalui sebuah proses yang dikenal sebagai routing untuk menyambungkan jaringan yang luas (*Wide Area Network* – WAN). Proses routing terjadi pada OSI layer 3 (Lapisan jaringan seperti *Internet Protocol*) dari stack protokol tujuh lapis OSI. Sehingga router yang digunakan untuk menyambungkan LAN (*Local Area Network*) dan WAN harus mampu mendukung.

Berbeda dengan penggunaan hub dan switch, hub merupakan perangkat jaringan yang bekerja di OSI layer 1, *Physical Layer*. Sehingga dia hanya bekerja tak lebih sebagai penyambung atau *concentrator* saja, dan hanya menguatkan sinyal di kabel UTP, sedangkan switch merupakan perangkat jaringan yang bekerja pada OSI layer 2. Meski bentuknya serupa dengan hub, kita tidak sebaiknya menyebutnya dengan istilah switch hub.

a. Pengertian Router

Router adalah perangkat yang dikhususkan untuk menangani koneksi antara dua atau lebih jaringan yang terhubung melalui *packet swicthing*.

Router bekerja dengan melihat alamat asal dan alamat tujuan dari paket yang melewatinya dan memutuskan rute yang akan dilewati paket tersebut untuk sampai ke tujuan. Router mengetahui alamat masing-masing komputer dilingkungan jaringan lokalnya, mengetahui alamat bridge dan router lainnya.

b. Fungsi Router

- 1) Router berfungsi utama sebagai penghubung antar dua atau lebih jaringan untuk meneruskan data dari satu jaringan ke jaringan lainnya.
- 2) Router mentransmisikan informasi dari satu jaringan ke jaringan lain yang sistem kerjanya mirip dengan bridge.
- 3) Digunakan juga untuk menghubungkan LAN ke sebuah layanan telekomunikasi seperti halnya telekomunikasi *leased line* atau Digital Subscriber Line (DSL). Router digunakan untuk menghubungkan LAN ke sebuah koneksi leased line seperti T1, atau T3, sering disebut sebagai *access server*.
- 4) Digunakan untuk menghubungkan jaringan lokal ke sebuah koneksi DSL disebut juga dengan DSL router. Router-router jenis tersebut umumnya memiliki fungsi firewall untuk melakukan

penapisan paket berdasarkan alamat sumber dan alamat tujuan paket tersebut, meski beberapa router tidak memilikinya. Router yang memiliki fitur penapisan paket disebut juga dengan *packet-filtering router*. Fungsi router umumnya memblokir lalu lintas data yang dipancarkan secara broadcast sehingga dapat mencegah adanya *broadcast storm* yang mampu memperlambat kinerja jaringan.

2. Pengertian Sistem Operasi

Sistem operasi dalam bahasa Inggris *operating system* atau yang disingkat dengan sebutan OS adalah komponen pengolah piranti lunak dasar (*essential component*) tersistem sebagai pengelola sumber daya perangkat keras komputer (*hardware*), dan menyediakan layanan umum untuk aplikasi perangkat lunak. Sistem operasi adalah jenis yang paling penting dari perangkat lunak sistem dalam sistem komputer. Tanpa sistem operasi, pengguna tidak dapat menjalankan program aplikasi pada komputer mereka, kecuali program *booting*.

Sistem operasi mempunyai penjadwalan yang sistematis mencakup perhitungan penggunaan memori, pemrosesan data, penyimpanan data, dan sumber daya lainnya.

Untuk fungsi-fungsi perangkat keras seperti sebagai masukan dan keluaran dan alokasi memori, sistem operasi bertindak sebagai perantara antara program aplikasi dan perangkat keras komputer, meskipun kode aplikasi biasanya dieksekusi langsung oleh perangkat keras dan seringkali

akan menghubungkan OS atau terputus oleh itu. Sistem operasi yang ditemukan pada hampir semua perangkat yang berisi komputer, dari ponsel dan *console* permainan video untuk super komputer dan *webserver*.

Adapun macam-macam dari sistem operasi modern adalah Linux, Android, iOS, Mac OS X, dan Microsoft Windows.

3. Pengertian Mikrotik

Mikrotik RouterOS adalah sistem operasi yang diperuntukkan sebagai *network* router. Didesain untuk memberikan kemudahan penggunaannya. Administrasinya bisa menggunakan Mikrotik RouterOS, merupakan sistem operasi linux base yang dilakukan melalui Windows *Application* (WinBox).

Selain itu instalasi bisa dilakukan pada PC Standard, PC yang akan dijadikan router mikrotik pun tidak memerlukan *resource* yang cukup besar untuk penggunaan *standard*, misalnya hanya sebagai gateway. Untuk keperluan beban yang besar (*network* yang kompleks, routing yang rumit) disarankan untuk mempertimbangkan pemilihan *resource* PC yang memadai.

a. Sejarah Mikrotik

Mikrotik adalah sebuah perusahaan kecil yang berkantor pusat di Latvia, bersebelahan dengan Rusia. Pembentukannya diprakasai oleh John Trully dan Arnis Riekstins. John Trully adalah seorang yang berasal dari Amerika yang bermigrasi ke Latvia. Di Latvia ia

berjumpa dengan Arnis, seorang sarjana Fisika dan Mekanik sekitar tahun 1995.

Di tahun 1996 John dan Arnis mulai me-routing dunia (misi Mikrotik adalah merouting seluruh dunia). Mulai dengan sistem Linux dan MS DOS yang dikombinasikan dengan teknologi Wireless-LAN (WLAN) Aeronet berkecepatan 2 Mbps di Moldova, negara tetangga Latvia, baru kemudian melayani lima pelanggannya di Latvia.

Prinsip dasar mereka bukan membuat Wireless ISP (W-ISP), tetapi membuat program router yang handal dan dapat dijalankan diseluruh dunia. Latvia hanya merupakan tempat eksperimen John dan Arnis, karena saat ini mereka sudah membantu negara-negara lain termasuk Srilanka yang melayani sekitar 400 pengguna.

Linux yang pertama kali digunakan adalah kernel 2.2 yang dikembangkan secara bersama-sama dengan bantuan 5-15 orang staf *Research and Development* (R&D). Mikrotik yang sekarang menguasai dunia routing di negara-negara berkembang. Menurut Arnis selain staf di lingkungan Mikrotik, mereka juga merekrut tenaga-tenaga lepas dan pihak ke tiga yang dengan intensif ikut mengembangkan Mikrotik secara *marathon*.

b. Jenis – jenis Mikrotik

Berdasarkan bentuk *hardware* yang di gunakan, mikrotik dapat digolongkan dalam dua jenis. Untuk memudahkan bagi pemula dalam memahami *Router Operating System* ini. dua jenis tersebut adalah :

1) Mikrotik RouterOS

MikroTik RouterOS adalah versi Mikrotik dalam bentuk perangkat lunak yang dapat diinstall pada komputer rumahan (PC) melalui media CD. File yang dibutuhkan dapat diunduh dalam bentuk file image Mikrotik RouterOS dari website resmi Mikrotik, www.mikrotik.com.

2) Built-In Hardware Mikrotik

Built-In Hardware Mikrotik merupakan Mikrotik dalam bentuk perangkat keras yang khusus dikemas dalam board router atau sering disebut routerBoard yang didalamnya sudah terinstall Mikrotik RouterOS.

4. PHP (*HyperText Preprocessor*)

Pada awalnya PHP merupakan akronim dari *Personal Home Page* (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilsan kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini, *interpreter* PHP sudah diimplementasikan dalam program C. Dalam rilis ini

disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang PHP: *Hypertext Preprocessing*.

Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek.

Versi terbaru dari bahasa pemrograman PHP 5 adalah versi 5.6.22 yang resmi dirilis pada tanggal 26 Mei 2016.

Beberapa kelebihan PHP untuk bahasa pemrograman web, antara lain:

- Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.

- *Web Server* yang mendukung PHP dapat ditemukan dimana – mana mulai dari *apache*, *IIS*, *Lighttpd*, hingga *Xitami* dengan konfigurasi yang relatif mudah.
- Dalam sisi pengembangan lebih mudah, karena banyaknya komunitas dan *developer* yang siap membantu dalam pengembangan,
- Dalam sisi pemahaman, PHP adalah bahasa pemrograman yang paling mudah karena memiliki referensi yang banyak.

PHP adalah bahasa pemrograman yang bersifat *open source*, dimana bahasa ini dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

5. MySQL

MySQL termasuk kedalam jenis perangkat lunak RDBMS (*Relational Database Management System*) yaitu sebuah perangkat lunak sistem manajemen basis data yang dirancang untuk mengatur sebuah basis data sebagai sekumpulan data yang disimpan secara terstruktur, dan melakukan operasi-operasi terhadap data atas permintaan penggunaanya, sebenarnya MySQL merupakan turunan dari SQL (*Structured Query Language*), yang merupakan suatu bahasa komputer yang mengikuti *standard ANSI (American National Standard institute)* yaitu sebuah bahasa *standard* yang digunakan untuk mengakses dan memanipulasi suatu sistem *database*, untuk kemudahan akses penulis menggunakan

paket XAMPP yang sudah termasuk *Apache server* dan *database* phpMyAdmin yang menggunakan bahasa program dari MySQL.

XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP, Perl. XAMPP merupakan *tool* yang menyediakan paket perangkat lunak. berikut penjelasan tentang *web server* Apache, PHP, MySQL, dan phpMyAdmin.

Apache bertugas untuk menghasilkan halaman web yang benar kepada peminta, berdasarkan kode php yang dituliskan oleh pembuat halaman web. jika diperlukan juga berdasarkan kode php yang dituliskan, maka dapat saja suatu database diakses terlebih dahulu (misal dalam MySQL) untuk mendukung halaman *web* yang dihasilkan.

PHP merupakan bahasa pemrograman untuk membuat *web* yang bersifat *server-side scripting*. PHP memungkinkan kita untuk membuat halaman *web* yang bersifat dinamis. PHP dapat di dajalankan pada berbagai macam *Operating System* (OS). PHP dapat memanfaatkan *database* untuk menghasilkan halaman *web* yang dinamis. sistem manajemen *database* yang sering digunakan bersama PHP adalah MySQL. namun PHP juga mendukung beberapa *database* seperti Oracle, MS. Acces, interbase, d-base, PostgreSQL dan lain-lain. Hingga kini PHP sudah berkembang hingga ke versi 7 yang menambah beberapa *syntax* yang lebih efektif untuk digunakan dan mendukung penuh *Object Oriented Programing* (OOP).

MySQL perkembangannya disebut SQL yang merupakan kepanjangan dari *Structured Query Language*. SQL merupakan bahasa terstruktur yang khusus digunakan untuk mengolah *database*. Kita dapat memanfaatkan MySQL untuk menambah, mengubah dan menghapus data yang berada di *database*. MySQL merupakan sistem manajemen *database* yang bersifat *at Relational* yang artinya data-data yang dikelola dalam *database* akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi data akan menjadi jauh lebih mudah dan cepat.

PhpMyAdmin merupakan perangkat lunak untuk mengelola *database* dalam MySQL, PhpMyAdmin memberikan kemudahan untuk membuat tabel, mengisi, mengubah dan lain-lain tanpa harus menghafal perintahnya.

6. API (*Application Program Interface*)

Application Programming Interface adalah kumpulan fungsi atau kumpulan kode program yang berfungsi mengomunikasikan sebuah program dengan dengan *kernel* dari sebuah sistem operasi. API mikrotik adalah kumpulan fungsi, layanan/*service* yang digunakan untuk mengomunikasikan data yang ada pada Mikrotik RouterOS™ untuk dapat dihubungkan menggunakan aplikasi bahasa pemrograman tertentu untuk menyimpan data maupun digunakan untuk mengakses mikrotik itu sendiri. Cara menggunakan API adalah dengan cara mengimpor *package/class import* yang ada dan mengaktifkan *service API* pada mikrotik.

7. Sejarah Singkat Pondok Pesantren Al-Luqmaniyyah

Pondok Pesantren Al-Luqmaniyyah merupakan suatu pesantren yang berbasiskan salaf (kesederhanaan) yang terletak tepatnya di pusat kota Yogyakarta. Pondok Pesantren Al-Luqmaniyyah mulai dibangun pada tahun 1998 M atas prakarsa H. Lukman Jamal Hasibuan, seorang pengusaha kelahiran Sumatera yang telah menetap di Yogyakarta, dan selesai akhir tahun 1999 M. Kemudian diresmikan pada tanggal 9 Februari 2000 M oleh KH. Salimi, seorang tokoh agama asal Mlangi Sleman, dengan nama Pondok Pesantren Salaf Putra Putri Asrama Perguruan Islam (API) “Al-Luqmaniyyah”. Penamaan ini diambil dari nama pendiri, yaitu Bapak H. Lukman.

Selanjutnya, Pondok Pesantren Al Luqmaniyyah diasuh oleh KH. Najib Salimi selama kurang lebih 11 tahun yakni dari tahun 2000 M sampai dengan tahun 2011 M, dan sepeninggal beliau yakni tepatnya pada tanggal 02 Dzulqo’dah 1432 H / 30 September 2011 M, Pondok Pesantren. Al-Luqmaniyyah diasuh oleh istri beliau yakni Nyai Hj. Siti Chamnah Najib dengan di bantu oleh sanak keluarga beliau. Nyai Hj. Siti Chamnah Najib merupakan putri dari KH. Chudlori Abdul Aziz, Pengasuh Pondok Pesantren Al Anwar, Ngrukem, Bantul, DI Yogyakarta. Saat ini, pengasuh Pondok Pesantren Al-Luqmaniyah telah dikaruniai 2 orang putra dan seorang putri.

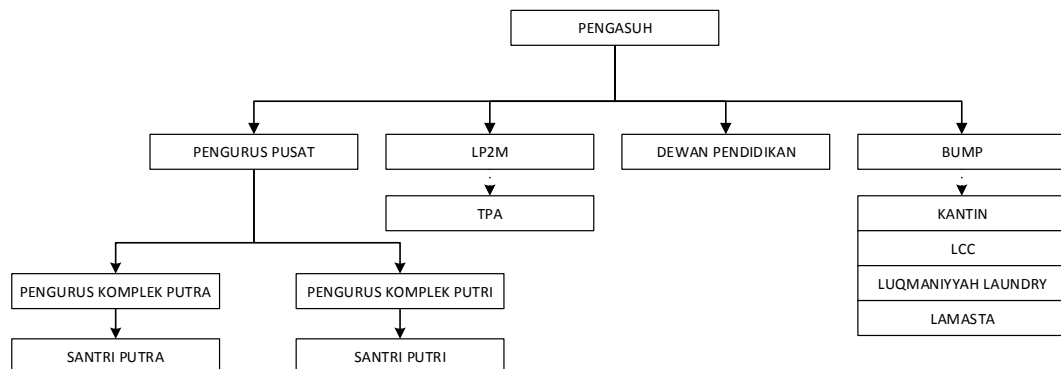
Dari segi materi pendidikan, Pondok Pesantren Al-Luqmaniyyah memiliki karakter yang mirip dengan sistem yang dipakai di API (Asrama

Perguruan Islam) Tegalrejo, Magelang. Sebagai salah satu contoh, Pondok Pesantren Al-Luqmaniyyah sangat menganjurkan para santrinya untuk mujahadah dan riyadloh sebagai sarana untuk mempersiapkan diri menerima ilmu yang bermanfaat. Setiap setelah maghrib dan sebelum subuh selalu terdengar lantunan dzikir mujahadah di masjid untuk santri putra dan aula untuk santri putri Pondok Pesantren Al-Luqmaniyyah.

a. Struktur Organisasi Pondok Pesantren Al-Luqmaniyyah

Susunan pengurus atau struktur organisasi merupakan susunan komponen-komponen (unit-unit kerja) dalam organisasi yang saling berinteraksi dan menunjukkan adanya pembagian kerja, pengelompokan-pengelompokan tugas, serta fungsi-fungsi dalam kegiatan yang berbeda-beda. Fungsi dari struktur organisasi adalah mengoordinasikan bagaimana pekerjaan dibagi, dikelompokkan, dan dikoordinasikan secara formal. Selain itu struktur organisasi juga menunjukkan tentang spesialisasi pekerjaan, saluran perintah, dan penyampaian laporan mulai dari pusat sampai dengan seluruh elemen pesantren. Pondok Pesantren Al-Luqmaniyyah Yogyakarta memiliki struktur organisasi meliputi Pengasuh Pondok Pesantren sebagai jabatan tertinggi yang memegang seluruh wewenang dalam ruang lingkup Pondok Pesantren Al-Luqmaniyyah. Di bawah pengasuh ada pengurus pusat, LP2M (Lembaga Pengabdian dan Pengembangan Masyarakat), dewan pendidikan dan BUMP (Badan Usaha Milik Pesantren). Untuk lebih jelasnya struktur organisasi kepengurusan

pada Pondok Pesantren Al-Luqmaniyyah Yogyakarta dapat dipahami dari susunan struktur organisasi berikut:

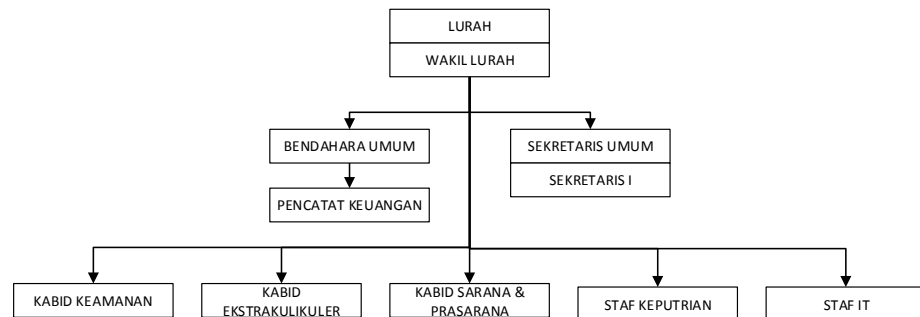


Gambar 2.1. Struktur Pengurus Pondok Pesantren Al-Luqmaniyyah

Secara global pengurus Pondok Pesantren Al-Luqmaniyyah tersusun seperti pada Gambar 2.1 dimana posisi pengasuh sebagai poros dari semua kepengurusan yang ada di Pondok Pesantren Al-Luqmaniyyah. Pengasuh membawahi Pengurus Pusat, LP2M (Pembaga Pengabdian dan Pengembangan Masyarakat), Dewan Pendidikan dan BUMP (Badan Usaha Milik Pesantren). Pengurus Pusat sejajar dengan LP2M (Lembaga Pengabdian dan Pengembangan Masyarakat), Dewan Pendidikan dan BUMP (Badan Usaha Milik Pesantren) yang dikoordinasi langsung oleh Pengasuh Pondok Pesantren Al-Luqmaniyyah.

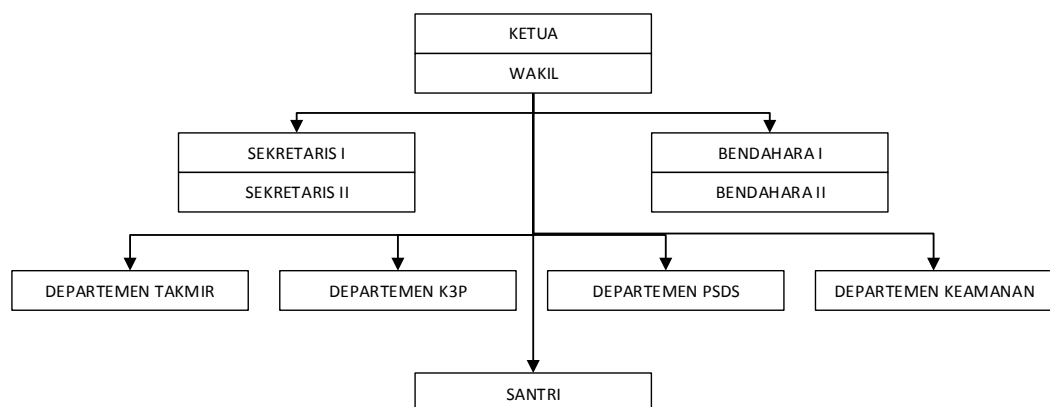
Pengurus Pusat mempunyai posisi membawahi pengurus Komplek Putra dan Komplek Putri. Sedangkan LP2M (Lembaga Pengabdian dan Pengembangan Masyarakat) mempunyai posisi untuk mengkoordinir TPA (Taman Pendidikan Al-Qur'an) yang ada di Pondok Pesantren Al-Luqmaniyyah. Kemudian Dewan Pendidikan

dan BUMP (Badan Usaha Milik Pesantren). Adapun usaha milik BUMP (Badan Usaha Milik Pesantren) adalah Kantin, LCC (Luqmaniyyah Copy Center), Luqmaniyyah *Laundry* dan LAMASTA *Mart*.



Gambar 2.2. Pengurus Pusat Pondok Pesantren Al-Luqmaniyyah

Pada Gambar 2.2 merupakan struktur kepengurusan pusat yang ada di Pondok Pesantren Al-Luqmaniyyah, dimana lurah sebagai poros organisasi yang membawahi langsung bendahara umum, sekretaris umum, kepala bidang keamanan, kepala bidang ekstrakurikuler, kepala bidang sarana dan prasarana, staf keputrian dan staf IT.



Gambar 2.3. Struktur Kepengurusan Komplek

Pada Gambar 2.3. Struktur Kepengurusan Komplek Gambar 2.3 merupakan struktur kepengurusan kompleks dari Pondok Pesantren Al-Luqmaniyyah. Dalam kepengurusan kompleks ini diterapkan kepengurusan tersendiri agar koordinasi dengan pengurus pusat lebih mudah. Dalam struktur tersebut jabatan tertinggi adalah ketua sebagai penanggungjawab keberlangsungan aktivitas kompleks, dibantu oleh sekretaris, bendahara dan departemen-departemen yang ada dalam struktural pengurus kompleks. Adapun struktur kepengurusan tersebut berlaku dan diterapkan di kompleks putra maupun kompleks putri.

b. Tugas dan Wewenang Pengurus Pondok Pesantren Al-Luqmaniyyah

Pada struktur organisasi Pondok Pesantren Al-Luqmaniyyah tiap-tiap jabatan memiliki fungsi tugas dan wewenang yang berbeda-beda, fokus tugas dan wewenang berada di Pengurus Pusat. Adapun masing-masing jabatan adalah sebagai berikut:

1) Pengasuh

Pengasuh pada Pondok Pesantren Al-Luqmaniyyah merupakan bagian dari struktur yang memiliki tugas dan wewenang untuk:

- a) Mengesahkan calon santri yang akan masuk ke Pondok Pesantren sebelum calon santri melakukan konfirmasi terhadap pengurus pusat.
- b) Mengadakan pengawasan terhadap seluruh aktivitas dalam Pondok Pesantren, baik secara langsung maupun secara tidak langsung.

- c) Memberi masukan terhadap masalah yang dihadapi oleh perangkat organisasi untuk mendapatkan solusi dari permasalahan.
 - d) Menerima laporan kegiatan yang telah dilaksanakan dan mengevaluasinya demi kelancaran kegiatan yang akan datang.
- 2) LP2M (Lembaga Pengabdian dan Pengembangan Masyarakat)
- a) Menjalin silaturahmi dengan warga sekitar pondok, dengan menghadiri acara-acara yang ada di rumah warga sekitar.
 - b) Membuat program pengabdian kepada masyarakat selama bulan ramadhan dengan menerjunkan santri untuk ikut berperan aktif pada kegiatan masjid di masyarakat sesuai lokasi yang telah ditentukan.
 - c) Menjaga hubungan dengan santri maupun alumni Pondok Pesantren Al-Luqmaniyyah.
- 3) Dewan Pendidikan
- a) Membuat kurikulum pendidikan di Pondok Pesantren Al-Luqmaniyyah.
 - b) Membuat jadwal mata pelajaran.
 - c) Mengadakan ujian dan munaqosah
- 4) BUMP (Badah Usaha Milik Pesantren)

Mengelola dan memantau berjalannya usaha yang dimiliki pesantren seperti Kantin, LCC (Luqmaniyyah Copy Center), Luqmaniyyah *Laundry* dan LAMASTA *Mart*.

5) Lurah

Dalam menjalankan tugas, lurah dalam organisasi ini memiliki tugas dan wewenang sebagai berikut:

- a) Membawahi semua bagian kepengurusan yang ada pada Pondok Pesantren.
- b) Melakukan koordinasi kepada semua pengurus
- c) Mengadakan rapat program kerja pengurus, dan melakukan rapat evaluasi.
- d) Bertanggungjawab terhadap hubungan eksternal kepengurusan.
- e) Membentuk panitia even (event organization).
- f) Mengadakan rapat koordinasi (dengan pengurus putri/ Dewan ustadz/ Pengasuh).
- g) Melakukan *reshuffle* pengurus yang nonaktif.
- h) Menjembatani jalanya koordinasi kepengurusan putra dan putri.

6) Wakil Lurah

Adapun tugas dan wewenang dari seorang wakil lurah adalah:

- a) Membantu kinerja ketua Umum (Lurah).
- b) Menggantikan posisi sebagai ketua bila ketua umum berhalangan.

7) Bendahara Umum

- a) Melakukan pelaporan keuangan.

- b) Mencatat setiap transaksi keuangan dan melakukan pendokumentasian.
 - c) Menertibkan dan berusaha memaksimalkan pemasukan yang berasal dari syahriah santri.
 - d) Melakukan estimasi anggaran dan pemetaan aliran kas.
 - e) Melakukan pengendalian terhadap aliran kas.
 - f) Melakukan pengejaran aset keuangan yang tertahan di santri baik yang masih berada di pondok ataupun yang sudah keluar.
 - g) Menertibkan pemasukkan keuangan berupa pajak penggunaan aliran listrik pondok.
 - h) Menangani registrasi santri.
 - i) Mempersiapkan dana untuk perbaikan dan perluasan fisik Pesantren.
 - j) Rapat Koordinasi Internal
- 8) Sekretaris Umum
- a) Bertanggung jawab terhadap inventaris kantor.
 - b) Melakukan sensus penghuni kamar santri.
 - c) Perawatan, perapihan, dan penertiban dokumen pesantren.
 - d) Mengagendakan Rapat koordinasi.
 - e) Pendataan identitas santri baru dalam buku induk.
 - f) Pembuatan KTS (Kartu Tanda Santri).
 - g) Membuat agenda surat masuk dan surat keluar.
 - h) Rapat Koordinasi Internal.

9) Sekretaris I

- a) Membantu kinerja Sekretaris Umum.
- b) Menggantikan posisi sebagai Sekretaris Umum bila ketua sekretaris berhalangan.

10) Kepala Bidang Keamanan

Tugas dan wewenang dari kepala bidang keamanan adalah:

- a) Mengawasi seluruh aktivitas kegiatan yang ada dalam Pondok Pesantren Al-Luqmaniyyah.
- b) Memberi masukan dan nasihat terkait usulan-usulan program yang ada sebelum diserahkan kepada pengasuh Pondok Pesantren Al-Luqmaniyyah Yogyakarta.
- c) Mengklarifikasi dan menginterogasi perangkat atau anggota elemen Pondok Pesantren yang menyimpang dari ketentuan yang telah dibuat serta memberi hukuman terhadap perangkat atau anggota elemen Pondok Pesantren jika terbukti melanggar ketentuan yang berlaku.

11) Kepala Bidang Ekstrakurikuler

- a) Menggali dan mengembangkan kreatifitas santri.
- b) Menghidupkan dan mendukung halaqah-halaqah intelektual.
- c) Bertanggung jawab terhadap hidup dan matinya kegiatan ekstra pondok.
- d) Menjembatani Badan ekstra kepada pengurus

12) Kepala Bidang Sarana dan Prasarana

- a) Mendata Perlengkapan/inventaris Pondok Pesantren Al-Luqmaniyyah.
- b) Bertanggungjawab terhadap Kebersihan, keindahan, dan kesehatan Pondok Pesantren Al-Luqmaniyyah.
- c) Pengadaan fasilitas kesehatan Pondok Pesantren Al-Luqmaniyyah.
- d) Kerjasama dengan pihak depkes (pukesmas).
- e) Rapat Koordinasi Internal

13) Pencatat Keuangan

- a) Mencatat segala aktivitas keuangan harian maupun aksidental. Baik itu pemasukan dan pengeluaran.

14) Staf Keputrian

Tugas dari staf keputrian adalah menjalin koordinasi antara pengurus pusat dengan pengurus putri.

15) Staf IT

- a) Mengelola semua media online yang dimiliki oleh Pondok Pesantren Al-Luqmaniyyah.
- b) Menerima, memprioritaskan dan menyelesaikan permintaan bantuan IT di Pondok Pesantren Al-Luqmaniyyah.
- c) Instalasi, perawatan dan penyediaan dukungan harian baik untuk hardware maupun software, peralatan termasuk printer, scanner, dan lain-lain.

- d) Menyediakan data atau informasi yang dibutuhkan untuk pembuatan laporan.

16) Ketua Komplek

17) Departemen Kebersihan, Keindahan, Kesehatan dan Perlengkapan (K3P).

- a) Mendata Perlengkapan/inventaris Pondok Pesantren Al-Luqmaniyyah.
- b) Bertanggungjawab terhadap Kebersihan, keindahan, dan kesehatan Pondok Pesantren Al-Luqmaniyyah.
- c) Pengadaan fasilitas kesehatan Pondok Pesantren Al-Luqmaniyyah.
- d) Kerjasama dengan pihak depkes (pukesmas).
- e) Rapat Koordinasi Internal.

18) Departemen Takmir.

- a) Penjadwalan kegiatan keagamaan.
- b) Bertanggungjawab terhadap fasilitas masjid.
- c) Bertanggung jawab atas pelaksanaan Mujahadah.
- d) Rapat Koordinasi Internal.

19) Departemen Keamanan dan Ketertiban.

- a) Penertiban santri terhadap kegiatan pondok.
- b) Bertanggungjawab terhadap birokrasi dan administrasi perizinan.
- c) Penertiban Jaga malam.

d) Operasi Santri dan sidang pelanggaran.

e) Rapat Koordinasi Internal.

20) Departemen Pengembangan Sumber Daya Santri (PSDS).

a) Menggali dan mengembangkan kreatifitas santri.

b) Menghidupkan dan mendukung halaqah-halaqah intelektual.

c) Bertanggung jawab terhadap hidup dan matinya kegiatan ekstra pondok.

d) Menjembatani Badan ekstra kepada pengurus.

BAB III

METODE PENELITIAN

A. Obyek Penelitian

Subyek penelitian ini adalah membuat Sistem Manajemen *User Hotspot* yang berada di Pondok Pesantren Al-Luqmaniyyah menggunakan MikroTik PHP API. Dengan mengumpulkan data dari pondok pesantren untuk kemudian diproses menjadi bahan penelitian, sistem ini diharapkan dapat membantu mempermudah dalam mengelola *user* pada *hotspot* yang ada di Pondok Pesantren Al-Luqmaniyyah.

B. Metode Pengumpulan Data

Teknik pengumpulan data yang digunakan dalam penyusunan skripsi ini adalah :

1. Studi Pustaka

Dalam metode ini, penulis mempelajari buku-buku yang bersangkutan sebagai dasar dan acuan dalam memahami MikroTik RouterOS. Selain itu, juga memanfaatkan internet sebagai sumber data guna menambah referensi terkait penggunaan API dalam *router* Mikrotik.

2. Observasi

Metode pengumpulan data dengan cara mengamati dan menganalisa guna mendapatkan data-data sebagai penunjang perancangan aplikasi pihak ketiga untuk mempermudah manajemen *user* pada *router* MikroTik yang berada di Pondok Pesantren Al-Luqmaniyyah.

3. Studi dan *Interview* (Wawancara)

Merupakan suatu pengumpulan data yang dilakukan dengan cara tanya jawab atau dialog secara langsung dengan pihak-pihak yang terkait dengan penelitian yang akan dilakukan. Dalam hal ini penulis melakukan tanya jawab kepada *admin* yang mengelola jaringan di Pondok Pesantren Al-Luqmaniyyah.

C. Alat-alat Penelitian

Alat-alat yang digunakan dalam penelitian ini terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*software*) dengan spesifikasi dan kegunaan masing-masing sebagai berikut:

1. Perangkat Keras (*Hardware*)

- a. Processor : AMD E-450 APU Radeon (2 CPUs), ~1.6 GHz.
- b. RAM : 4 GB.
- c. VGA : AMD Radeon HD 6320 Graphics.
- d. Harddisk : 500 GB.
- e. *Router Board* : MikroTik RB951Ui-2HnD.

2. Perangkat Lunak (*Software*)

- a. *Router OS* : MikroTik versi 6.38 Level 4.
- b. Sistem Operasi : Windows 7 Ultimate 64 bit.
- c. Design Diagram : Microsoft Visio 2013.
- d. Wireframe : Balsamiq 3.2.3
- e. Text Editor : Sublime Text 3.
- f. *Server Lokal* : XAMPP 1.8.3.

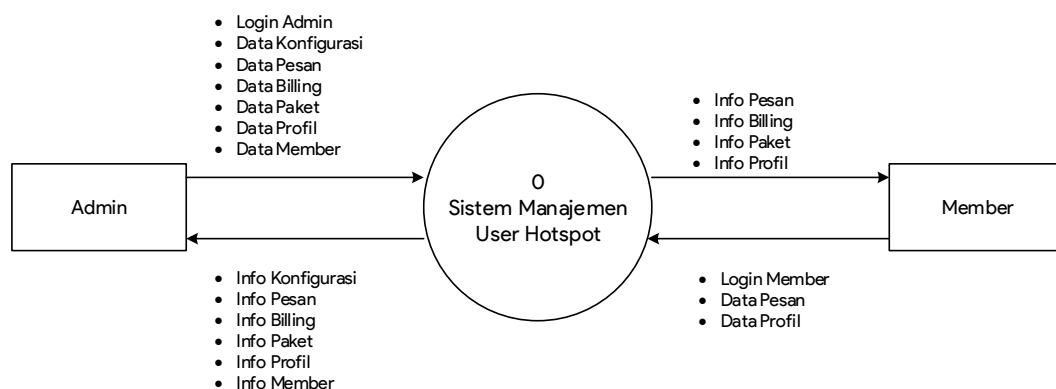
D. Perancangan Sistem

Perancangan sistem merupakan langkah awal sebelum dilaksanakan penyelesaian terhadap suatu masalah yang ada, untuk dapat mencapai tujuan atau hasil yang memadai sesuai kebutuhan dari permasalahan yang ada, maka diperlukan suatu rancangan sistem yang dapat menggambarkan secara garis besar seluruh masalah yang akan dikomputerisasikan.

Perancangan sistem diperlukan untuk mempermudah proses pengembangan sistem yang akan dibuat menurut kebutuhan. Perancangan sistem meliputi perancangan proses maupun *design* sistem yang akan dibuat.

1. DFD (Data Flow Diagram)

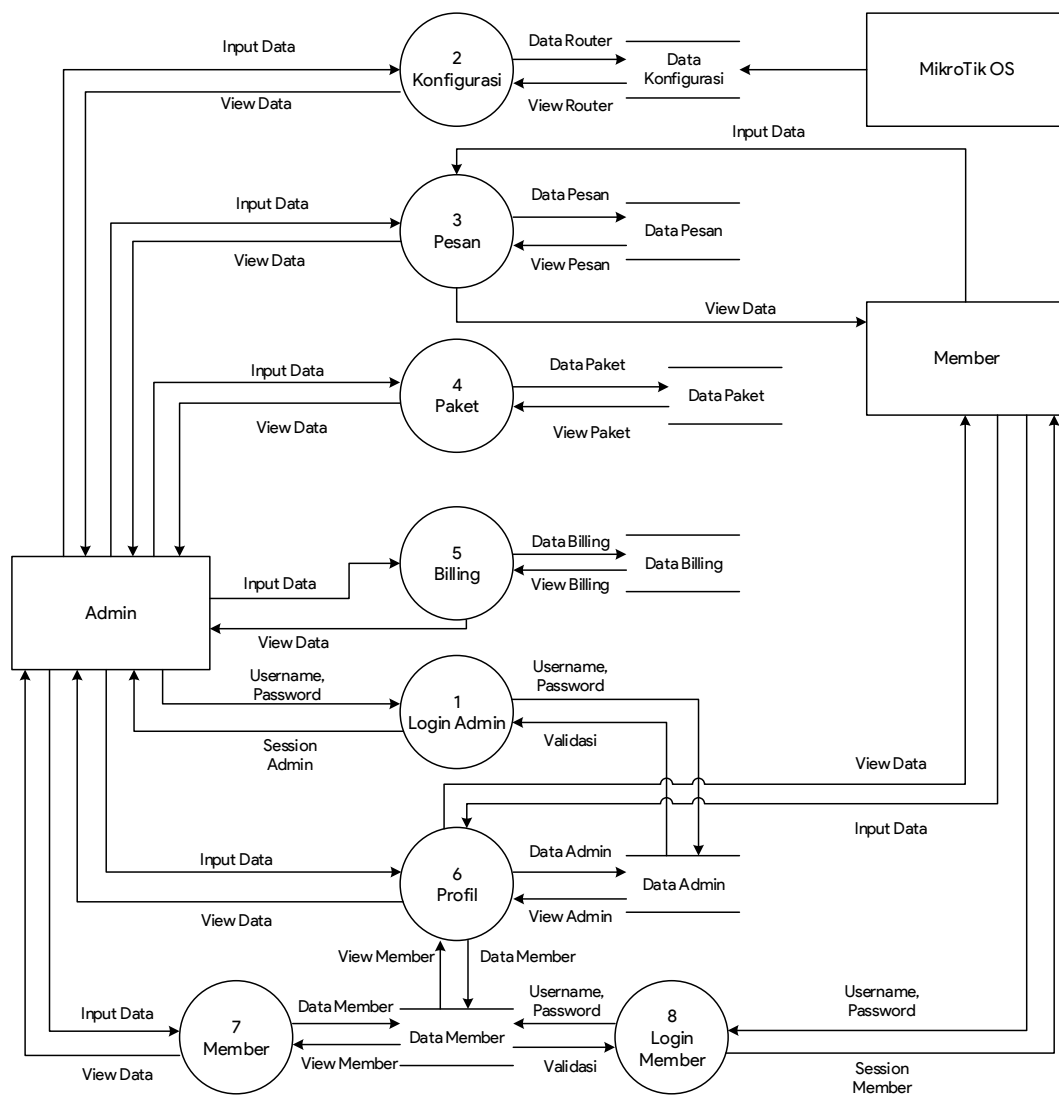
Data Flow Diagram (DFD) atau disebut dengan Diagram konteks merupakan diagram yang menggambarkan arus data antara sistem dengan pengguna yang berhubungan dengan sistem. Sistem manajemen *user hotspot* ini berhubungan dengan dua jenis pengguna yaitu, *administrator*, dan *member*.



Gambar 3.1. Data Flow Diagram Level 0

Berdasarkan diagram konteks diatas, dapat dijelaskan bahwa ada 2 pihak yang terlibat dalam aplikasi ini, yaitu *Admin* dan *Member*. Dimana *Admin* mempunyai hak akses untuk menambah, mengedit dan menghapus data pada sistem. Sedangkan *member* hanya bisa melihat informasi, mengirim pesan dan mengedit data profil pribadi *member* saja.

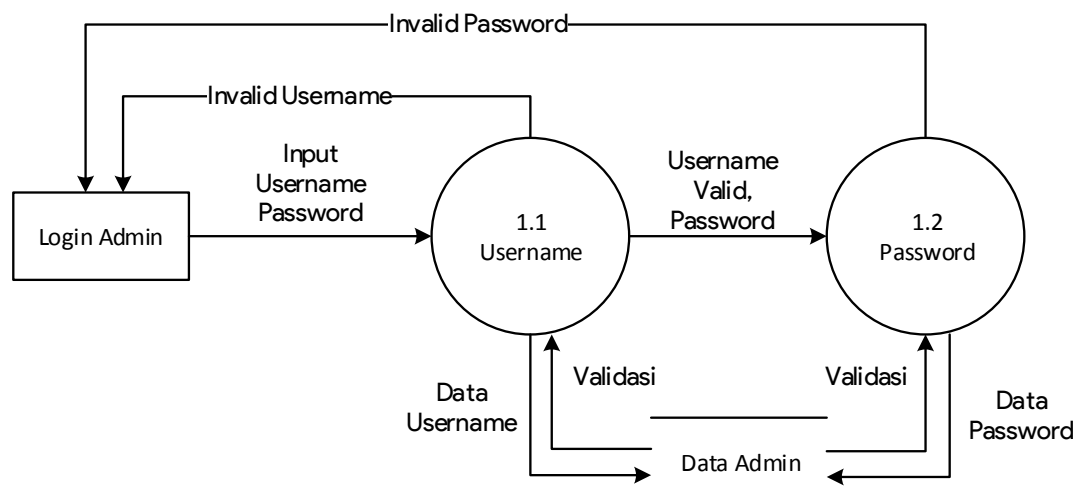
2. Data Flow Diagram Level 1



Gambar 3.2. DFD Level 1 Sistem Manajemen User Hotspot

Berdasarkan diagram konteks diatas, dapat disimpulkan bahwa ada delapan proses, dimana *Login Admin* merupakan proses pertama yang harus dilakukan seorang Administrator untuk dapat menambah data pada sistem ini. Pada gambar tersebut dijelaskan bagaimana seluruh usulan sistem yang akan dibangun.

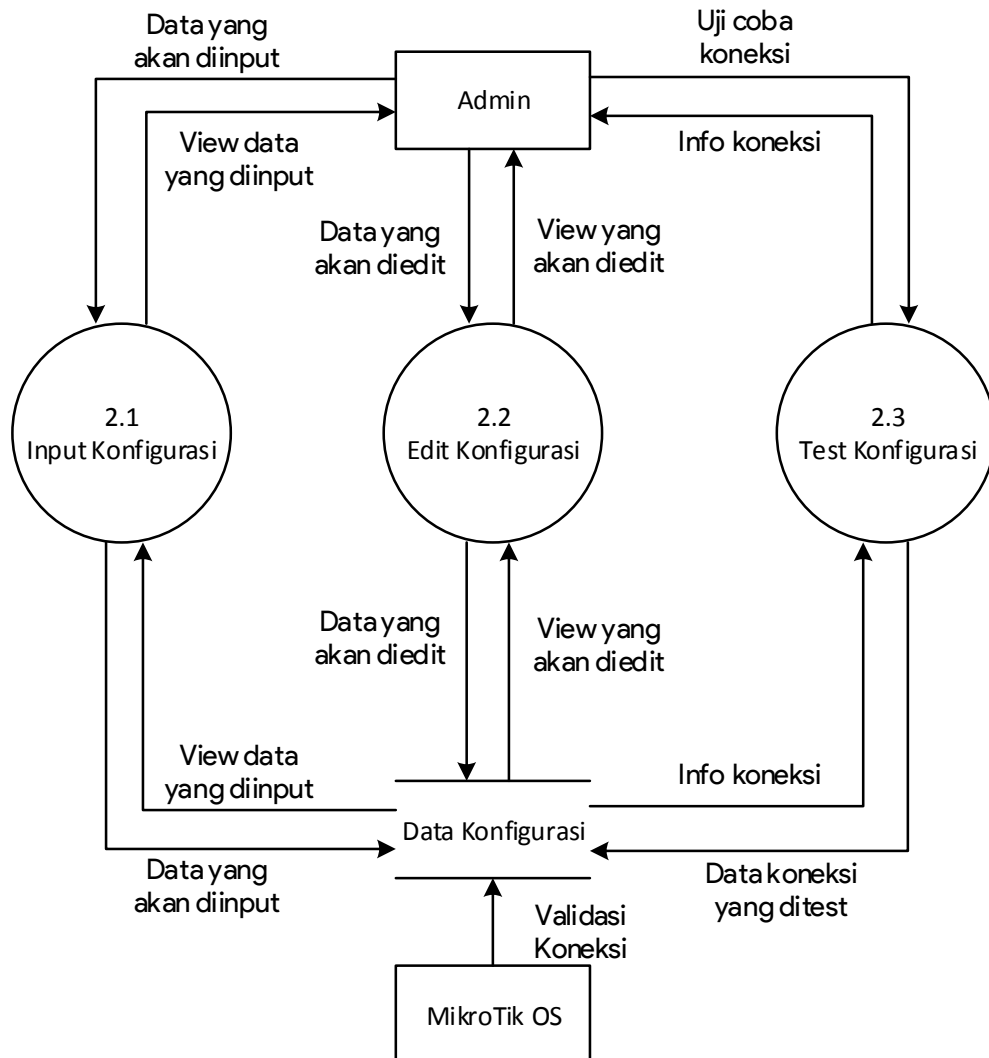
3. Data Flow Diagram Level 2 Proses 1



Gambar 3.3. DFD Level 2 Proses Login Admin

Pada diagram konteks diatas merupakan *Data Flow Diagram Level 2* yang menjelaskan bagaimana proses *login admin*, *Admin* memasukan *Username* dan *Password* untuk dapat masuk ke dalam Web, jika verifikasi *username* dan *password* Valid *admin* dapat memproses untuk mengunggah, penambahan, pengubahan, dan penghapusan data, namun jika *username* dan *password* invalid maka akan kembali ke tampilan awal *Login* untuk memasukan *username* dan *password* kembali.

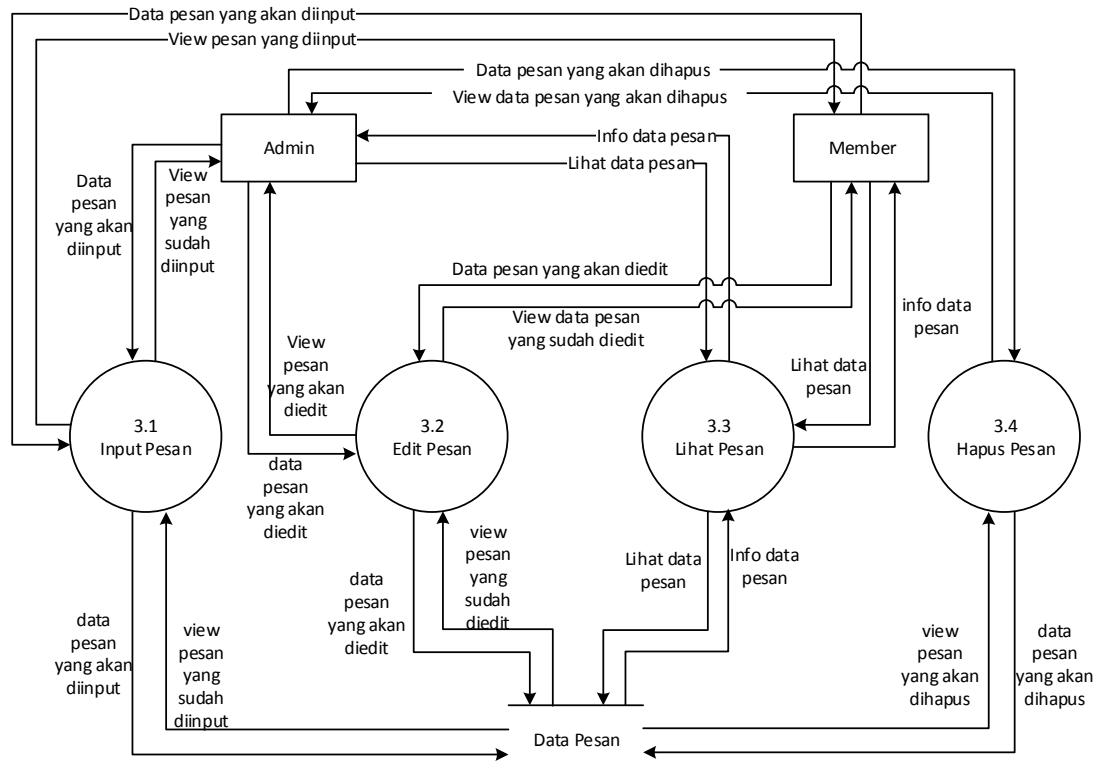
4. Data Flow Diagram Level 2 Proses 2



Gambar 3.4. DFD Level 2 Proses Konfigurasi

Pada diagram konteks diatas merupakan DFD Level 2 Proses Konfigurasi aplikasi terhadap *router* agar saling terhubung. Dimana pada gambar tersebut bisa dijelaskan bagaimana proses *input*, edit dan *test* koneksi pada konfigurasi. Pada proses test koneksi membutuhkan validasi dari MikroTik OS yang diambil dari data konfigurasi. Proses ini hanya bisa dilakukan oleh *admin* dengan melakukan proses *login* terlebih dahulu.

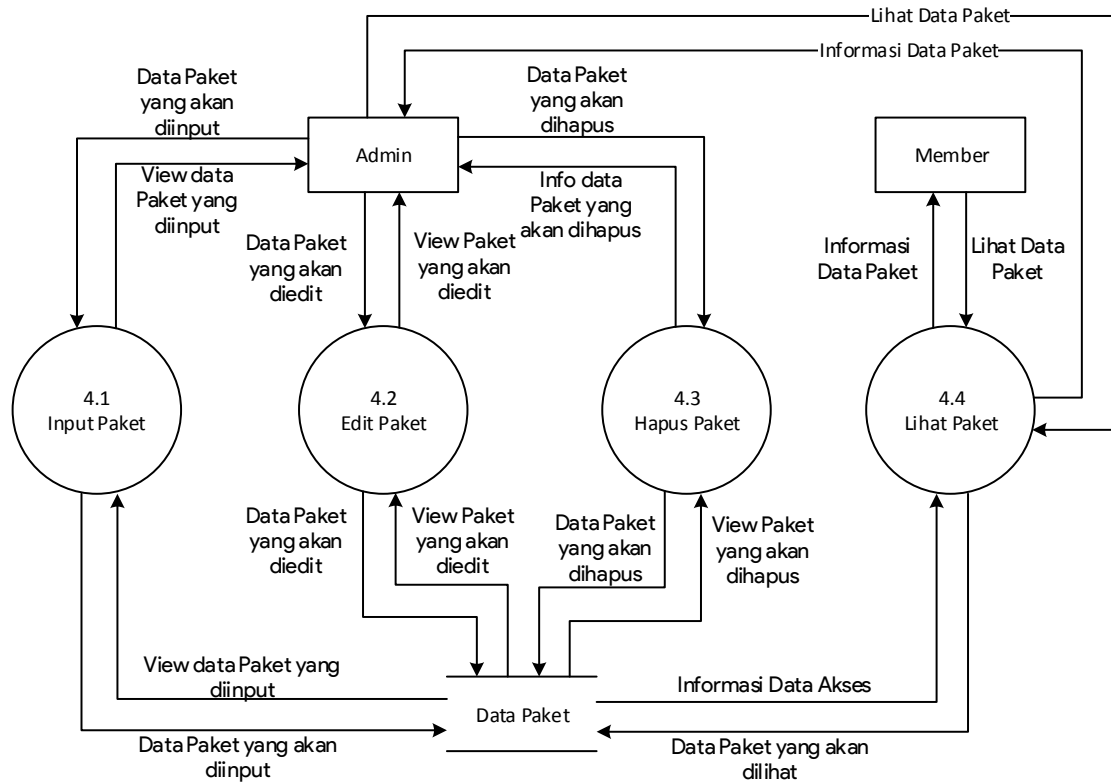
5. Data Flow Diagram Level 2 Proses 3



Gambar 3.5. DFD Level 2 Proses Pesan

Pada diagram konteks diatas merupakan proses pengelolaan Data Pesan. Gambar tersebut menjelaskan bagaimana proses *input* data pesan. Untuk melakukan *input* data pesan bisa dilakukan oleh *admin* ke *member* atau sebaliknya dengan melakukan proses *login* terlebih dahulu. Seorang *admin* bisa mengolah data pesan seperti *input* data pesan, Edit data pesan dan hapus data pesan. Sedangkan *member* bisa melakukan seperti *admin* kecuali menghapus data pesan.

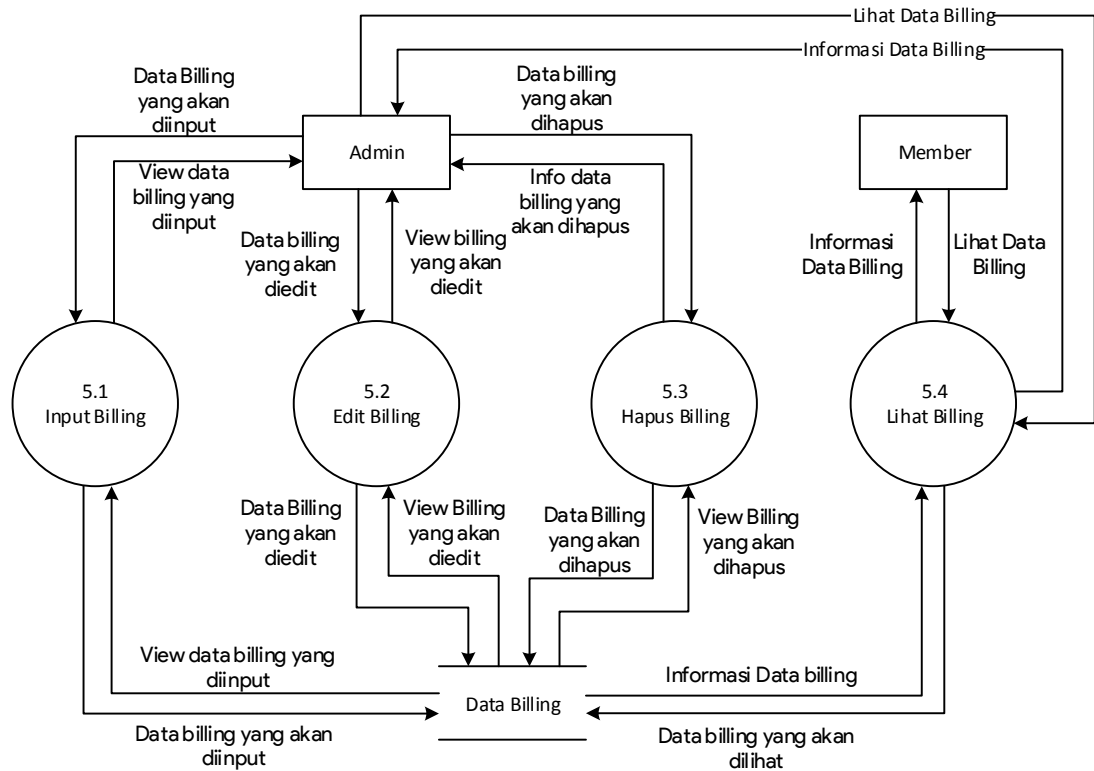
6. Data Flow Diagram Level 2 Proses 4



Gambar 3.6. DFD Level 2 Proses Paket

Pada diagram konteks diatas merupakan proses pengelolaan Data Paket. Dimana dalam gambar tersebut menjelaskan bagaimana proses input data akses. Untuk melakukan pengolahan terhadap data akses hanya bisa dilakukan oleh admin. Sedangkan member hanya bisa melihat data paket yang dipakai.

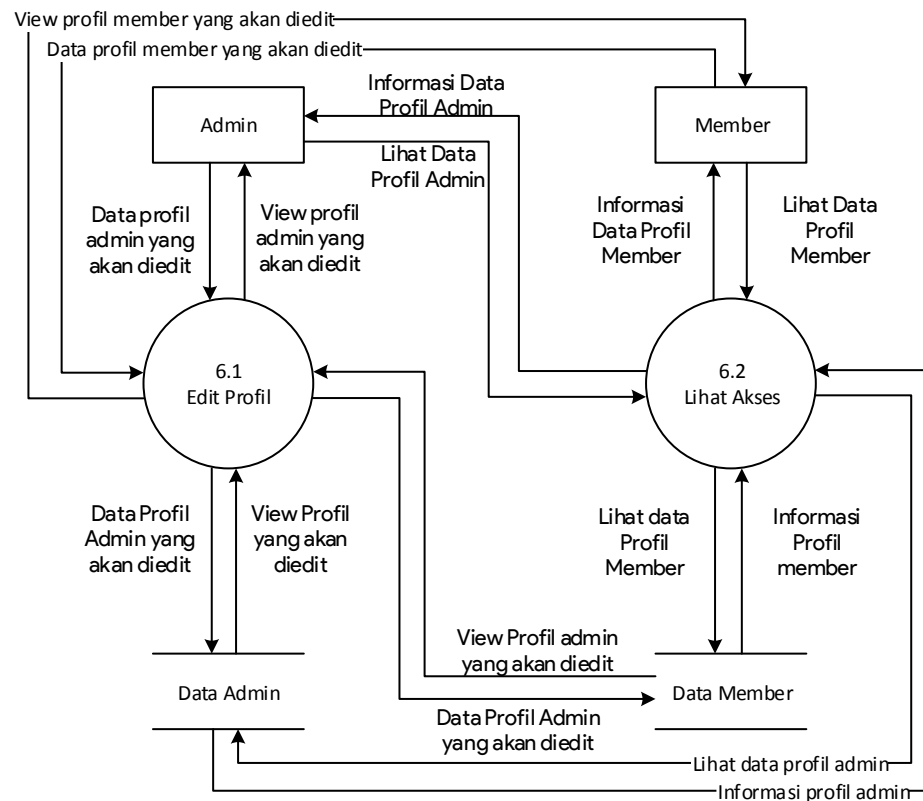
7. Data Flow Diagram Level 2 Proses 5



Gambar 3.7. DFD Level 2 Proses Billing

Pada diagram konteks diatas merupakan proses pengelolaan Data Pesan. Gambar tersebut menjelaskan bagaimana proses *input* data pesan. Untuk melakukan *input* data pesan bisa dilakukan oleh *admin* ke *member* atau sebaliknya dengan melakukan proses *login* terlebih dahulu. Seorang *admin* bisa mengolah data pesan seperti *input* data pesan, Edit data pesan dan hapus data pesan. Sedangkan *member* bisa melakukan seperti *admin* kecuali menghapus data pesan.

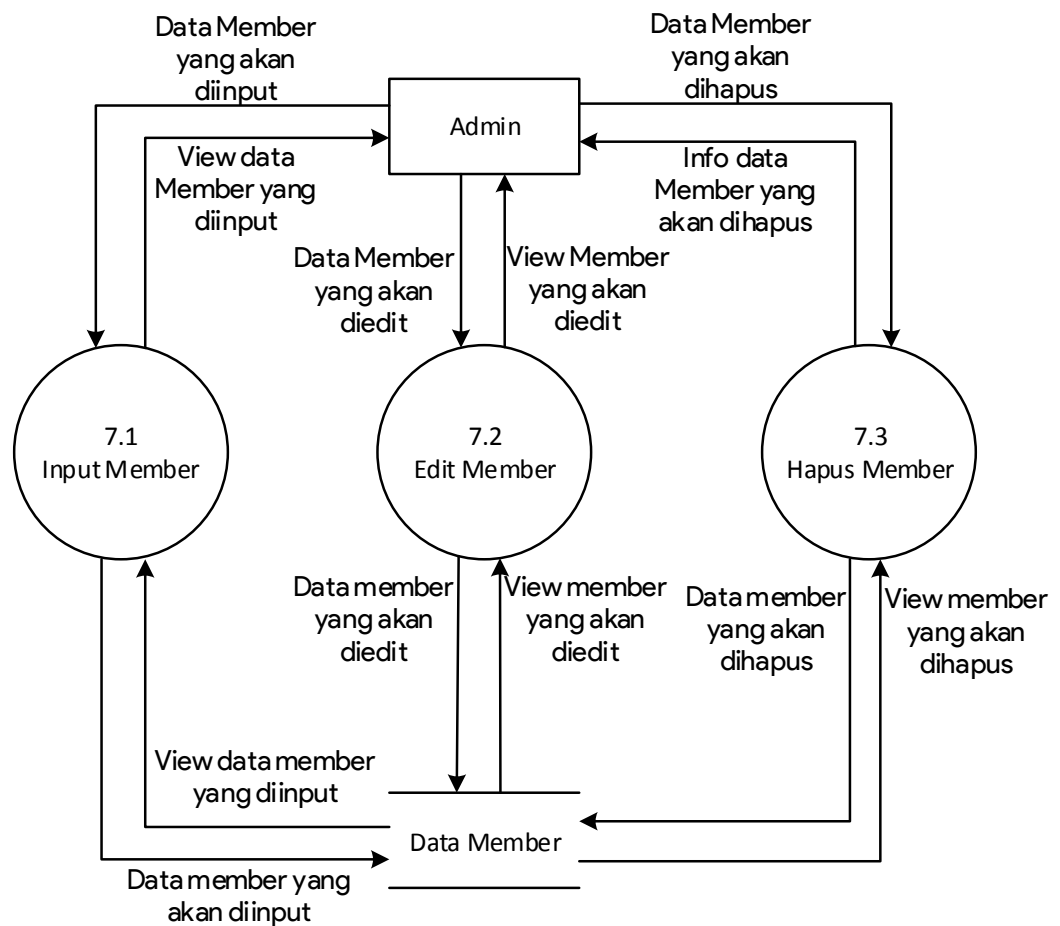
8. Data Flow Diagram Level 2 Proses 6



Gambar 3.8. DFD Level 2 Proses Profil

Diagram konteks diatas merupakan proses pengelolaan Data Pesan. Gambar tersebut menjelaskan bagaimana proses input data pesan. Untuk melakukan input data pesan bisa dilakukan oleh admin ke member atau sebaliknya dengan melakukan proses login terlebih dahulu. Seorang admin bisa mengolah data pesan seperti input data pesan, Edit data pesan dan hapus data pesan. Sedangkan member bisa melakukan seperti admin kecuali menghapus data pesan.

9. Data Flow Diagram Level 2 Proses 7

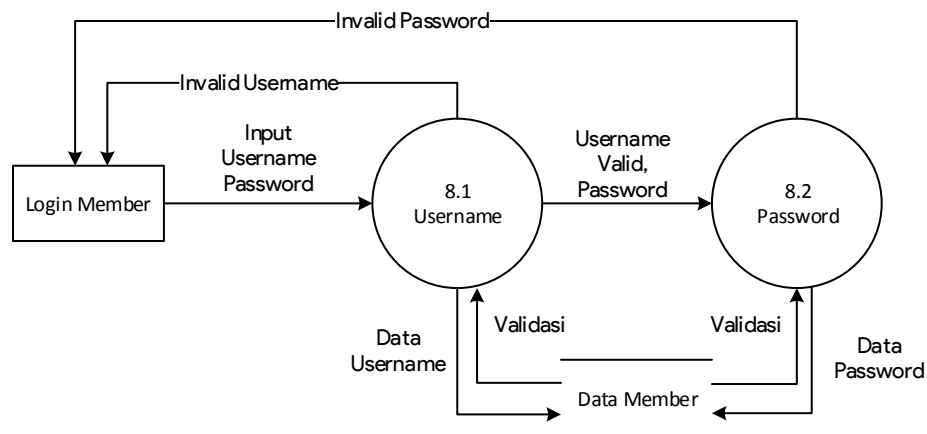


Gambar 3.9. DFD Level 2 Proses Pengolahan Member

Pada diagram konteks diatas merupakan proses pengelolaan Data Pesan. Gambar tersebut menjelaskan bagaimana proses input data pesan. Untuk melakukan input data pesan bisa dilakukan oleh admin ke member atau sebaliknya dengan melakukan proses login terlebih dahulu. Seorang admin bisa mengolah data pesan seperti input data pesan, Edit data pesan

dan hapus data pesan. Sedangkan member bisa melakukan seperti admin kecuali menghapus data pesan.

10. Data Flow Diagram Level 2 Proses 8



Gambar 3.10. DFD Level 2 Proses Login Member

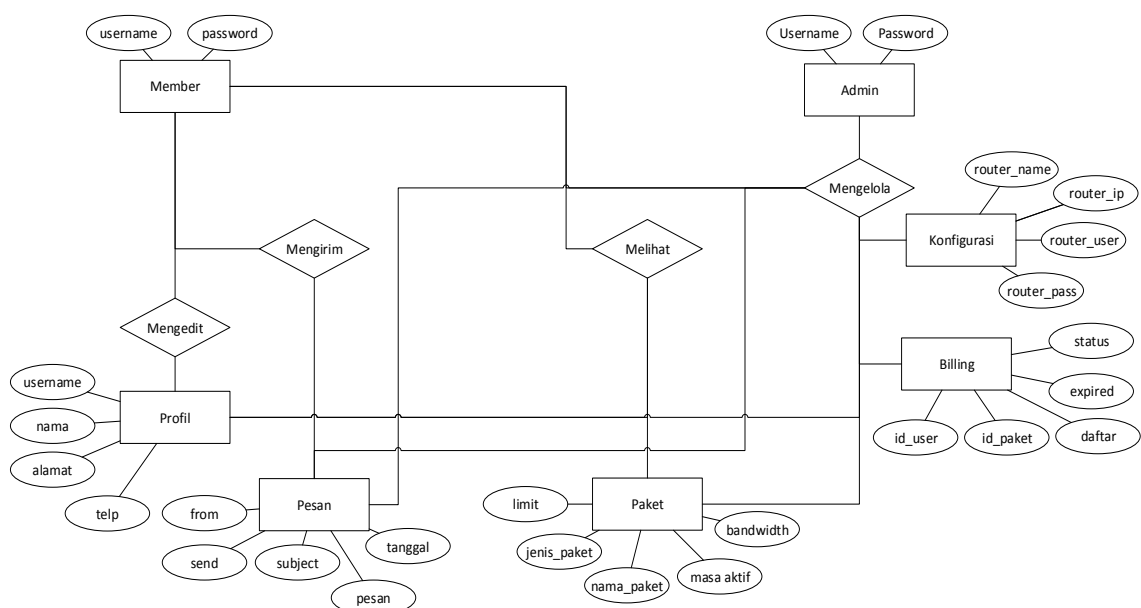
Pada diagram konteks diatas merupakan proses pengelolaan Data Pesan. Gambar tersebut menjelaskan bagaimana proses input data pesan. Untuk melakukan input data pesan bisa dilakukan oleh admin ke member atau sebaliknya dengan melakukan proses login terlebih dahulu. Seorang admin bisa mengolah data pesan seperti input data pesan, Edit data pesan dan hapus data pesan. Sedangkan member bisa melakukan seperti admin kecuali menghapus data pesan.

11. Entity Relationship Diagram

ERD merupakan suatu model jaringan yang menggunakan susunan data yang disimpan pada sistem secara abstrak. ERD digunakan oleh perancang sistem untuk memodelkan data yang nantinya akan dikembangkan menjadi *database*. ERD merupakan model konseptual yang

mendeskripsikan hubungan antara *file* yang digunakan untuk memodelkan struktur data serta hubungan antar data (Yakub, 2008).

ERD terbagi atas 3 komponen yaitu entitas, atribut, relasi. Secara garis besar entitas merupakan dasar yang terlibat dalam sistem. Atribut berperan sebagai penjelas dari entitas dan relasi menunjukkan hubungan yang terjadi antara dua entitas.



Gambar 3.11. Diagram ERD Sistem Manajemen User Hotspot

E. Perancangan Tabel Basis Data

Basis data berfungsi untuk menampung data-data yang akan digunakan serta ditampilkan pada sistem manajemen *user hotspot*. Terdiri dari beberapa rancangan tabel sebagai berikut:

1. Tabel Admin

Tabel admin berfungsi sebagai penampung data *Administrator* dalam sistem manajemen *user hotspot*.

Tabel 3. 1. Tabel Admin

Field	Type	Lenght/Value	Index
id_admin	INTEGER	10	Primary
username	VARCHAR	50	
password	VARCHAR	50	
nama	VARCHAR	90	
alamat	TEXT		
telp	VARCHAR	50	

2. Tabel Member

Tabel ini berisi data *member* yang terintegrasi dengan *hotspot*. Dimana *username* dan *password*nya bisa digunakan langsung untuk *login hotspot*.

Tabel 3. 2. Tabel Member

Field	Type	Lenght/Value	Index
id_member	INTEGER	10	Primary
username	VARCHAR	50	
password	VARCHAR	50	
nama	VARCHAR	90	
alamat	TEXT		
telp	VARCHAR	50	

3. Tabel Konfigurasi

Tabel konfigurasi dirancang untuk mengkonfigurasi koneksi aplikasi sistem manajemen *user hotspot* dengan *router* MikroTik.

Tabel 3. 3. Tabel Konfigurasi

Field	Type	Lenght/Value	Index
id_config	INTEGER	10	Primary
router_name	VARCHAR	50	
router_ip	VARCHAR	50	
router_user	VARCHAR	50	
router_pass	VARCHAR	50	

4. Tabel Pesan

Tabel pesan dirancang kepada *member* untuk mengirim kritik, pesan, maupun saran kepada *Administrator* sebagai media dalam berkomunikasi.

Tabel 3. 4. Tabel Pesan

Field	Type	Lenght/Value	Index
id_pesan	INTEGER	10	Primary
id_sender	INTEGER	10	
id_receiver	INTEGER	10	
subject	VARCHAR	50	
pesan	TEXT		
time_pesan	TIMESTAMP		
status	ENUM	('0','1')	
kat	ENUM	('N','B')	

5. Tabel Paket

Tabel paket dirancang untuk mengatur paket yang akan disediakan untuk *member* dalam menggunakan *hotspot*.

Tabel 3. 5. Tabel Paket

Field	Type	Lenght/Value	Index
id_paket	INTEGER	10	Primary
nama_paket	VARCHAR	50	
jenispaket	ENUM	('0','1')	
bandwidth	VARCHAR	50	
masa_aktif	VARCHAR	50	

6. Tabel Billing

Tabel billing dirancang untuk mengetahui masa aktif member, jenis paket yang dipakai.

Tabel 3. 6. Tabel Billing

Field	Type	Lenght/Value	Index
id_billing	INTEGER	10	Primary
id_user	INTEGER	10	
id_paket	INTEGER	10	
daftar	VARCHAR	100	
expire	VARCHAR	100	
status	ENUM	('0','1')	
id_admin	INTEGER	10	

F. Perancangan Antar Muka Sistem

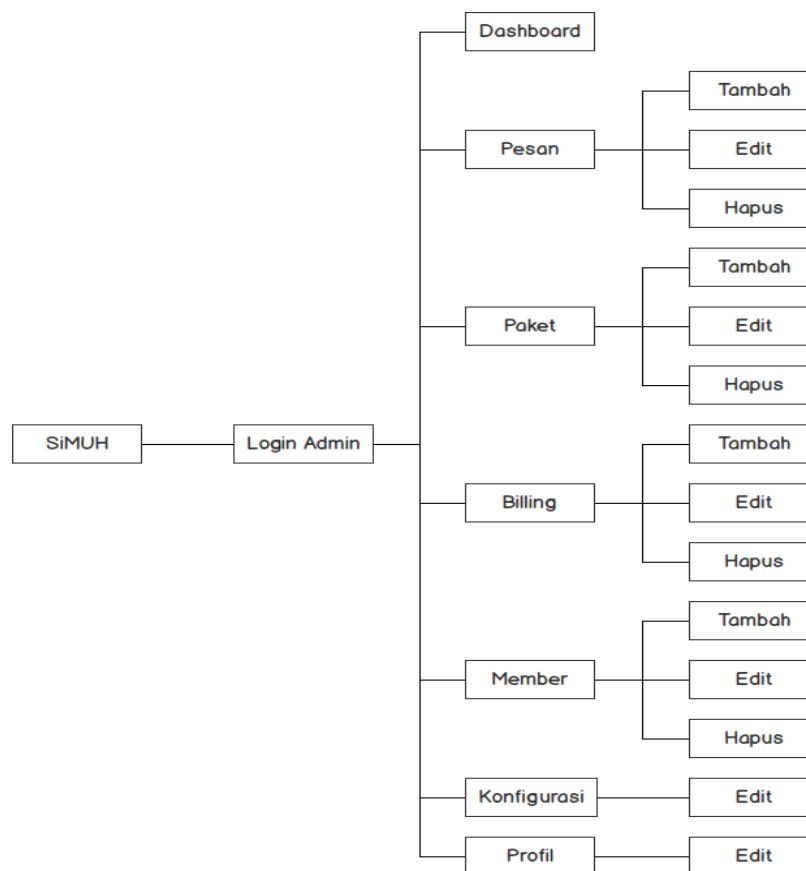
Dalam pembuatan sistem perancangan antarmuka dibuat guna untuk mempermudah dalam pembuatan program. Perancangan antarmuka sistem ini

merupakan salah satu bagian dari pembuatan sistem yang sangat penting. Dalam perancangan antarmuka sistem ini terdiri dari perancangan struktur menu dan perancangan halaman antarmuka sistem.

1. Perancangan Struktur Menu

Struktur menu merupakan suatu rancangan beberapa pilihan menu yang bertujuan untuk memudahkan dalam mengoperasikan program sehingga pengguna tidak mengalami kesulitan untuk memilih menu-menu yang diinginkan sesuai dengan kebutuhannya. Dalam sistem ini terdiri dari struktur menu *Admin* dan *Member*.

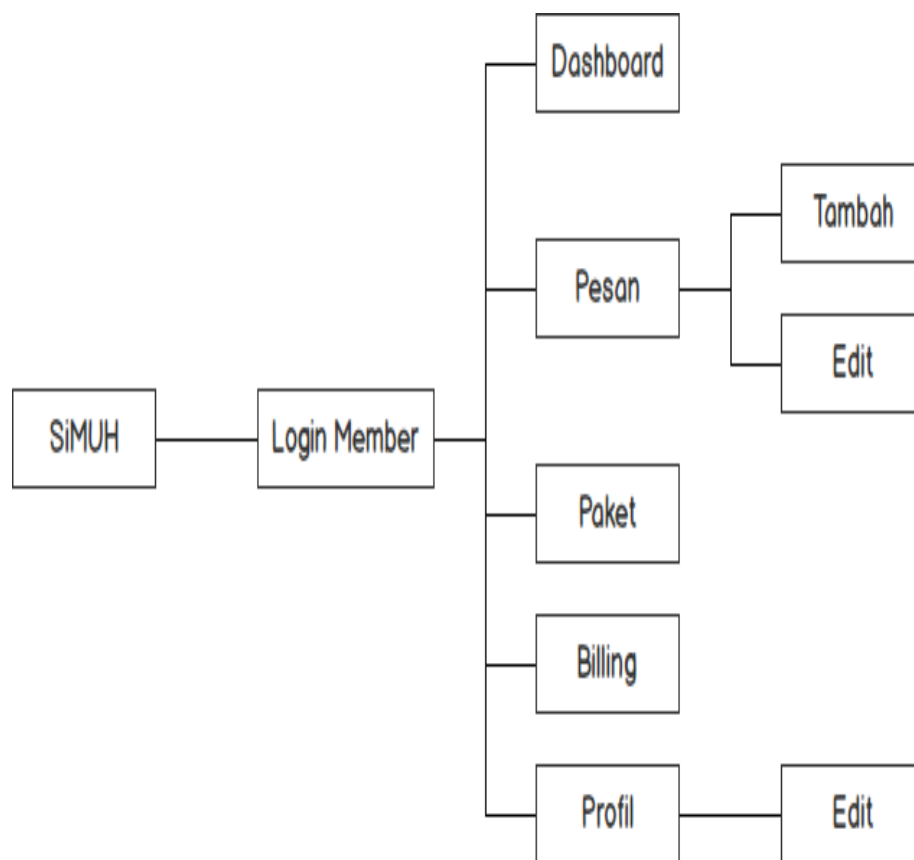
a. Rancangan Struktur Menu *Admin*



Gambar 3.12. Rancangan Struktur Menu Admin

Pada gambar diatas dapat dijelaskan beberapa hak akses yang dimiliki oleh *admini*, dimana keseluruhan sistem bisa diakses oleh user dengan tipe tersebut diantaranya adalah menambah, mengedit maupun menghapus sesuai kriteria yang ada dalam gambar.

b. Rancangan Struktur Menu *Member*



Gambar 3.13. Rancangan Struktur Menu *Member*

Pada gambar diatas merupakan rancangan menu pada halaman member, dimana pengguna dengan kategori member hanya bisa melihat beberapa fitur yang ada dalam sistem. Selebihnya hanya bisa menambah dan mengedit data yang telah disediakan.

2. Perancangan Antarmuka Sistem

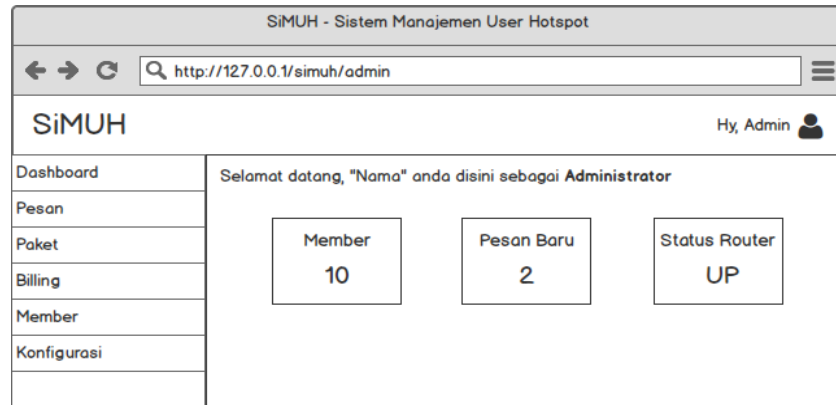
Rancangan Sistem Manajemen *User Hotspot* yang akan diimplementasikan menggunakan rancangan berbentuk *wireframe* dan mengikuti rancangan struktur menu yang sudah ada.

a. Rancangan Halaman *Login System*

Gambar 3.14. Rancangan Halaman *Login System*

Pada gambar diatas merupakan rancangan halaman utama sekaligus untuk *login*. Login dalam sistem ini dibagi menjadi dua bagian *login admin* dan *login member*, karena dalam sistem *login* sendiri dibagi menjadi dua jenis otentikasi, yaitu *member* dengan data POST yang bersifat *plaintext* dan *admin* dengan data POST yang menggunakan *encrypt MD5 hash*.

b. Rancangan Halaman *Dashboard Admin*



Gambar 3.15. Rancangan Halaman *Dashboard Admin*

Gambar diatas merupakan rancangan halaman *dashboard* untuk *admin* setelah *login*. Pada halaman ini menampilkan navigasi menu pada sisi kiri untuk mempermudah *admin* ketika berpindah halaman untuk mengelola sistem. Sedangkan pada sisi kanan untuk mengetahui informasi jumlah *member*, pesan yang baru yang belum dibaca dan status sambungan sistem dengan *router*.

c. Rancangan Halaman *Pesan Admin*



Gambar 3.16. Rancangan Halaman *Pesan Admin*

Gambar diatas merupakan rancangan halaman pesan dengan tampilan dari pesan masuk ketika *admin* memilih menu pesan. Dalam halaman tersebut *admin* bisa mengelola data pesan yang ada, seperti membuat pesan baru, melihat pesan masuk, melihat pesan keluar, dan menghapus pesan.

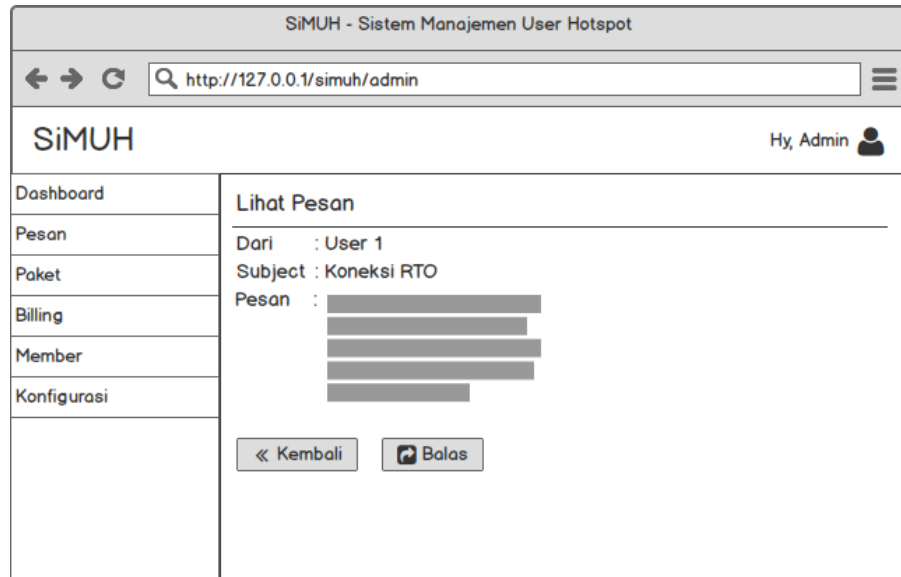
d. Rancangan Halaman Pesan Baru

The screenshot displays the SiMUH web application interface. At the top, the browser address bar shows 'http://127.0.0.1/simuh/admin'. The page header includes 'SiMUH' on the left and 'Hy, Admin' with a user icon on the right. A sidebar menu on the left lists: Dashboard, Pesan, Paket, Billing, Member, and Konfigurasi. The main content area is titled 'Pesan Baru' and contains a form with the following fields: 'Kepada' (To) with a dropdown menu labeled 'Pilih', 'Subject' with a text input field labeled 'Title', and 'Pesan' (Message) with a large text area labeled 'Isi Pesan'. A 'Kirim' (Send) button is located at the bottom of the form.

Gambar 3.17. Rancangan Halaman Pesan Baru

Gambar diatas merupakan rancangan halaman untuk membuat pesan baru. Dimana pada halaman tersebut terdapat *form* kepada, subject, dan isi pesan. Pada *form* kepada merupakan *dropdown* dari daftar *member* yang sudah ada pada sistem kemudian dipilih untuk dikirim pesan.

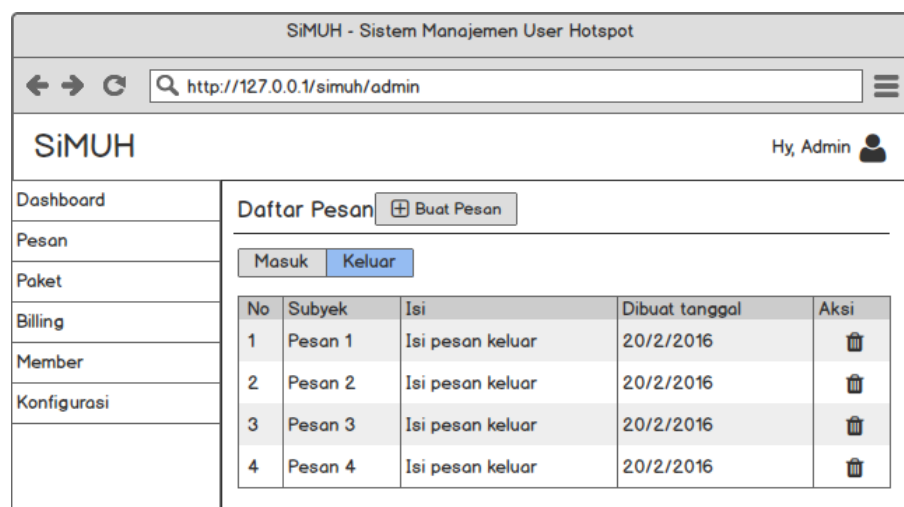
e. Rancangan Halaman Lihat Pesan



Gambar 3.18. Rancangan Halaman Lihat Pesan

Gambar diatas merupakan rancangan halaman untuk melihat pesan secara detil. Pada rancangan tersebut terdapat tombol balas jika ingin membalas pesan yang diterima.

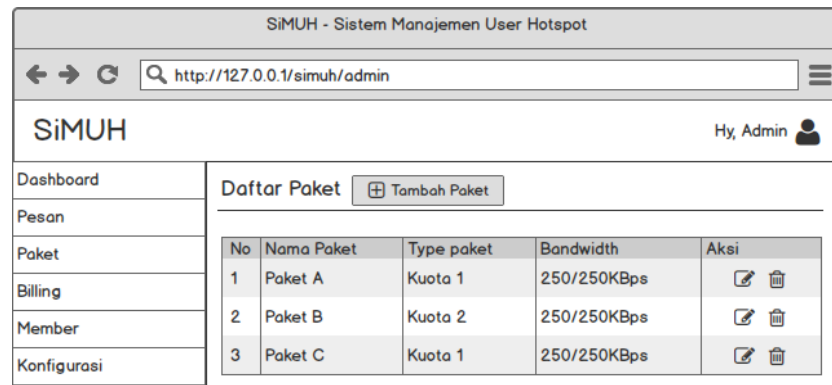
f. Rancangan Halaman Pesan Keluar



Gambar 3.19. Rancangan Halaman Pesan Keluar

Gambar diatas merupakan rancangan halaman untuk menampilkan pesan yang sudah terkirim. Pada halaman ini tersedia tombol hapus jika ingin menghapus pesan.

g. Rancangan Halaman Paket *Admin*

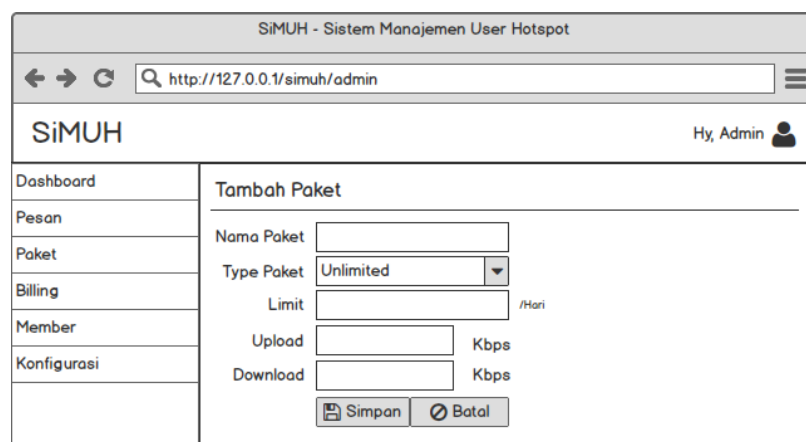


No	Nama Paket	Type paket	Bandwidth	Aksi
1	Paket A	Kuota 1	250/250KBps	
2	Paket B	Kuota 2	250/250KBps	
3	Paket C	Kuota 1	250/250KBps	

Gambar 3.20. Rancangan Halaman Paket Admin

Gambar diatas merupakan rancangan halaman paket, dimana pada halaman ini menampilkan paket-paket yang sudah dibuat oleh admin. Pada halaman ini terdapat tombol-tombol untuk mengelola paket, yaitu tambah paket, edit paket, dan hapus paket.

h. Rancangan Halaman Tambah Paket




Gambar 3.21. Rancangan Halaman Tambah Paket

Gambar diatas merupakan rancangan halaman untuk menambah paket, dimana fitur ini untuk mengklasifikasi kapasitas *bandwidth* untuk disesuaikan untuk *user*.

i. Rancangan Halaman *Billing Admin*

SiMUH

Hy, Admin 

Dashboard

Pesan







Paket

Billing

Member

Konfigurasi

Daftar Billing

No	Username	Paket	Expired	Status	Aksi
1	username2	Kuota A	11/2/2016	Aktif	 
2	username1	Kuota C	11/2/2016	Non-Aktif	 
3	username3	Kuota B	11/2/2016	Aktif	 

Gambar 3.22. Rancangan Halaman *Billing Admin*

Gambar diatas merupakan rancangan halaman *billing* untuk mengetahui info paket, tanggal *expired*, dan status dari *username hotspot member*. Pada halaman ini juga tersedia menu hapus jika *admin* ingin menghapus *billing*, dan menu edit jika *admin* ingin mengedit status dari *username*.

j. Rancangan Halaman Daftar *Member*

SiMUH

Hy, Admin

Dashboard

Pesan

Paket

Billing

Member

Konfigurasi

Daftar Member

Tambah Member

No	Nama	Username	Kontak	Aksi
1	Nama 1	Username 1	1234567890	<div><div></div><div></div><div></div></div>
2	Nama 2	Username 2	1234567890	<div><div></div><div></div><div></div></div>
3	Nama 3	Username 3	1234567890	<div><div></div><div></div><div></div></div>
4	Nama 4	Username 4	1234567890	<div><div></div><div></div><div></div></div>

Gambar 3.23. Rancangan Halaman Daftar *Member*

Gambar diatas merupakan rancangan halaman daftar *member*, dimana pada halaman ini menampilkan data *member* yang ada pada sistem. Dalam halaman ini ada pilihan menu diantaranya adalah tambah *member*, edit *member*, info *member*, dan hapus *member*.

k. Rancangan Halaman Tambah Member

SiMUH		Hy, Admin
Dashboard	Tambah Member	
Pesan	Username <input type="text"/>	
Paket	Nama <input type="text"/>	
Billing	Password <input type="text"/>	
Member	Alamat <input type="text"/>	
Konfigurasi	Kontak <input type="text"/>	
	<input type="button" value="Simpan"/>	

Gambar 3.24. Rancangan Halaman Tambah Member

Gambar diatas merupakan rancangan halaman tambah *member*. Dalam halaman ini berupa *interface form username*, nama, *password*, halaman, dan kontak.

l. Rancangan Halaman Aktivasi

SiMUH - Sistem Manajemen User Hotspot	
Dashboard	Aktivasi User Hotspot
Pesan	Username <input type="text" value="username"/>
Paket	Paket <input type="text" value="Pilih Paket..."/>
Billing	<input type="button" value="Activate"/> <input type="button" value="Batal"/>
Member	
Konfigurasi	

Gambar 3.25. Rancangan Halaman Aktivasi

Gambar rancangan diatas merupakan rancangan halaman untuk mengaktifkan *user hotspot* agar bisa menggunakan internet.

m. Rancangan Halaman Konfigurasi

SiMUH		Hy, Admin
Dashboard	Konfigurasi	
Pesan		
Paket		
Billing		
Member		
Konfigurasi		
	Nama Router: Router SiMUH Alamat IP: 192.168.1.1 Router User: root Password: **** Simpan Test	

Gambar 3.26. Rancangan Halaman Konfigurasi

Gambar di atas merupakan rancangan halaman konfigurasi *router*. Dimana dalam halaman ini merupakan konfigurasi sambungan antara sistem dan *router* MikroTik yang dipakai. Konfigurasi ini tersambung jika alamat IP, *router user*, dan *router password* sesuai. Dalam halaman ini tersedia tombol *test* untuk menguji sambungan, apakah sistem sukses tersambung dengan *router* atau tidak.

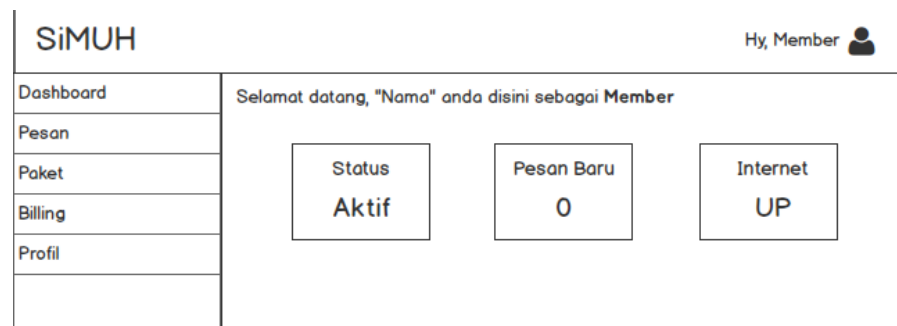
n. Rancangan Halaman Profil

SiMUH		Hy, Admin
Dashboard	Profil Anda	
Pesan		
Paket		
Billing		
Member		
Konfigurasi		
	Username: admin Nama: Sukmo Wijoyo Alamat: Umbulharjo, Yogyakarta Kontak: 085729815915 Password: ***** *) Kosongkan jika tidak mengubah password Simpan Batal	

Gambar 3.27. Rancangan Halaman Profil

Gambar di atas merupakan rancangan halaman untuk menampilkan profil pengguna, dimana pada halaman ini menampilkan *username* yang bersifat statis, nama, alamat, dan kontak. Di sini juga tersedia *form password* jika ingin mengganti *password*.

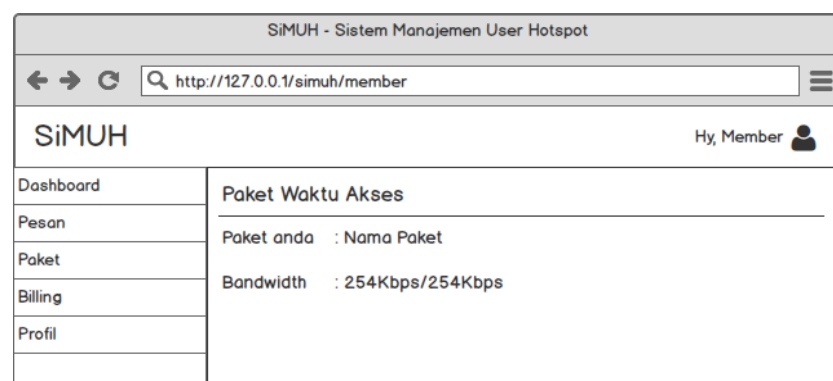
o. Rancangan Halaman *Dashboard Member*



Gambar 3.28. Rancangan Halaman *Dashboard Member*

Gambar di atas merupakan rancangan halaman *dashboard* dari *member* ketika sudah *login*. Dalam halaman ini menampilkan data status *member*, pesan baru dan informasi tersambungny internet atau tidak.

p. Rancangan Halaman *Paket Member*



Gambar 3.29. Rancangan Halaman *Paket Member*

BAB IV

IMPLEMENTASI DAN PEMBAHASAN

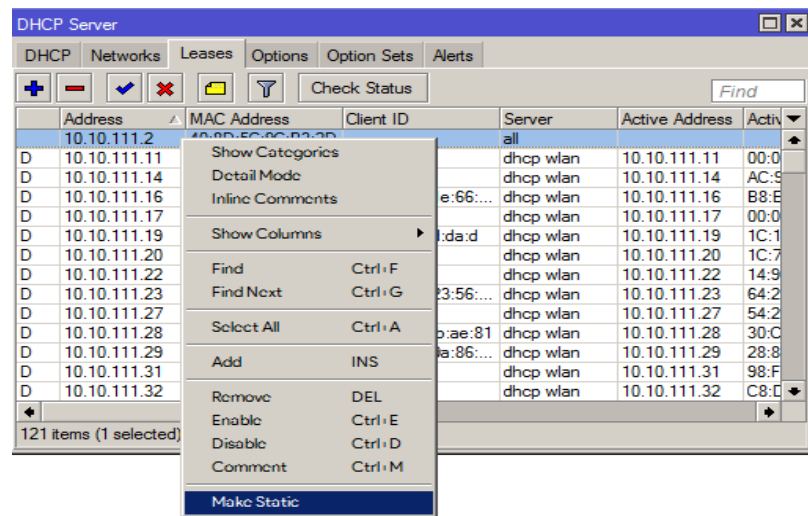
A. Implementasi

Implementasi Sistem Manajemen *User Hotspot* membahas konfigurasi *router* dan tampilan halaman yang sesuai dengan rancangan struktur menu. Aplikasi ini dibagi menjadi dua bagian yaitu panel administrator, panel *member*. Masing-masing halaman terdapat beberapa menu, menu yang ditampilkan untuk admin di antaranya yaitu *Dashboard*, *Pesan*, *Paket*, *Billing*, *Member*, *Konfigurasi*, *Profil* dan *Logout*. Pada halaman member terdapat halaman *Dashboard*, *Pesan*, *Paket*, *Billing*, *Profil* dan *Logout*. Pada halaman administrator, staf bisa mengelola dan manajemen pengguna *hotspot* yang ada di Pondok Pesantren Al-Luqmaniyyah dengan menambah, mengubah dan menghapus data. Sedangkan pada halaman *member* terdapat informasi data yang telah diproses oleh administrator namun pengunjung tidak dapat menambah atau mengubah data kecuali untuk mengubah informasi profil dan mengirim pesan.

1. Konfigurasi Router Mikrotik

Pada proses ini seorang administrator melakukan konfigurasi jaringan yang akan digunakan. Dalam konfigurasi sistem yang dibangun ini menggunakan fitur *captive portal/hotspot gateway* yang disediakan oleh mikrotik. Administrator menggunakan aplikasi winbox untuk mengkonfigurasi mikrotik. Berikut gambaran umum dalam konfigurasi mikrotik:

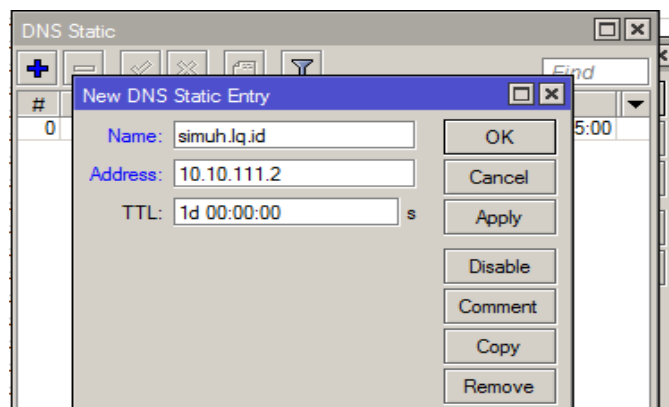
a. Konfigurasi *IP Address* Statis



Gambar 4.1. Konfigurasi *IP Address* Statis

Pada gambar diatas menunjukkan pengaturan agar *IP address* bisa statis pada komputer *client*. Hal ini untuk menghindari perubahan *IP* pada *client* yang dijadikan *server* aplikasi.

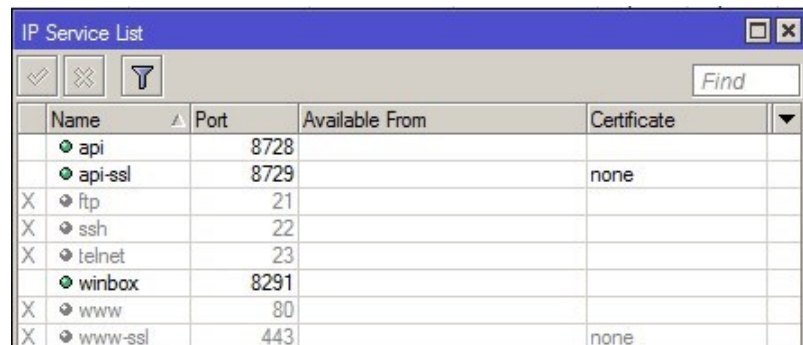
b. Konfigurasi DNS



Gambar 4.2. Pengaturan DNS pada Mikrotik

Pada Gambar 4.2 merupakan konfigurasi untuk mengatur alokasi domain sebagai pengganti *IP address* untuk sistem manajemen *user hotspot*.

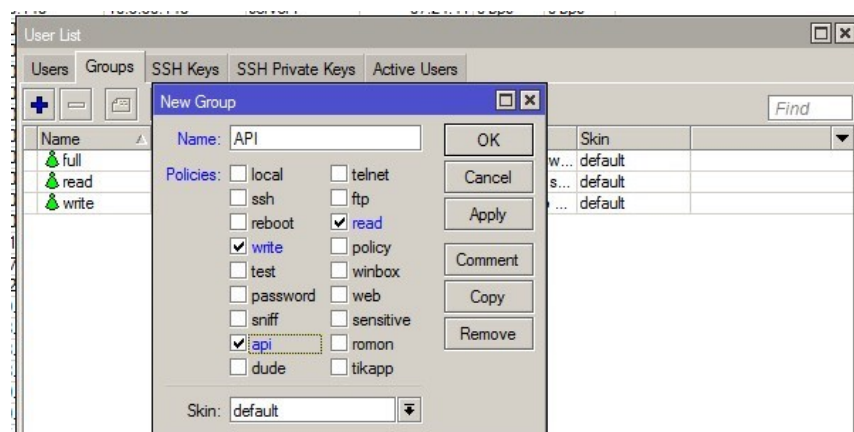
c. Konfigurasi Service API



Gambar 4.3. Konfigurasi Service pada Mikrotik

Pada Gambar 4.3 merupakan konfigurasi Service API pada mikrotik agar sistem yang dibangun dapat tersambung dengan router mikrotik yang digunakan. Service yang diaktifkan hanya beberapa saja, hal ini untuk menghindari dari serangan *brute force* yang dilakukan *attacker*.

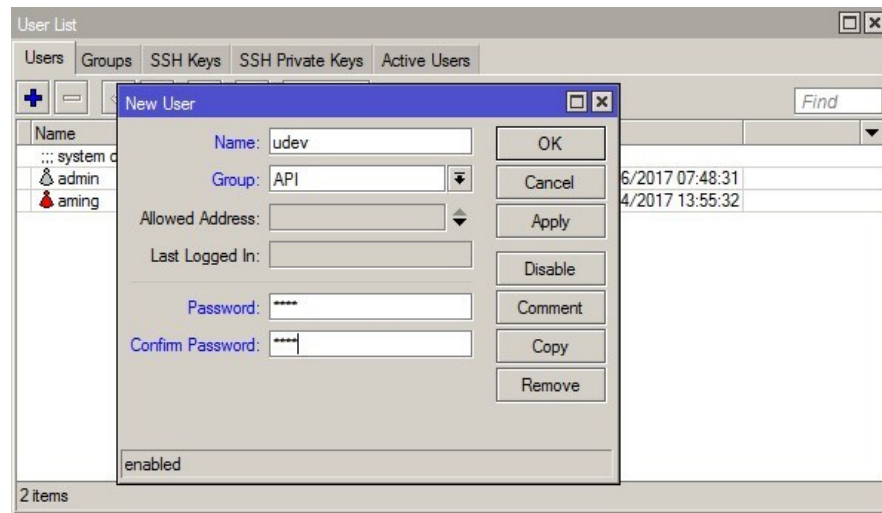
d. Konfigurasi User API



Gambar 4.4. Konfigurasi Grup User API Pada Mikrotik

Pada Gambar 4.4 merupakan konfigurasi grup *user* API pada mikrotik. Dimana konfigurasi hak akses yang diaktifkan hanya *read*,

write dan *api*. Hal ini sebagai langkah mencegah *spoofing* yang ada di jaringan *hotspot*.



Gambar 4.5. Membuat User API Pada Mikrotik

Pada Gambar 4.5 merupakan pembuatan *username*, *password* dan pengelompokan grup yang sudah dibuat sebelumnya. Langkah ini ditempuh agar user ketika login sesuai dengan grup pengaturan di *router* mikrotik,.

2. Tampilan Halaman Admin


a. Halaman *Login*


Halaman *login* yang akan muncul pertama kali saat sistem diakses. Halaman ini berisi *form* isian *username*, *password* dan pilihan jenis akun. Halaman ini merupakan portal utama sebelum mengakses panel *admin* atau *member* sesuai pilihan yang dipilih. Tampilan halaman admin ditunjukkan pada Gambar 4.6.

[SiMUH]

Selamat Datang di
Sistem Manajemen User Hotspot Pondok Pesantren Al-Luqmaniyyah - Yogyakarta


Sign In
Login to access your account.

Username 

Password 

☒ Member ☐ Admin

Masuk

© 2017. All Rights Reserved. Coded with  sukmo

Gambar 4.6. Tampilan Halaman Login

```

require '../core/config.php';
if($_POST['level']=='member'){
    $uname = Filter($_POST['username']);
    $db->go("SELECT `username`, `password` FROM
`tbl_member` WHERE `username` = '$uname'");
    $row = $db->fetchArray();

    $passwd = $_POST['password'];

    if($db->numRows() == 0) {
        Message(2, 'Akun belum terdaftar');
    } else if($row['password'] != $passwd) {
        Message(4, 'Password salah');
    } else {
        $username = $row['username'];
        $level = $_SESSION['level'] = 'member';
        $a = $_SESSION['username'] = $username;
        if($a && $level){
            Redirect(base_url.'/index.php');
            exit();
        } else {
            Message(3, 'Tolong hubungi admin');
        }
    }
}
elseif($_POST['level']=='admin'){
    $uname = Filter($_POST['username']);
    $db->go("SELECT `username`, `password` FROM
`tbl_admin` WHERE `username` = '$uname'");
    $row = $db->fetchArray();

    $passwd = md5($_POST['password']);

    if($db->numRows() == 0) {
        Message(2, 'Akun belum terdaftar');
    } else if($row['password'] != $passwd) {
        Message(4, 'Password salah');
    }
}

```

```

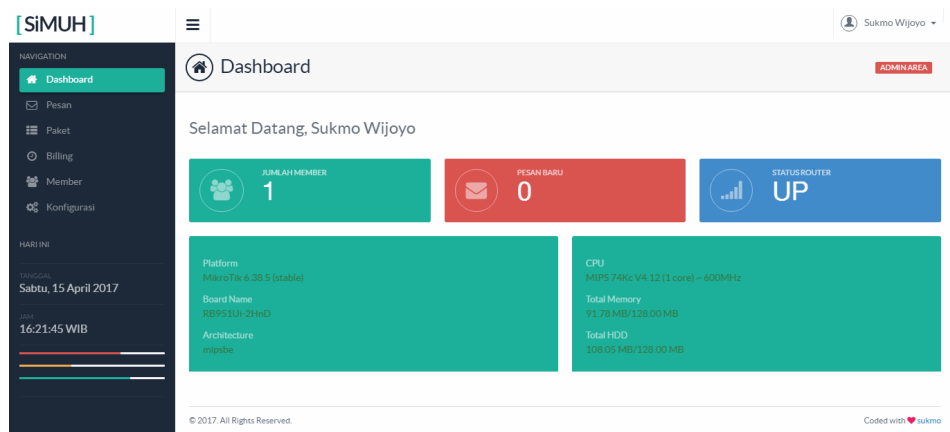
    } else {
        $username = $row['username'];
        $level = $_SESSION['level'] = 'admin';
        $a = $_SESSION['username'] = $username;
        if($a && $level){
            Redirect(base_url.'/index.php');
            exit();
        } else {
            Message(3, 'Tolong hubungi admin');
        }
    }
}
Redirect(base_url.'/login.php');

```

Modul 1. Script Proses Login

Pada Modul 1 merupakan script php untuk proses login pada sistem, kemudian jika username dan password benar maka pengguna akan diarahkan ke halaman yang telah ditentukan sesuai dengan level pengguna. Jika login gagal, maka akan dikembalikan ke halaman login.

b. Halaman *Dashboard* Admin



Gambar 4.7. Halaman *Dashboard* Admin

Pada Gambar 4.7 merupakan halaman utama dari panel *admin*. Pada halaman ini berisi jumlah ember yang terdaftar, jumlah pesan

masuk yang belum dibaca, status koneksi antara sistem dengan *router* dan informasi *hardware router* yang digunakan.

```
//hitung member
$db->go("SELECT * FROM `tbl_member`");
$hmemb=$db->numRows();

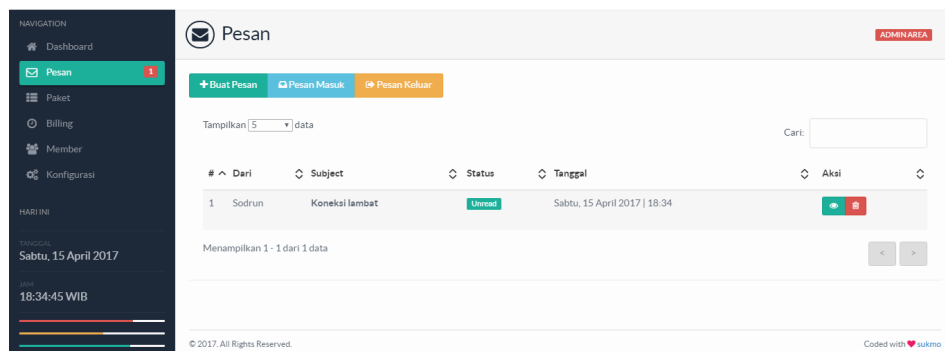
//hitung pesan
$db->go("SELECT * FROM `tbl_pesan` WHERE `status` = '1'
AND `id_receiver` = '$ids'");
$hmail=$db->numRows();

//status router
if($conroute){$info=$conroute-
>getall("/system/resource", FALSE, FALSE, FALSE);}
```

Modul 2. Script Menampilkan Info di Dashboard

Pada Modul 2 merupakan potongan kode untuk menampilkan info jumlah *member*, jumlah pesan, status *router* dan informasi *hardware router*.

c. Halaman Pesan



Gambar 4.8. Halaman Pesan

Pada Gambar 4.8 merupakan daftar pesan masuk admin, dimana pada halaman ini admin bisa membuat, menghapus dan melihat pesan yang dikirim oleh member.

```
$db->go("SELECT tbl_pesan.*, tbl_member.nama,
tbl_admin.username FROM tbl_admin, tbl_pesan JOIN
tbl_member ON tbl_pesan.id_sender = tbl_member.id_member
```

```

WHERE tbl_pesan.id_receiver = '$ids' AND
tbl_admin.username = '$username' ORDER BY
tbl_pesan.time_pesan DESC");

$count = $db->numRows();
no=1;
while ($rowp = $db->fetchArray()){
$sender = ucfirst($rowp['nama']);
$receiver = $rowp['id_receiver'];
$subject = $rowp['subject'];
$kat = $rowp['kat'];
$pesan = $rowp['pesan'];
$time = $rowp['time_pesan'];
$status = $rowp['status'];
if($status=='0'){
$inf="<span class='badge badge-
info'>Read</span>";}else{$inf="<span class='badge badge-
success'>Unread</span>";}

```

Modul 3. Script Query Menampilkan Pesan Admin

Potongan kode pada Modul 3 merupakan *syntax* untuk memanggil data pesan sesuai dengan identitas penerima yang kemudian ditampilkan pada halaman pesan.

d. Halaman Kirim Pesan

Gambar 4.9. Halaman Kirim Pesan

Pada Gambar 4.9 adalah tampilan halaman untuk mengirim pesan dari *admin* ke *member* yang berisi *form* ke, subyek dan pesan.

```

$db->go("SELECT * FROM `tbl_admin` WHERE `username` =
'$username'");
$row = $db->fetchArray();

```

```

$id=$row['id_admin'];
$pilmember=Filter($_POST['pilmember']);
$subyek = Filter($_POST['subyek']);
$pesan = Filter($_POST['pesan']);

if(isset($_POST[' kirim'])) {
if(empty($pilmember) || empty($subyek) ||
empty($pesan)) {
    Message(2, 'Form ada yang kosong');
} else {
    $a = $db->go("INSERT INTO tbl_pesan (id_sender,
id_receiver, subject, pesan) VALUES
('$id', '$pilmember', '$subyek', '$pesan')");
    }
}
if($a) {
    Message(1, 'Pesan telah dikirim');
} else {
    Message(4, 'Pesan gagal dikirim!!');
}
Redirect(base_url.'/admin/add-pesan.php');

```

Modul 4. Script Proses Mengirim Pesan

Potongan kode pada Modul 4 merupakan *syntax* untuk mengirim pesan dari *admin* ke *member* yang telah dipilih.

e. Halaman Lihat Pesan

The screenshot shows a web interface for viewing a message. At the top, there's a header with a mail icon and the title 'Lihat Pesan'. Below the header, there are three tabs: '+ Buat Pesan' (green), 'Pesan Masuk' (blue), and 'Pesan Keluar' (orange). The main content area displays the message details: 'Dari: sodrun', 'Subyek: Koneksi lambat', and 'Pesan: Maaf, koneksi internet lambat'. Below this, there's a 'Balas' (Reply) section with a text input field containing 'Balas disini...'. At the bottom, there are two buttons: 'Balas' (blue) and 'Back' (grey).

Gambar 4.10. Halaman Lihat Pesan

Pada Gambar 4.10 merupakan tampilan halaman untuk melihat pesan masuk yang diterima *admin*. Pada halaman tersebut dilengkapi dengan *form* balas untuk membalas pesan yang dikirim *member*.

```
$idp=Filter($_GET['id']);
$db->go("UPDATE `tbl_pesan` SET status='0' WHERE
id_pesan='$idp'");

$db->go("SELECT tbl_pesan.*, tbl_member.username,
tbl_member.nama FROM tbl_pesan JOIN tbl_member ON
tbl_pesan.id_sender = tbl_member.id_member WHERE
id_pesan = '$idp' AND tbl_pesan.id_receiver='$ids'");
$psn=$db->fetchArray();
```

Modul 5. Script Menampilkan Pesan

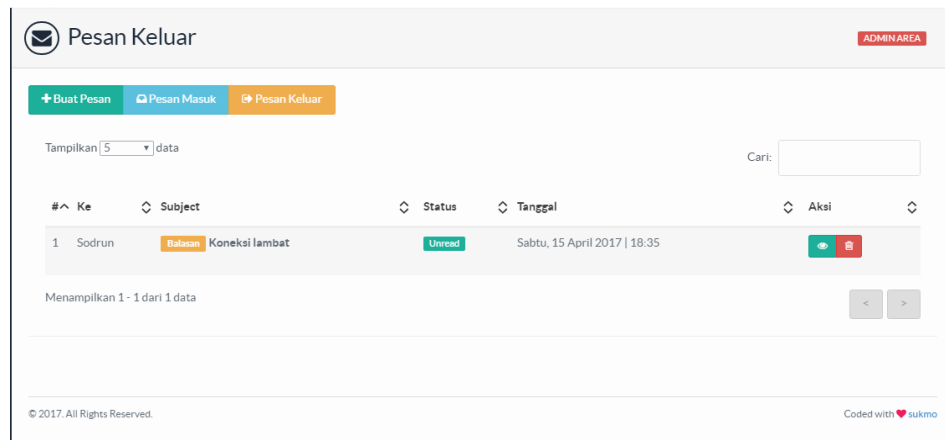
Potongan kode diatas merupakan *syntax* untuk mengambil data untuk menampilkan pesan yang diterima sesuai dengan identitas pesan dan identitas *session* dari pengguna.

```
$idp=$psn['id_pesan'];
$dari = $psn['id_sender'];
$sub = $psn['subject'];
$bls = $_POST['balasan'];
if(isset($_POST['balas']) && !empty($bls)){
$db->go("INSERT INTO tbl_pesan (id_sender, id_receiver,
subject, pesan, kat) VALUES
('$ids', '$dari', '$sub', '$bls', 'B')");
Message(1, 'Balasan dikirim');
Redirect(base_url.'/admin/v-pesan.php?id='.$idp);
}elseif(isset($_POST['balas']) && empty($bls)){
Message(4, 'Balasan gagal/kosong!!!');
Redirect(base_url.'/admin/v-pesan.php?id='.$idp);
}
```

Modul 6. Script Membalas Pesan

Potongan kode diatas adalah *syntax* untuk mengirim balasan kepada pengirim pesan, dengan menyertakan syarat *form* balasan tidak boleh kosong.

f. Halaman Pesan Keluar



Gambar 4.11. Halaman Pesan Keluar

Pada Gambar 4.11 merupakan tampilan dari daftar pesan keluar yang pernah dikirim. Pada informasi tersebut berisi status baca, agar pengirim mengetahui bahwa pesan yang dikirim sudah dibaca atau belum.

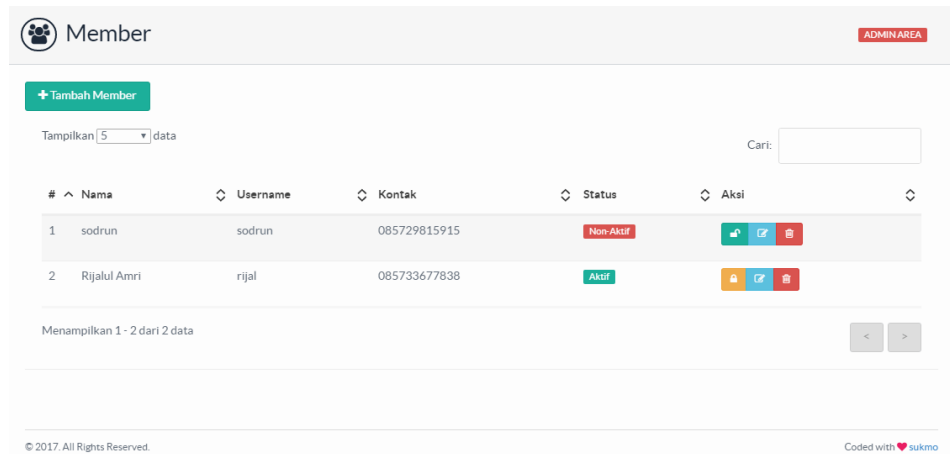
```
$db->go("SELECT tbl_pesan.*, tbl_member.nama,
tbl_admin.username FROM tbl_admin, tbl_pesan JOIN
tbl_member ON tbl_pesan.id_receiver =
tbl_member.id_member WHERE tbl_pesan.id_sender = '$ids'
AND tbl_admin.username = '$username' ORDER BY
tbl_pesan.time_pesan DESC");

$count = $db->numRows();
$no=1;
while ($rowp = $db->fetchArray()) {
$sender = ucfirst($rowp['nama']);
$receiver = $rowp['id_receiver'];
$subject = $rowp['subject'];
$kat = $rowp['kat'];
$pesan = $rowp['pesan'];
$time = $rowp['time_pesan'];
$status = $rowp['status'];
if ($status=='0') {
$inf="<span class='badge badge-info'>Read</span>";
} else { $inf="<span class='badge badge-success'>Unread</span>"; }
}
```

Modul 7. Script Menampilkan Pesan Keluar

Potongan kode diatas merupakan *syntax* untuk mengambil data pesan dari *database*, dimana pesan yang ditampilkan sesuai dengan *session* pengguna.

g. Halaman Daftar *Member*



Gambar 4.12. Halaman Daftar *Member*

Pada Gambar 4.12 merupakan tampilan halaman daftar *member* yang berisi informasi seluruh daftar *member* yang ada di sistem.

```
$db->go("SELECT * FROM `tbl_member` ORDER BY id_member
ASC");
$count = $db->numRows();
$no=1;
while ($row = $db->fetchArray()) {
    $nama = $row['nama'];
    $uname = $row['username'];
    $kontak = $row['telp'];
    $status = $row['status'];
    if($status=='0'){
        $inf="<span class='badge badge-danger'>Non-
Aktif</span>";
    }elseif($status=='1'){ $inf="<span class='badge badge-
success'>Aktif</span>"; }
```

Modul 8. Script Menampilkan Daftar *Member*

Potongan kode diatas digunakan untuk memanggil data member dari database untuk ditampilkan di halaman daftar member.

h. Halaman Tambah *Member*

Gambar 4.13. Halaman Tambah Member

Pada Gambar 4.13 merupakan tampilan halaman tambah member. Pada halaman tersebut disediakan *form username, password, nama, alamat dan kontak*.

```
require '../core/config.php'; require '../inc/auth.php';
$username=Filter($_POST['username']);
$password=Filter($_POST['password']);
$name = Filter($_POST['nama']);
$alamat = Filter($_POST['alamat']);
$kontak = Filter($_POST['kontak']);

if(isset($_POST['save'])){
    if($db->go("SELECT `username` FROM `tbl_member`
    WHERE `username` = '$username'")){
        Message(3,'Username sudah ada');
        Redirect(base_url.'/admin/add-member.php');
    }
    if(empty($username) || empty($password) ||
empty($name) || empty($alamat) || empty($kontak)){
        Message(2, 'Form ada yang kosong');
    }else{
        if($conroute){
            $conroute->add("/ip/hotspot/user",
array("name"=>"$username", "password"=>"$password",
            "disabled"=>"yes"));
            $a = $db->go("INSERT INTO tbl_member
            (username, password, nama, alamat, telp) VALUES
            ('$username','$password','$name','$alamat','$kontak')");
        }else{
            Message(2, 'Koneksi ke-server tidak
            tersambung');
```

```

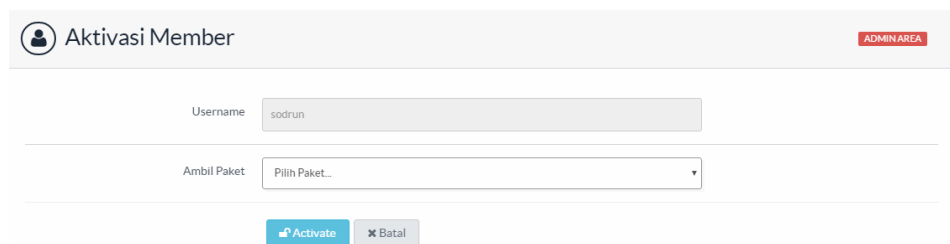
    }
}
if($a){
    Message(1, 'Data telah ditambah');
}else{
    Message(4, 'Data gagal ditambah!!');
}
Redirect(base_url.'/admin/add-member.php');

```

Modul 9. Script Menambah Member

Potongan kode diatas merupakan *syntax* untuk memasukkan data *member* ke dalam *database* untuk disimpan. Dalam *syntax* tersebut *form* yang disediakan tidak boleh ada yang kosong.

i. Halaman Aktivasi Member



Gambar 4.14. Halaman Aktivasi Member

Pada Gambar 4.14 merupakan tampilah halaman untuk mengaktivasi member yang dipilih. Disediakan form drop down untuk memilih paket yang akan digunakan member yang terpilih.

```

$idp=$_POST['piltype'];
$db->go("SELECT * FROM tbl_paket WHERE id_paket = '$idp'");
$pe=$db->fetchArray();
$masa=$pe['masa_aktif'];
$prof=$pe['nama_paket'];
$newmasa=date('Y-m-d H:i:s', strtotime("$masa days"));
$masabarudate('H:i:s', strtotime("$masa days"));
$sekar=date('Y-m-d H:i:s');
$username=$row['username'];
$uptim=$masa.'d '.$masabarudate;

if($masa!=='Unlimited'){

```

```





        if($conroute){
            $conroute->set("/ip/hotspot/user",
            array(".id"=>"$username", "disabled"=>"no", "limit-
            uptime"=>"$uptim", "profile"=>"$prof"));
            $a=$db->go("INSERT INTO tbl_billing (id_user,
            id_paket, daftar, expire, id_admin) VALUES ('$id',
            '$idp', '$sekar', '$newmasa', '$ids')");
            $db->go("UPDATE tbl_member SET status='1' WHERE
            id_member='$id'");
        }else{Message(2, 'Mikrotik tidak menyambung!!');}
    }else{
        if($conroute){
            $conroute->set("/ip/hotspot/user",
            array(".id"=>"$username",
            "disabled"=>"no", "profile"=>"$prof"));
            $a=$db->go("INSERT INTO tbl_billing (id_user,
            id_paket, daftar, id_admin) VALUES ('$id', '$idp',
            '$sekar', '$ids')");
            $db->go("UPDATE tbl_member SET status='1' WHERE
            id_member='$id'");
        }else{Message(2, 'Mikrotik tidak menyambung!!');}
    }
    if($a){
        Message(1, 'Data telah diaktivasi');
    }else{
        Message(4, 'Data gagal diaktivasi!!');
    }
    Redirect(base_url.'/admin/member.php');

```

Modul 10. Script Aktivasi member

Potongan kode diatas adalah proses aktivasi *member* yang telah dipilih sesuai paket yang dipilih dan *syntax* tersebut terintegrasi dengan *router* mikrotik.

j. Halaman Daftar Paket

#	Nama Paket	Type Paket	Bandwidth	Aksi
1	3 hari	Limited	123k/123k	 
2	Ustadz	Unlimited	123k/123k	 

Gambar 4.15. Halaman Daftar Paket

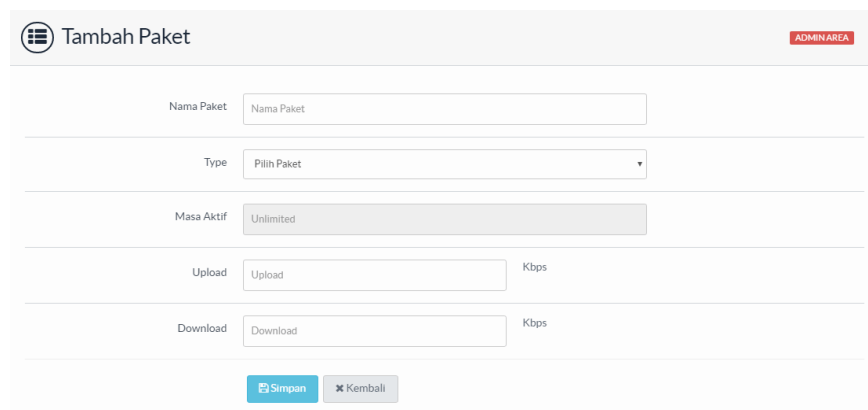
Pada Gambar 4.15 merupakan tampilan dari halaman daftar paket. Halaman ini menampilkan nama, *type* paket dan *bandwidth*. Pada halaman tersebut juga terdapat tombol untuk menambah, mengedit dan menghapus paket.

```
$db->go("SELECT * FROM tbl_paket");
$count = $db->numRows();
$no=1;
while ($rowp = $db->fetchArray()){
    $idpaket = $rowp['id_paket'];
    $namapaket = $rowp['nama_paket'];
    $type = $rowp['jenis_paket'];
    $bandwidth = $rowp['bandwidth'];
```

Modul 11. Script Menampilkan Daftar Paket

Potongan kode diatas merupakan *syntax* untuk mengambil data dari *database* untuk ditampilkan pada halaman daftar paket.

k. Halaman Tambah Paket



The screenshot shows a web form titled 'Tambah Paket' with an 'ADMIN AREA' label in the top right. The form contains the following elements:

- Nama Paket:** A text input field.
- Type:** A dropdown menu with 'Pilih Paket' as the selected option.
- Masa Aktif:** A text input field with 'Unlimited' entered.
- Upload:** A text input field with 'Upload' entered, followed by a 'Kbps' label.
- Download:** A text input field with 'Download' entered, followed by a 'Kbps' label.
- Buttons:** 'Simpan' (Save) and 'Kembali' (Back) buttons at the bottom.

Gambar 4.16. Halaman Tambah Paket

Pada Gambar 4.16 merupakan tampilan halaman untuk menambah paket yang berisi *form* nama paket, tipe paket, masa aktif dan *bandwidth*. Pada *form* tipe paket terdapat dua pilihan yaitu: *unlimited* dan *based time*. Secara *default* penambahan paket dengan

type *unlimited* dan *form* masa aktif tidak aktif. Jika pengguna memilih type paket *based time* maka *form* masa aktif akan aktif.

```

require '../core/config.php'; require '../inc/auth.php';
$namapaket=Filter($_POST['namapaket']);
$type=Filter($_POST['piltype']);
$masaaktif = Filter($_POST['masaaktif']);
$bw['0'] = Filter($_POST['upload'].'k');
$bw['1'] = Filter($_POST['download'].'k');
$band=implode("/", $bw);

if(isset($_POST['simpan'])) {
    $db->go("SELECT `nama_paket` FROM `tbl_paket`
WHERE `nama_paket` = '$namapaket'");
    $cep=$db->fetchArray();
    if($cep['nama_paket']!=NULL) {
        //Message(3, 'Maaf, Nama paket sudah
ada dan tidak boleh sama. ');
        Redirect(base_url.'/admin/add-
paket.php');
    } elseif($cep['nama_paket']==NULL) {
        if($type=='0') {
            if(empty($namapaket) ||
empty($bw['0']) || empty($bw['1'])) {
                Message(2, 'Form ada yang
kosong');
            } else {
                if($conroute) {
                    $conroute-
>add("/ip/hotspot/user/profile",
array("name"=>"$namapaket", "rate-limit"=>"$band"));
                    $a = $db->go("INSERT INTO
tbl_paket (nama_paket, jenis_paket, bandwidth,
masa_aktif) VALUES ('$namapaket','$type','$band',
'Unlimited')");
                } else {
                    Message(2, 'Koneksi ke-
server tidak tersambung');
                }
            }
        } elseif($type=='1') {
            if(empty($namapaket) ||
empty($masaaktif) || empty($bw['0']) ||
empty($bw['1'])) {
                Message(2, 'Form ada yang
kosong');
            } else {
                if($conroute) {
                    $conroute-
>add("/ip/hotspot/user/profile",
array("name"=>"$namapaket", "rate-limit"=>"$band"));
                    $a = $db->go("INSERT INTO
tbl_paket (nama_paket, jenis_paket, bandwidth,

```

```

masa_aktif) VALUES
('$namapaket','$type','$band','$masaaktif');
    }else{
        Message(2, 'Koneksi ke-
server tidak tersambung');
    }
    }
}

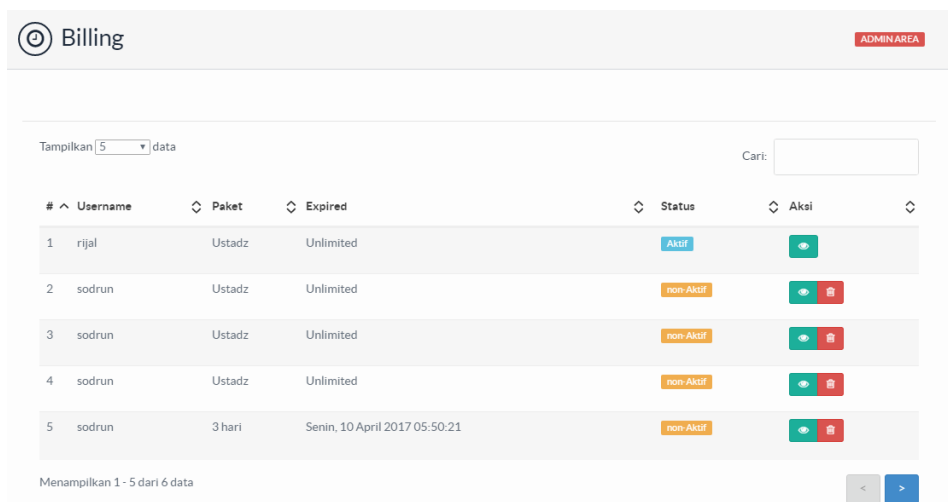
if($a){
    Message(1, 'Data telah ditambah');
}else{
    Message(4, 'Data gagal ditambah!!');
}
Redirect(base_url.'/admin/add-paket.php');

```










Modul 12. Script Proses Tambah Paket

Potongan kode diatas adalah proses memasukkan data paket yang telah diisi. Dalam *syntax* tersebut semua *form* harus diisi sesuai dengan tipe data yang telah diatur. Jika tidak sesuai maka data tidak akan bisa masuk dalam *database* dan *router*.

1. Halaman Daftar *Billing*



The screenshot shows a web interface for a billing system. At the top, there is a header with a 'Billing' logo and an 'ADMIN AREA' button. Below the header, there is a search bar and a table listing members. The table has columns for '#', 'Username', 'Paket', 'Expired', 'Status', and 'Aksi'. The first member is 'rijal' with 'Ustadz' package and 'Unlimited' expiration, status 'Aktif'. The next three members are 'sodrun' with 'Ustadz' package and 'Unlimited' expiration, status 'non-Aktif'. The last member is 'sodrun' with '3 hari' package and 'Senin, 10 April 2017 05:50:21' expiration, status 'non-Aktif'. At the bottom, there is a pagination bar showing 'Menampilkan 1 - 5 dari 6 data' and navigation buttons.

#	Username	Paket	Expired	Status	Aksi
1	rijal	Ustadz	Unlimited	Aktif	
2	sodrun	Ustadz	Unlimited	non-Aktif	 
3	sodrun	Ustadz	Unlimited	non-Aktif	 
4	sodrun	Ustadz	Unlimited	non-Aktif	 
5	sodrun	3 hari	Senin, 10 April 2017 05:50:21	non-Aktif	 

Gambar 4.17. Halaman Daftar Member

Pada Gambar 4.17 merupakan tampilan dari halaman daftar *billing*. Pada halaman ini terdapat informasi *username member*, paket yang digunakan, masa berakhir paket dan status. Dalam halaman ini juga terdapat tombol lihat detail *billing* dan hapus *billing*.

```
$db->go("SELECT tbl_billing.*, tbl_member.username,
tbl_paket.nama_paket FROM tbl_paket, tbl_billing JOIN
tbl_member ON tbl_billing.id_user = tbl_member.id_member
WHERE tbl_billing.id_paket=tbl_paket.`id_paket` ORDER BY
tbl_billing.daftar DESC");

$count = $db->numRows();
$no=1;
while ($rowp = $db->fetchArray()){
    $member = $rowp['username'];
    $paket = $rowp['nama_paket'];
    $expire = $rowp['expire'];
    $status = $rowp['status'];
    if($status=='1'){$inf="<span class='badge badge-
info'>Aktif</span>";
    }else{$inf="<span class='badge badge-warning'>non-
Aktif</span>";}
```

Modul 13. Script Menampilkan Daftar Billing

Potongan kode diatas merupakan *syntax* untuk mengambil data *billing*. Adapun untuk data yang disajikan sesuai dengan yang ada dalam rancangan, maka pada *query mysql* menggunakan *table join*.

m. Halaman Konfigurasi

Gambar 4.18. Halaman Konfigurasi

Pada Gambar 4.18 merupakan tampilan halaman untuk mengonfigurasi sistem agar komunikasi antara sistem yang dipakai bisa terintegrasi dengan *router* yang dipakai. Halaman ini menampilkan *form* nama *router*, *IP router*, *username router* dan *password router*. Terdapat juga tombol *test* untuk mengetes koneksi antara sistem dengan *router* tersambung atau tidak.

```
require '../core/config.php'; require '../inc/auth.php';
if(isset($_POST['save'])){
    $nama = Filter($_POST['rname']);
    $rip = Filter($_POST['rip']);
    $ruser = Filter($_POST['ruser']);
    $rpass = Filter($_POST['rpass']);

    $a = $db->go("UPDATE `tbl_config` SET
`router_name` = '$nama', `router_ip`= '$rip',
`router_user` = '$ruser', `router_pass` = '$rpass'");
    if($a){
        Message(1, 'Konfigurasi telah disimpan');
    }else{
        Message(3, 'Konfigurasi gagal disimpan!!');
    }
}

if(isset($_POST['test'])){
    $rip = Filter($_POST['rip']);
    $ruser = Filter($_POST['ruser']);
    $rpass = Filter($_POST['rpass']);

    $conroute = RouterOS::connect($rip, $ruser,
$rpass);
    if($conroute){
        Message(1, 'Sambungan sukses');
    }else{
        Message(3, 'Sambungan gagal!!');
    }
}

Redirect(base_url.'/admin/config.php');
```

Modul 14. Script Proses Konfigurasi

Potongan kode diatas adalah proses konfigurasi antara sistem dengan router. Terdapat juga uji coba sambungan dengan router untuk memberikan informasi bahwa pengaturan yang dimasukkan benar.

n. Halaman Profil

Gambar 4.19. Halaman Profil

Pada Gambar 4.19 adalah tampilan halaman profil berisi informasi diri pengguna, lengkap dengan edit data jika ada kekeliruan dalam mengisi identitas.

```
require '../core/config.php'; require '../inc/auth.php';
if($_SESSION['level']=='admin') {

include '../header.php';
$db->go("SELECT * FROM `tbl_admin` WHERE `username` =
'$username'");
$row = $db->fetchArray();
```

Modul 15. Script Menampilkan Data Pengguna

Potongan data diatas untuk menampilkan data pada halaman profil yang sesuai dengan *session username* yang dipakai.

```
if($_SESSION['level']=='admin')
{
    if(isset($_POST['save'])) {
```

```

        $password = Filter(md5($_POST['password']));
        if(empty($_POST['password'])) {
            $a = $db->go("UPDATE `tbl_admin` SET `nama`
= '$nama', `alamat` = '$alamat', `telp` = '$kontak'
WHERE `username` = '$username'");
        } else {
            $a = $db->go("UPDATE `tbl_admin` SET
`password` = '$password', `nama` = '$nama', `alamat` =
'$alamat', `telp` = '$kontak' WHERE `username` =
'$username'");
        }
    }
    if($a) {
        Message(1, 'Data telah diperbarui');
    } else {
        Message(4, 'Data gagal diperbarui!!');
    }
    Redirect(base_url.'/admin/profile.php');
}
elseif($_SESSION['level']=='member')
{
    if(isset($_POST['save'])) {
        $password = Filter($_POST['password']);
        if(empty($_POST['password'])) {
            $a = $db->go("UPDATE `tbl_member` SET `nama`
= '$nama', `alamat` = '$alamat', `telp` = '$kontak'
WHERE `username` = '$username'");
        } else {
            if($conroute) {
                $conroute-
>set("/ip/hotspot/user", array(".id"=>"$username",
"password"=>"$password"));
                $a = $db->go("UPDATE
`tbl_member` SET `password` = '$password', `nama` =
'$nama', `alamat` = '$alamat', `telp` = '$kontak' WHERE
`username` = '$username'");
            }
        }
    }
    if($a) {
        Message(1, 'Data telah diperbarui');
    } else {
        Message(4, 'Data gagal diperbarui!!');
    }
    Redirect(base_url.'/member/profile.php');
}

```

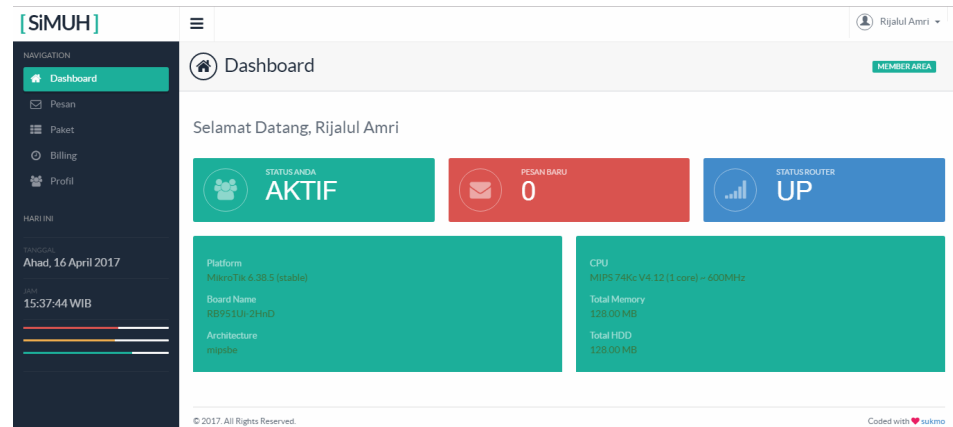
Modul 16. Script Edit Profil

Potongan kode diatas merupakan syntax untuk mengedit data pada session level dan session usir yang digunakan login. Hal ini

untuk menghindari kesalahan dalam mengedit data dan otoritas dalam mengedit data.

3. Tampilan Halaman *Member*

a. Halaman *Dashboard Member*

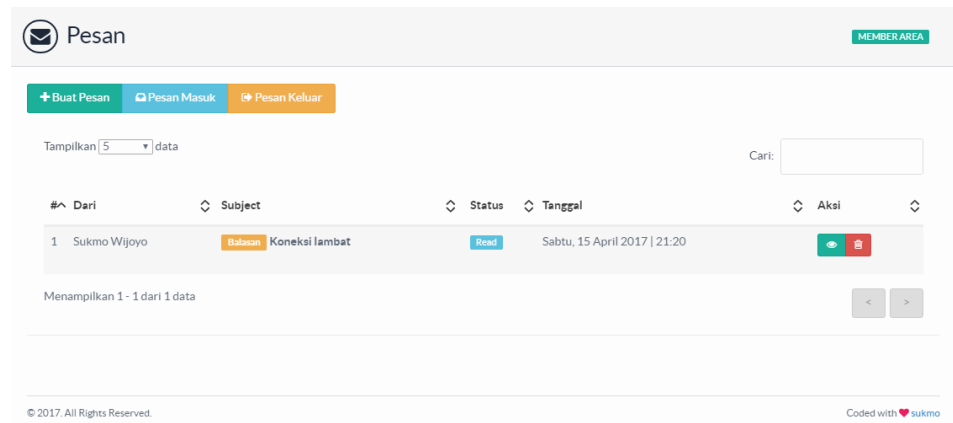


Gambar 4. 20. Halaman *Dashboard Member*

Pada Gambar 4. 20 merupakan tampilan *dashboard member* ketika berhasil *login*. Halaman ini berisi informasi status pengguna, jumlah pesan yang belum dibaca, status *router* dan informasi *hardware router*. Secara struktur *syntax* yang digunakan sama dengan *dashboard admin*, letak perbedaannya hanya pada level pengguna yang digunakan.

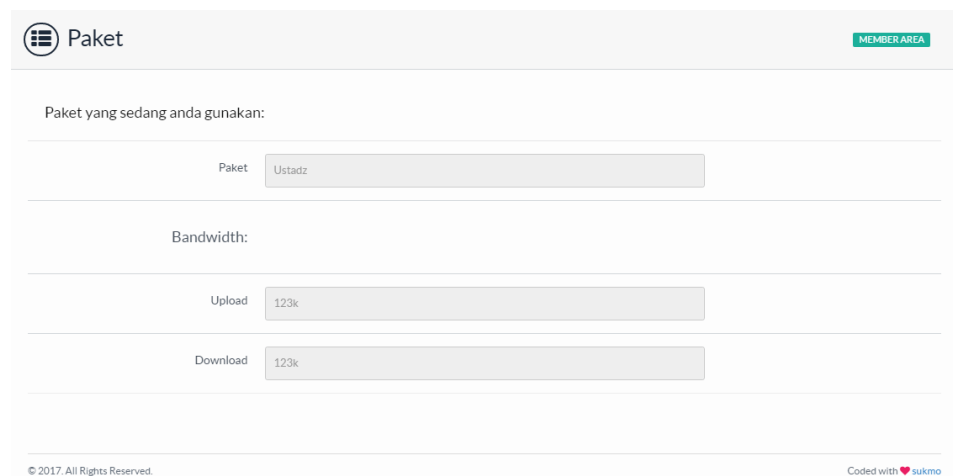
b. Halaman *Pesan Member*

Seperti pada halaman pesan *admin*, pada halaman pesan *member* berisi pesan yang dikirim dari *admin* kepada pengguna yang sesuai dengan *id session* yang digunakan. Seperti pada gambar dibawah ini:



Gambar 4. 21. Halaman Pesan Member

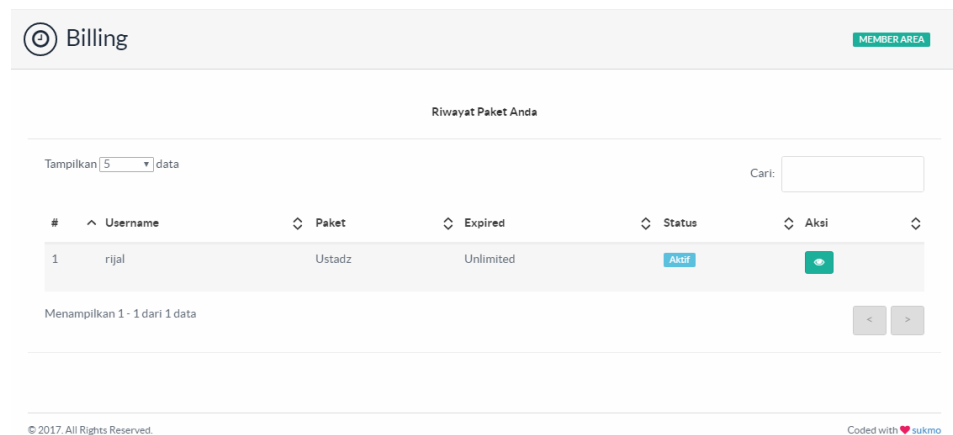
c. Halaman Paket *Member*



Gambar 4.22. Halaman Paket Member

Pada Gambar 4.22 merupakan tampilan dari paket yang digunakan oleh *member* yang berisi info nama paket dan *bandwidth* yang diatur oleh *admin*.

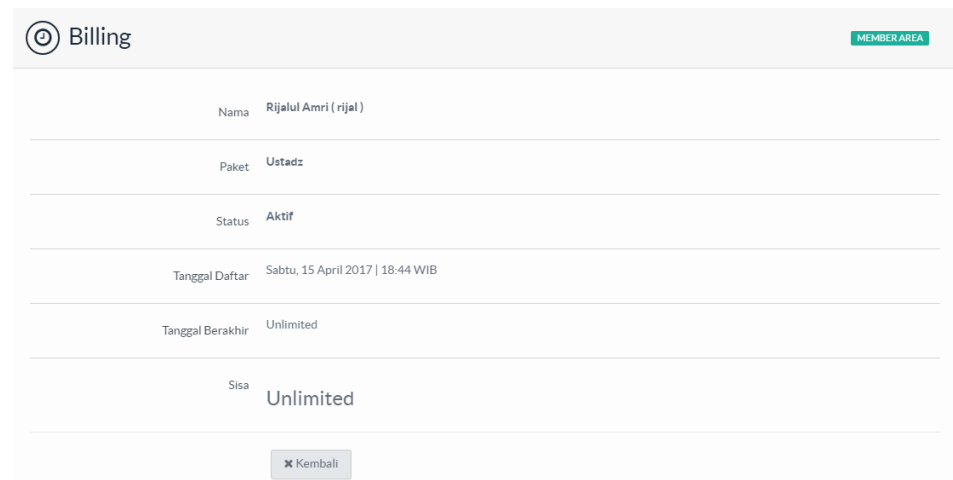
d. Halaman *Billing Member*



Gambar 4. 23. Halaman *Billing Member*

Pada Gambar 4. 23 merupakan tampilan halaman member yang berisi informasi singkat riwayat paket yang pernah digunakan.

e. Halaman *Lihat Billing*

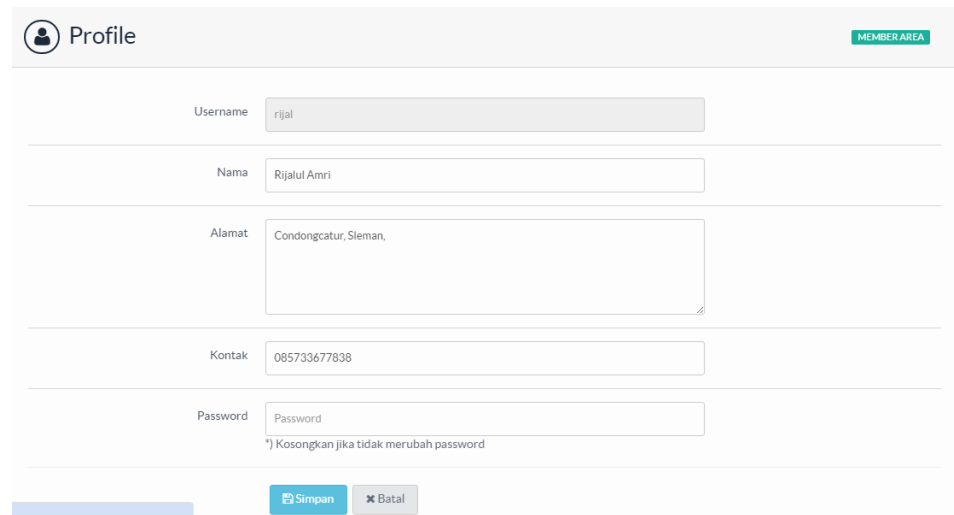


Gambar 4. 24. Halaman *Lihat Billing*

Pada Gambar 4. 24 merupakan tampilan halaman *billing* yang lebih rinci. Dalam halaman tersebut berisi informasi nama, *username*, paket yang digunakan, tanggal aktivasi paket, tanggal berakhir paket

dan sisa paket jika menggunakan paket *time based*. Jika paket yang digunakan *unlimited* maka yang ditampilkan tulisan *unlimited*.

f. Halaman Profil *Member*



The screenshot shows a web interface for a member's profile. At the top, there is a header bar with a user icon and the word "Profile" on the left, and a green "MEMBER AREA" button on the right. Below the header, the profile information is displayed in a form with the following fields:

- Username:** A text input field containing the value "rijal".
- Nama:** A text input field containing the value "Rijalul Amri".
- Alamat:** A larger text input field containing the value "Condongcatur, Sleman,".
- Kontak:** A text input field containing the value "085733677838".
- Password:** A text input field containing the value "Password". Below this field, there is a small note: "*) Kosongkan jika tidak merubah password".

At the bottom of the form, there are two buttons: a blue "Simpan" (Save) button and a grey "Batal" (Cancel) button.

Gambar 4. 25. Halaman Profil *Member*

Pada Gambar 4. 25 merupakan tampilan halaman *member* yang berisi informasi identitas diri dari pengguna. Pada halaman ini juga pengguna bisa mengedit data maupun *password* jika ada kekeliruan.

B. Pembahasan

Pembahasan berisi tentang hasil pengujian, kelebihan dan kekurangan aplikasi sistem manajemen *user hotspot* yang ada di Pondok Pesantren Al-Luqmaniyyah.

1. Hasil Pengujian

Setelah membahas implementasi menu maupun tampilan kemudian dilanjutkan dengan pengujian program yang telah dibuat dengan menggunakan dua metode yaitu pengetesan kotak hitam (*Black Box Test*

Method), dan metode pengetesan Alfa (*Alfa Test Method*). Hasil dari masing-masing pengetesan tersebut adalah sebagai berikut:

a. Pengetesan Kotak Hitam (*Black Box Test*)

Pengujian dilakukan oleh dosen pembimbing dan mahasiswa. pengujian dilakukan dengan cara menjalankan program. Pengujian dilakukan untuk mengetahui apakah aplikasi manajemen *user hotspot* ini dapat berjalan dengan baik.

b. Pengetesan Alfa (*Alpha Test*)

Pelaksanaan pengujian *alpha* dilakukan oleh 30 (tiga puluh) orang responden. Responden menjalankan program dan selanjutnya mengisi daftar pertanyaan (kuisisioner) sebagai respons terhadap kinerja sistem yang dibangun.

Respons yang dihasilkan dari pengguna terhadap kinerja sistem yang dibangun adalah sebagai berikut:

Tabel 4. 1. Pengujian Usabilitas Aplikasi

No	Pertanyaan	Penilaian				
		SS	S	N	TS	STS
1.	Saya memahami cara menggunakan aplikasi SiMUH	10	20	0	0	0
2.	Saya merasa mudah memanajemen <i>user hotspot</i> setelah menggunakan aplikasi SiMUH	8	22	0	0	0
3.	Saya setuju SiMUH ini memiliki tampilan yang menarik	18	12	0	0	0
Jumlah		36	54	0	0	0

Keterangan:**SS:** Sangat setuju**TS:** Tidak setuju**S:** Setuju**STS:** Sangat tidak setuju**N:** Netral*Tabel 4. 2. Pengujian Fungsionalitas Aplikasi*

No	Pertanyaan	Penilaian	
		Ya	Tidak
1.	Sistem dapat menampilkan halaman muka	30	0
2.	Sistem dapat tersambung dengan mikrotik	30	0
3.	Sistem dapat menambah, mengedit dan menghapus paket	30	0
4.	Sistem dapat menambah, mengedit dan menghapus <i>member</i>	30	0
5.	Sistem dapat menampilkan <i>billing</i>	30	0
6.	Sistem dapat mengirim pesan dari <i>admin</i> ke <i>member</i> maupun sebaliknya.	30	0
7.	Hak akses halaman sesuai dengan level yang ditentukan	30	0
Jumlah		210	0

c. Hasil Pengujian Alpha Testing

Presentase hasil pengujian *usability* sistem :

- Sangat setuju : $(36/90) \times 100 = 40\%$
- Setuju: $(54/90) \times 100 = 60\%$
- Netral: $(0/90) \times 100 = 0\%$
- Tidak setuju: $(0/90) \times 100 = 0\%$
- Sangat tidak setuju: $(0/90) \times 100 = 0\%$

Presentase hasil pengujian fungsionalitas sistem :

- Menjawab YA : $(210/210) \times 100 = 100\%$
- Menjawab TIDAK : $(0/210) \times 100 = 0\%$

Berdasarkan hasil pengujian yang melibatkan 30 (tiga puluh) responden atau pengguna di atas, dapat diketahui bahwa sebagian besar pengguna menyatakan penilaian yang baik terhadap aplikasi yang telah dibuat, maka didapat hasil pengujian yang menunjukkan bahwa 100% pengguna menyatakan fungsionalitas sistem telah berjalan dengan baik dan 0% responden menyatakan fungsional sistem tidak berjalan dengan baik.

Berdasarkan pengujian dari segi *usability* diperoleh kesimpulan bahwa sebagian besar responden cukup puas dengan aplikasi yang dibuat. Data hasil pengujian *usability* bahwa responden sangat setuju sebanyak 40%, setuju sebanyak 60%, netral 0%, yang menyatakan tidak setuju sebanyak 0% dan yang menyatakan sangat tidak setuju sebanyak 0%.

Berdasarkan hasil pengujian tersebut, dapat disimpulkan bahwa aplikasi yang telah dibuat ini layak untuk digunakan. Akan tetapi perlu adanya pengembangan aplikasi yang lebih lanjut untuk mendapatkan aplikasi yang optimal.

2. Kelebihan

Berdasarkan perencanaan dan pembuatan rancang bangun sistem manajemen *user hotspot* ini mempunyai beberapa kelebihan yang dapat disampaikan, diantaranya:

- a. Aplikasi ini dapat berjalan pada berbagai sistem operasi, karena aplikasi ini dijalankan melalui *browser*.
- b. Aplikasi ini menggunakan bahasa pemrograman PHP dan MySQL, sehingga sangat mudah dalam mengembangkan sistem ini.
- c. Aplikasi ini dapat berjalan baik pada *smartphone*, karena desain yang buat pada aplikasi menggunakan teknologi *web* yang *responsive*.
- d. Penggunaan aplikasi yang lebih simpel, sehingga lebih mudah dipahami oleh pengguna.
- e. Kelengkapan *form* data pada *member* memudahkan pemetaan dalam manajemen pengguna.

3. Kekurangan

Berdasarkan hasil yang telah dibuat dapat disampaikan beberapa kelemahan, diantaranya:

- a. Fitur-fitur penting yang ada pada mikrotik belum seluruhnya masuk dalam modul sistem.
- b. Perlu konfigurasi terlebih dahulu pada *router* mikrotik sebelum aplikasi ini digunakan.
- c. Memerlukan *server* eksternal untuk menjalankan aplikasi ini.

- d. Paket pada sistem yang dirancang hanya mendukung kategori paket *unlimited* dan *time based* saja, belum mendukung *volume based*.

BAB V

PENUTUP

A. Kesimpulan

Setelah melakukan pengamatan, mengumpulkan data dan melakukan analisa terhadap rancang bangun sistem manajemen *user hotspot* di Pondok Pesantren Al-Luqmaniyyah ini dapat ditarik kesimpulan kesimpulan sebagai berikut:

1. Sistem mampu berfungsi dengan baik sebagai aplikasi manajemen *user hotspot* di Pondok Pesantren Al-Luqmaniyyah.
2. Sistem bisa terintegrasi dengan *router* mikrotik yang ada di Pondok Pesantren Al-Luqmaniyyah.
3. Berdasarkan hasil pengujian *alpha test* yang melibatkan 30 responden memperoleh hasil 100% fungsi aplikasi berjalan sesuai dengan rancangan.

B. Saran

Mengingat berbagai keterbatasan yang dialami penulis terutama masalah pemikiran dan waktu, maka dari itu penyusun memberikan saran yang dapat dipertimbangkan, sebagai berikut :

1. Aplikasi yang dibuat belum sempurna, untuk itu diharapkan nantinya dapat diberikan fitur-fitur baru yang lebih lengkap dan lebih baik, karena pada aplikasi ini baru mampu memanajemen *user hotspot* saja, belum mencakup manajemen *bandwidth* yang lebih terstruktur, pengaturan *firewall*, dan lain-lain.

2. Pada sistem ini masih menggunakan fitur *unlimited* dan *time based*.

Penulis berharap ada tambahan fitur yang dibatasi kuota pada mikrotik.

3. Pengembangan dalam hal keamanan *database* dan *web* jaringan lokal.

DAFTAR PUSTAKA

- Buchory, Purwaningsih, O., & dkk. (2016). *Pedoman Penulisan Skripsi*. Yogyakarta: Universitas PGRI Yogyakarta.
- Irvantino, H., & Ino. (2011). *Konfigurasi Wireless Routerboard Mikrotik*. Yogyakarta: CV. Andi Offset.
- Kuswaryadi, D. (2011). *New UserManager v.5*.
- Mardiana, G. (2015). *Sistem Pemesanan Menu Berbasis Web Memanfaatkan Mikrotik API (Studi Kasus: Miaw Shake Cat Cafe)*. Yogyakarta: Universitas Islam Negeri Sunan Kalijaga Yogyakarta.
- Okto, Y. (2012). *Pemanfaatan API pada Perangkat Hotspot Gateway dalam Pembuatan Sistem Akses Internet Berdasarkan Volume Based dan Time Based Access Berbasis Web*. Salatiga: Universitas Kristen Satya Wacana.
- Saputro, W. A. (2013, Juli 20). *Mengenal Router*. Dipetik Mei 30, 2016, dari Wahyu Computer IT: <http://wahyucomputerit.blog.amikom.ac.id/2013/07/20/mengenal-router/>
- Setiawan, M. Y. (2015). *Perancangan Dan Implementasi Billing Hotspot Dengan Menggunakan PHP dan API pada Mikrotik di Cybercity Networks*. Bandung: Universitas Komputer Indonesia.
- The PHP Group. (2016, Juni 2). *PHP 5 ChangeLog*. Diambil kembali dari PHP: <http://php.net/ChangeLog-5.php#5.6.22>
- Towidjojo, R. (2013). *Mikrotik Kung Fu: Kitab 1*. Jakarta: Jasakom.
- _____. (2013). *Mikrotik Kung Fu: Kitab 2*. Jakarta: Jasakom.
- Trzcinski, K. (2017, Februari 5). *RouterOS PHP class*. Diambil kembali dari Mikrotik Documentation: https://wiki.mikrotik.com/wiki/RouterOS_PHP_class
- Valade, J. (2004). *PHP 5 For Dummies*. Indianapolis, Indiana: Wiley Publishing, Inc.
- Yakub. (2008). *Sistem Basis Data Tutorial Konseptual*. Yogyakarta: Graha Ilmu.

LAMPIRAN

Lampiran 1: Script Program

Keseluruhan kode sumber dapat diakses secara publik dengan alamat tautan

<http://gitlab.com/lorosukmo/skripsi>

Script 1. Function.inc.php

```
<?php

function indoDate($timestamp = '', $date_format = 'l, j F Y |
H:i') {
    if (trim ($timestamp) == '')
    {
        $timestamp = time ();
    }
    elseif (!ctype_digit ($timestamp))
    {
        $timestamp = strtotime ($timestamp);
    }
    # remove S (st,nd,rd,th) there are no such things in indonesia
:p
    $date_format = preg_replace ("/S/", "", $date_format);
    $pattern = array (

'/Mon[^day]/','/Tue[^sday]/','/Wed[^nesday]/','/Thu[^rsday]/',
'/Fri[^day]/','/Sat[^urday]/','/Sun[^day]/','/Monday/', '/Tuesday/'
,
'/Wednesday/', '/Thursday/', '/Friday/', '/Saturday/', '/Sunday/',
'/Jan[^uary]/','/Feb[^ruary]/','/Mar[^ch]/','/Apr[^il]/','/May/',
'/Jun[^e]/','/Jul[^y]/','/Aug[^ust]/','/Sep[^tember]/','/Oct[^ober]
/'/,
'/Nov[^ember]/','/Dec[^ember]/','/January/', '/February/', '/March/'
,
'/April/', '/June/', '/July/', '/August/', '/September/', '/October/',
'/November/', '/December/',
);
    $replace = array ( 'Sen', 'Sel', 'Rab', 'Kam', 'Jum', 'Sab', 'Aha',
        'Senin', 'Selasa', 'Rabu', 'Kamis', 'Jumat', 'Sabtu', 'Ahad',
        'Jan', 'Feb', 'Mar', 'Apr', 'Mei', 'Jun', 'Jul', 'Ags', 'Sep', 'Okt', 'Nov',
        'Des',
        'Januari', 'Februari', 'Maret', 'April', 'Juni', 'Juli', 'Agustus', 'Sepe
mber',
```

```

        'Oktober','November','Desember',
    );
    $date = date ($date_format, $timestamp);
    $date = preg_replace ($pattern, $replace, $date);
    $date = "{$date}";
    return $date;
}

function formatBytes($bytes, $dec = 2){
    $size = array(' B', ' kB', ' MB', ' GB', ' TB', ' PB', '
EB', ' ZB', ' YB');
    $factor = floor((strlen($bytes) - 1) / 3);

    return sprintf("%.{$dec}f", $bytes / pow(1024, $factor)) .
@$size[$factor];
}

function Status($data){
    if($data=='1'){
        $data="Aktif";
    }else{
        $data="Non-Aktif";
    }
}

function classAktif($requestUri){
    $current_file_name = basename($_SERVER['PHP_SELF']);

    if ($current_file_name == $requestUri)
        echo 'class="active"';
}

function Filter($data){
    $block = array('","',';');
    return Sqli(trim(str_replace($block, '', $data)));
}

function Sqli($data){
    global $setting;
    $block = array('concat', 'union', 'base64_decode',
'group_concat', 'tables', 'public_html', '../', 'column', 'cmd',
'cookie', 'from',
'where', 'exec', 'shell', 'wget', 'axel', 'curl', 'truncate', '/*',
'0x3a', 'null', 'bun', 's@bun', '%', 'char', 'or%', 'insert',
"'='", "'or'");
    $b = count($block);
    $url = strtolower($_SERVER['REQUEST_URI']);
    $url2 = strtolower($_SERVER['QUERY_STRING']);
    for ($i=0; $i<$b; $i++) {
        if(strpos($data, $block[$i]) || strpos($data,
$block[$i]) || strpos($url, $block[$i]) || strpos($url,
$block[$i]) || strpos($url2, $block[$i]) || strpos($url2,
$block[$i])){
            Redirect(base_url.'/noob.html');
            exit();
        }
    }
}

```



```

        } else {
            return $data;
        }
    }
}

function Redirect($url = '/') {
    header('location:'.$url);
}

function DateTime() {
    return Date('H:i:s d M Y', time());
}

function Message($no, $msg = '') {
    if($no == 1) {
        $type = 'success';
    } else if($no == 2) {
        $type = 'info';
    } else if($no == 3) {
        $type = 'warning';
    } else if($no == 4) {
        $type = 'danger';
    }
    $_SESSION[$type]['Message'] =
    !isset($_SESSION[$type]['Message']) ? $msg :
    $msg.'  
'.$_SESSION[$type]['Message'];
}

function ViewMessage() {
    if(isset($_SESSION['success']['Message'])) {
        $type = 'success';
    } else if(isset($_SESSION['info']['Message'])) {
        $type = 'info';
    } else if(isset($_SESSION['warning']['Message'])) {
        $type = 'warning';
    } else if(isset($_SESSION['danger']['Message'])) {
        $type = 'danger';
    }

    if(isset($type)) {
        $msg = trim($_SESSION[$type]['Message']);
        unset($_SESSION[$type]['Message']);
        echo '<div class="alert alert-block alert-'. $type .' fade in">
            <button data-dismiss="alert" class="close close-sm"
type="button">
                <i class="fa fa-times"></i>
            </button>'. $msg .'</div>';
    }
}

```

Script 2. Database.class.php

```
<?php

class config {
    public $db, $result, $mysqli_flag = MYSQLI_ASSOC; //
    MYSQLI_BOTH or MYSQLI_ASSOC or MYSQLI_NUM

    //Sets up database link using variables from a config file
    public function __construct($hostname = '', $username =
    'root', $password = '', $database = '') {
        $this->db = mysqli_connect($hostname, $username,
        $password, $database) or
        //die('Failed to connect to MySQL:
        ('.mysqli_connect_errno().')<br><br>'.mysqli_connect_error().')<br>');

    die('<html><head><title>Maintenance</title></head><body><center><h
    1><b>Under Maintenance :</b></h1></body></html>');
    }

    // Preform any sql query
    public function go($query = null){
        $this->use_log = 'go: ';
        if ($query != null) {
            $this->result = mysqli_query($this->db, $query) or
            die('Error executing query: ('.mysqli_errno($this-
            >db).')<br><br>'.mysqli_error($this->db).')<br>'. $query);
            return $this->result;
        } else {
            return false;
        }
    }

    // Return array with one result
    public function fetchArray(){
        if (isset($this->result) && !(empty($this->result))) {
            return mysqli_fetch_array($this->result, $this-
            >mysqli_flag);
        } else {
            return false;
        }
    }

    // Fetches one row of data from the result set and returns it
    as an enumerated array
    public function fetchRows(){
        if (isset($this->result) && !(empty($this->result))) {
            return mysqli_fetch_row($this->result);
        } else {
            return false;
        }
    }

    // Number of rows returned from last called query
    public function numRows() {
```

```

        if (isset($this->result) && !(empty($this->result))) {
            return mysqli_num_rows($this->result);
        } else {
            return false;
        }
    }

    // Charset
    public function charset($set){
        if (isset($this->db) && !(empty($this->db))) {
            if (mysqli_set_charset($this->db, $set)) {
                return true;
            } else {
                return false;
            }
        } else {
            return false;
        }
    }

    // Closes database connection
    public function close(){
        if (isset($this->db) && !(empty($this->db))) {
            if (mysqli_close($this->db)) {
                return true;
            } else {
                return false;
            }
        } else {
            return false;
        }
    }
}

```

Script 3. *Loader.php*

```

<?php

$database = 'database.class.php';
$function = 'function.inc.php';
$coreapi = 'api_mikrotik.php';

// do not edit this code
$root = $_SERVER['DOCUMENT_ROOT']. "/";
if(file_exists($root.'/core/'.$database)){ //db
    require $root.'/core/'.$database;
    if(file_exists($root.'/core/'.$function)){ //fungsi
        require $root.'/core/'.$function;
        if(file_exists($root.'/core/'.$coreapi)){ //api
mikrotik
            require $root.'/core/'.$coreapi;
        } else {
            die("file <b>".$coreapi. "</b> tidak ada");
        }
    }
}

```

```

    } else {
        die("file <b>".$function. "</b> tidak ada");
    }
} else {
    die("file <b>".$database. "</b> tidak ada");
}

```

Script 4. *Config.php*

```

<?php

error_reporting(0);
session_start();
ob_start();
date_default_timezone_set("Asia/Jakarta");
// Open loader
require 'loader.php';
// Open Connection
define('db_host', 'localhost'); // hostname
define('db_user', 'root'); // mysql username
define('db_pass', 'toor'); // mysql password
define('db_name', 'db_simuh'); // mysql database
// Settings
define('base_url', 'http://simuh.sukmo.dev'); // Base URL
define('app_name', 'SiMUH');
define('app_tagline', 'Sistem Manajemen User Hotspot');
define('app_desc', 'Sistem Manajemen User Hotspot Pondok Pesantren
Al-Luqmaniyyah - Yogyakarta');
// Do not edit this code
$db = new config(db_host,db_user,db_pass,db_name); // Connection
if(!$db) {
    die('There is something were wrong :( ');
}

```

Script 5. *Auth.php*

```

<?php

//router connect
$db->go("SELECT * FROM `tbl_config`");
$rmik = $db->fetchArray();
$conroute=RouterOS::connect($rmik['router_ip'],
$rmik['router_user'], $rmik['router_pass']);

$username = $_SESSION['username'];
$level = $_SESSION['level'];
if ($_SESSION['level']=='admin') {
    $db->go("SELECT * FROM `tbl_admin` WHERE `username` =
'$username'");
    $row = $db->fetchArray();
    $ids=$row['id_admin'];
    $fullname=$row['nama'];
}

```

```

}elseif($_SESSION['level']=='member'){
    $db->go("SELECT * FROM `tbl_member` WHERE `username` =
'$_username'");
    $row = $db->fetchArray();
    $ids=$row['id_member'];
    $fullname=$row['nama'];
}
if(empty($_username) && empty($_level)) {
    Redirect(base_url.'/login.php'); }
// }elseif(!empty($_username) && !empty($_level)) {
//     Redirect(base_url.'/index.php');
// }

```

Script 6. *Signin.php*

```

<?php
require '../core/config.php';
if($_POST['level']=='member'){
    $uname = Filter($_POST['username']);
    $db->go("SELECT `username`, `password` FROM `tbl_member`
WHERE `username` = '$uname'");
    $row = $db->fetchArray();

    $passwd = $_POST['password'];

    if($db->numRows() == 0) {
        Message(2, 'Akun belum terdaftar');
    } else if($row['password'] != $passwd) {
        Message(4, 'Password salah');
    } else {
        $username = $row['username'];
        $level = $_SESSION['level'] = 'member';
        $a = $_SESSION['username'] = $username;
        if($a && $level){
            Redirect(base_url.'/index.php');
            exit();
        } else {
            Message(3, 'Tolong hubungi admin');
        }
    }
}
elseif($_POST['level']=='admin'){
    $uname = Filter($_POST['username']);
    $db->go("SELECT `username`, `password` FROM `tbl_admin`
WHERE `username` = '$uname'");
    $row = $db->fetchArray();

    $passwd = md5($_POST['password']);

    if($db->numRows() == 0) {
        Message(2, 'Akun belum terdaftar');
    } else if($row['password'] != $passwd) {

```

```

        Message(4, 'Password salah');
    } else {
        $username = $row['username'];
        $level = $_SESSION['level'] = 'admin';
        $a = $_SESSION['username'] = $username;
        if($a && $level){
            Redirect(base_url.'/index.php');
            exit();
        } else {
            Message(3, 'Tolong hubungi admin');
        }
    }
}
Redirect(base_url.'/login.php');

```

Script 7. *Setting.php*

```

<?php
require '../core/config.php'; require '../inc/auth.php';
if(isset($_POST['save'])){
    $nama = Filter($_POST['rname']);
    $rip = Filter($_POST['rip']);
    $ruser = Filter($_POST['ruser']);
    $rpass = Filter($_POST['rpass']);

    $a = $db->go("UPDATE `tbl_config` SET `router_name` =
'$nama', `router_ip` = '$rip', `router_user` = '$ruser',
`router_pass` = '$rpass'");
    if($a){
        Message(1, 'Konfigurasi telah disimpan');
    }else{
        Message(3, 'Konfigurasi gagal disimpan!!');
    }
}

if(isset($_POST['test'])){
    $rip = Filter($_POST['rip']);
    $ruser = Filter($_POST['ruser']);
    $rpass = Filter($_POST['rpass']);

    $conroute = RouterOS::connect($rip, $ruser, $rpass);
    if($conroute){
        Message(1, 'Sambungan sukses');
    }else{
        Message(3, 'Sambungan gagal!!');
    }
}

Redirect(base_url.'/admin/config.php');

```

Script 8. Add-paket.php

```
<?php

require '../core/config.php'; require '../inc/auth.php';
$namapaket=Filter($_POST['namapaket']);
$type=Filter($_POST['piltipe']);
$masaaktif = Filter($_POST['masaaktif']);
$bw['0'] = Filter($_POST['upload'].'k');
$bw['1'] = Filter($_POST['download'].'k');
$band=implode("/", $bw);

if(isset($_POST['simpan'])){
    $db->go("SELECT `nama_paket` FROM `tbl_paket` WHERE
`nama_paket` = '$namapaket'");
    $cep=$db->fetchArray();
    if($cep['nama_paket']!=NULL){
        //Message(3, 'Maaf, Nama paket sudah ada dan
tidak boleh sama.');
```

Redirect(base_url.'/admin/add-paket.php');

```
    }elseif($cep['nama_paket']==NULL){
        if($type=='0'){
            if(empty($namapaket) || empty($bw['0']) ||
empty($bw['1'])){
                Message(2, 'Form ada yang kosong');
```

}else{

```
                if($conroute){
                    $conroute-
>add("/ip/hotspot/user/profile", array("name"=>"$namapaket",
"rate-limit"=>"$band"));
                    $a = $db->go("INSERT INTO tbl_paket
(nama_paket, jenis_paket, bandwidth, masa_aktif) VALUES
('$namapaket','$type','$band', 'Unlimited')");
                }else{
                    Message(2, 'Koneksi ke-server tidak
tersambung');
```

}

```
                }
            }elseif($type=='1'){
                if(empty($namapaket) || empty($masaaktif) ||
empty($bw['0']) || empty($bw['1'])){
                    Message(2, 'Form ada yang kosong');
```

}else{

```
                if($conroute){
                    $conroute-
>add("/ip/hotspot/user/profile", array("name"=>"$namapaket",
"rate-limit"=>"$band"));
                    $a = $db->go("INSERT INTO tbl_paket
(nama_paket, jenis_paket, bandwidth, masa_aktif) VALUES
('$namapaket','$type','$band','$masaaktif')");
                }else{
                    Message(2, 'Koneksi ke-server tidak
tersambung');
```

}

```
                }
            }
        }
    }
}
```

```

    }
}

if($a){
    Message(1, 'Data telah ditambah');
}else{
    Message(4, 'Data gagal ditambah!!');
}
Redirect(base_url.'/admin/add-paket.php');

```

Script 9. edt-paket.php

```

<?php
require '../core/config.php'; require '../inc/auth.php';
$namapaket=Filter($_POST['namapaket']);
$type=Filter($_POST['piltype']);
$masaaktif = Filter($_POST['masaaktif']);
$bw['0'] = Filter($_POST['upload'].'k');
$bw['1'] = Filter($_POST['download'].'k');
$band=implode("/", $bw);

if(isset($_POST['simpan'])){
    if($type=='0'){
        if($conroute){
            $conroute->set("/ip/hotspot/user/profile",
array(".id"=>"$namapaket", "rate-limit"=>"$band"));
            $a = $db->go("UPDATE tbl_paket SET
bandwidth='$band' WHERE nama_paket = '$namapaket'");
        }else{
            Message(2, 'Koneksi ke-server tidak
tersambung');
        }
    }elseif($type=='1'){
        if($conroute){
            $conroute->set("/ip/hotspot/user/profile",
array(".id"=>"$namapaket", "name"=>"$namapaket", "rate-
limit"=>"$band"));
            $a = $db->go("UPDATE tbl_paket SET
masa_aktif='$masaaktif', bandwidth='$band' WHERE nama_paket =
'$namapaket'");
        }else{
            Message(2, 'Koneksi ke-server tidak
tersambung');
        }
    }
}

if($a){
    Message(1, 'Data telah diperbarui');
}else{
    Message(4, 'Data gagal diperbarui!!');
}
Redirect(base_url.'/admin/paket.php');

```


Script 10. Del-paket.php

```

<?php
require '../core/config.php'; require '../inc/auth.php';
$id=Filter($_GET['id']);
$idm=Filter($_GET['del']);
if($_SESSION['level']=='admin'){
if(isset($_GET['id']) && isset($_GET['del'])){
    if($conroute){
        $conroute->remove("/ip/hotspot/user/profile",
array(".id"=>"$idm"));
        $a = $db->go("DELETE FROM tbl_paket WHERE `id_paket` =
'$id'");
    }
}

if($a){
    Message(1, 'Data telah dihapus');
}else{
    Message(4, 'Data gagal dihapus!!');
}
Redirect(base_url.'/admin/paket.php');
}

```

Script 11. Krm-pesan.php

```

<?php
require '../core/config.php'; require '../inc/auth.php';
if($_SESSION['level']=='admin'){
    $db->go("SELECT * FROM `tbl_admin` WHERE `username` =
'$username'");
    $row = $db->fetchArray();

    $id=$row['id_admin'];
    $pilmember=Filter($_POST['pilmember']);
    $subyek = Filter($_POST['subyek']);
    $pesan = Filter($_POST['pesan']);

    if(isset($_POST[' kirim'])){
        if(empty($pilmember) || empty($subyek) ||
empty($pesan)){
            Message(2, 'Form ada yang kosong');
        }else{
            $a = $db->go("INSERT INTO tbl_pesan (id_sender,
id_receiver, subject, pesan) VALUES
('$id', '$pilmember', '$subyek', '$pesan')");
        }
    }

    if($a){
        Message(1, 'Pesan telah dikirim');
    }else{
        Message(4, 'Pesan gagal dikirim!!');
    }
}

```

```

    }
    Redirect(base_url.'/admin/add-pesan.php');
}elseif($_SESSION['level']=='member'){
    $db->go("SELECT * FROM `tbl_member` WHERE `username` =
'$username'");
    $row = $db->fetchArray();

    $id=$row['id_member'];
    $pilmember=Filter($_POST['pilmember']);
    $subyek = Filter($_POST['subyek']);
    $pesan = Filter($_POST['pesan']);

    if(isset($_POST['kirim'])){
        if(empty($pilmember) || empty($subyek) ||
empty($pesan)){
            Message(2, 'Form ada yang kosong');
        }else{
            $a = $db->go("INSERT INTO tbl_pesan (id_sender,
id_receiver, subject, pesan) VALUES
('$id','$pilmember','$subyek','$pesan')");
        }
    }

    if($a){
        Message(1, 'Pesan telah dikirim');
    }else{
        Message(4, 'Pesan gagal dikirim!!');
    }
    Redirect(base_url.'/member/add-pesan.php');
}

```

Script 12. Del-pesan.php

```

<?php
require '../core/config.php'; require '../inc/auth.php';
$id=Filter($_GET['id']);

if($_SESSION['level']=='admin'){
    if(isset($_GET['id'])){
        $a = $db->go("DELETE FROM tbl_pesan WHERE `id_pesan` =
'$id' AND `id_receiver` = '$ids'");
    }

    if($a){
        Message(1, 'Data telah dihapus');
    }else{
        Message(4, 'Data gagal dihapus!!');
    }
    Redirect(base_url.'/admin/pesan.php');
}elseif($_SESSION['level']=='member'){
    if(isset($_GET['id'])){
        $a = $db->go("DELETE FROM tbl_pesan WHERE `id_pesan` =
'$id' AND `id_receiver` = '$ids'");
    }
}

```

```

        if($a){
            Message(1, 'Data telah dihapus');
        }else{
            Message(4, 'Data gagal dihapus!!');
        }
        Redirect(base_url.'/member/pesan.php');
    }
}

```

Script 13. *Add-member.php*

```

<?php

require '../core/config.php'; require '../inc/auth.php';
$username=Filter($_POST['username']);
$password=Filter($_POST['password']);
$nama = Filter($_POST['nama']);
$alamat = Filter($_POST['alamat']);
$kontak = Filter($_POST['kontak']);

if(isset($_POST['save'])){
    if($db->go("SELECT `username` FROM `tbl_member` WHERE
`username` = '$username'")){
        Message(3, 'Username sudah ada');
        Redirect(base_url.'/admin/add-member.php');
    }
    if(empty($username) || empty($password) || empty($nama) ||
empty($alamat) || empty($kontak)){
        Message(2, 'Form ada yang kosong');
    }else{
        if($conroute){
            $conroute->add("/ip/hotspot/user",
array("name"=>"$username", "password"=>"$password",
"disabled"=>"yes"));
            $a = $db->go("INSERT INTO tbl_member (username,
password, nama, alamat, telp) VALUES
('$username', '$password', '$nama', '$alamat', '$kontak')");
        }else{
            Message(2, 'Koneksi ke-server tidak
tersambung');
        }
    }
}

if($a){
    Message(1, 'Data telah ditambah');
}else{
    Message(4, 'Data gagal ditambah!!');
}
Redirect(base_url.'/admin/add-member.php');

```

Script 14. Edt-member.php

```

<?php

require '../core/config.php'; require '../inc/auth.php';
$id=Filter($_POST['idmember']);
$idm=Filter($_POST['uname']);
$nama = Filter($_POST['nama']);
$alamat = Filter($_POST['alamat']);
$kontak = Filter($_POST['kontak']);
$password = Filter($_POST['password']);

if(isset($_POST['save'])) {
    if(empty($_POST['password'])) {
        $a = $db->go("UPDATE `tbl_member` SET `nama` =
'$nama', `alamat` = '$alamat', `telp` = '$kontak' WHERE
`id_member` = '$id'");
    } else {
        if($conroute) {
            $conroute->set("/ip/hotspot/user",
array(".id"=>"$idm", "password"=>"$password"));
            $a = $db->go("UPDATE `tbl_member` SET
`password` = '$password', `nama` = '$nama', `alamat` = '$alamat',
`telp` = '$kontak' WHERE `id_member` = '$id'");
        }
    }
}

if($a) {
    Message(1, 'Data telah diupdate');
} else {
    Message(4, 'Data gagal diupdate!!');
}
Redirect(base_url.'/admin/member.php');

```

Script 15. Del-member.php

```

<?php

require '../core/config.php'; require '../inc/auth.php';
$id=Filter($_GET['id']);
$idm=Filter($_GET['del']);
if($_SESSION['level']=='admin') {
    if(isset($_GET['id']) && isset($_GET['del'])) {
        if($conroute) {
            $conroute->remove("/ip/hotspot/user",
array(".id"=>"$idm"));
            $a = $db->go("DELETE FROM tbl_member WHERE `id_member` =
'$id'");
        }
    }
}

if($a) {

```

```
        Message(1, 'Data telah dihapus');  
    }else{  
        Message(4, 'Data gagal dihapus!!');  
    }  
    Redirect(base_url.'/admin/member.php');  
}
```

Script 16. Logout.php

```
require '../core/config.php'; require '../inc/auth.php';  
session_destroy();  
Redirect(base_url.'/login.php');
```

Lampiran 2: Hasil Kuisisioner

DAFTAR RIWAYAT HIDUP

Nama : Khoirul Anwar
Tempat, tanggal lahir : Pati, 11 Februari 1991
Jenis Kelamin : Laki-laki
Agama : Islam
Alamat Asal : Mojo 02/V – Cluwak – Pati
No. HP : 085729815915
Email : dev@ipnudi.or.id

Riwayat Pendidikan

1. MI Islamiyah 02 Mojo (1996-2002)
2. MTs Darul Falah Sirahan (2002-2005)
3. SMK N 2 Sragen Kelas Jauh di Ponpes. Al-Musanni, Gemolong (2005-2008)
4. S1 Teknik Informatika Universitas PGRI Yogyakarta (2010-2017)

