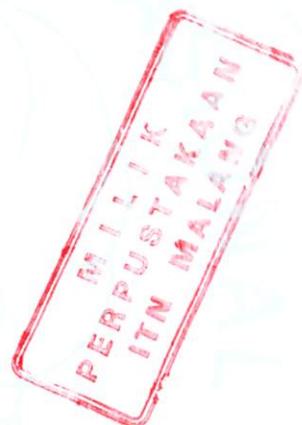


RANCANG BANGUN APLIKASI GUI MIKROTIK BERBASIS ANDROID

SKRIPSI



Disusun Oleh :

**GIO VANDY
08.18.095**

**PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2013**

СИАХНГО АЛГОРІТМ НУЗ ТАКІЛІР АУДАЛА ОНАДЫ
СИНОДА

ДЕСЯТЬ



ОТВЕРСИЕ

СИМВОЛ

600.21.20

1-2 АМЕРИКАНСКИЙ БЛЮЗ МАРШИН
ИМЕЮЩИЙ СОВОКУСНЫЙ ГАРМОНИК
ПРОДАЕТ САМЫЙ ПРОЧНЫЙ ПУСТИННЫЙ
БРОС

LEMBAR PERSETUJUAN

RANCANG BANGUN APLIKASI GUI MIKROTIK BERBASIS ANDROID

SKRIPSI

*Disusun dan diajukan untuk melengkapi dan memenuhi persyaratan
guna mencapai gelar Sarjana Komputer*

Disusun Oleh :

GIO VANDY

08.18.095

Diperiksa dan Disetujui,

Dosen Pembimbing I

Joseph Dedy Irawan, ST, MT.
NIP. 197404162005011002

Dosen Pembimbing II

Sonny Prasetyo, ST, MT.
NIP.P. 1031000433

Mengetahui,

Ketua Program Studi Teknik Informatika S-1

Joseph Dedy Irawan, ST, MT
NIP. 197404162005011002

PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2013

RANCANG BANGUN APLIKASI GUI MIKROTIK BERBASIS ANDROID

GIO VANDY

**Program Studi Teknik Informatika S-1,
Fakultas Teknologi Industri, Institut Teknologi Nasional Malang
Jln. Raya Karanglo Km 2 Malang**

Email: giovandy.sallolo@gmail.com

**Dosen Pembimbing : 1. Joseph Dedy Irawan, ST., MT.
2. Sonny Prasetyo. ST., MT.**

ABSTRAK

Android merupakan generasi baru platform mobile, platform yang memberikan pengembang untuk melakukan pengembangan sesuai dengan yang diharapkan. Seperti yang dibahas pada skripsi ini yaitu penerapan aplikasi winbox pada android, dimana winbox sendiri adalah sebuah aplikasi untuk meremote router mikrotik dengan mode tampilan GUI. Tujuan penelitian ini adalah menyediakan aplikasi winbox berbasis android dimana aplikasi ini pengembangan dari winbox yang berjalan pada pc. Dengan adanya aplikasi winbox berbasis android user dapat dengan mudah mengontrol mikrotik.

Metode penelitian yang digunakan pada skripsi ini adalah metode analisis dan metode perancangan, dimana pada metode analisis menganalisa data dan informasi yang telah diperoleh agar didapatkan suatu kerangka yang digunakan untuk acuan perancangan sistem pembuatan program dengan IDE Eclipse sebagai pengembangan pemrogramannya.

Hasil yang dicapai berupa suatu aplikasi untuk smartphone android yang dapat meremote mikrotik dan memberikan informasi dari jaringan mikrotik tersebut. Kesimpulan yang didapat dari penulis skripsi ini adalah sebuah aplikasi winbox yang berjalan pada smartphone android yang bisa digunakan untuk mengontrol mikrotik.

Kata Kunci : Mikrotik, Android



PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL
MALANG

PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertandatangan di bawah ini:

Nama : Gio vandy
Nim : 08.18.095
Program Studi : Teknik Informatika S-1
Fakultas : Fakultas Teknologi Industri

Menyatakan dengan sesungguhnya bahwa Skripsi saya yang berjudul:

“RANCANG BANGUN APLIKASI GUI MIKROTIK BERBASIS ANDROID”

Adalah Skripsi saya sendiri bukan duplikat serta mengutip atau menyadur seluruhnya karya orang lain kecuali dari sumber aslinya.

Malang, 23 Maret 2013

Yang membuat pernyataan



KATA PENGANTAR

Puji syukur penulis panjatkan kehadapan Tuhan Yang Maha Esa, sehingga penulis dapat menyelesaikan skripsi dengan judul "**RANCANG BANGUN APLIKASI GUI MIKROTIK BERBASIS ANDROID**" dengan lancar. Skripsi merupakan persyaratan kelulusan di Program Studi Teknik Informatika S-1 ITN Malang dan untuk mencapai gelar Sarjana Komputer.

Keberhasilan penyelesaian laporan skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak. Untuk itu penyusun menyampaikan terima kasih kepada :

1. Bapak Ir. Soeparno Djivo, MT. selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. Sidik Noetjahjono, MT. selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
3. Bapak Joseph Dedy Irawan, ST., MT. selaku Ketua Program Studi Teknik Informatika S-1, sekaligus sebagai Dosen Pembimbing I.
4. Bapak Sonny Prasetio, ST., MT. selaku Dosen Pembimbing II.
5. Orangtua yang selalu memberikan semangat dan motivasi selama penyusunan skripsi.
6. Teman-teman dan semua yang telah membantu dalam penyelesaian penyusunan skripsi ini.

Penulis telah berusaha semaksimal mungkin dan menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan ini. Untuk itu penyusun mengharapkan saran dan kritik yang membangun dari pembaca demi kesempurnaan laporan ini.

Harapan penulis semoga laporan skripsi ini memberikan manfaat bagi perkembangan ilmu pengetahuan dan pembaca.

Malang, Februari 2013

Penulis

DAFTAR ISI

	Halaman
LEMBAR PERSETUJUAN.....	i
ABSTRAKSI.....	ii
SURAT PERNYATAAN SKRIPSI.....	iii
KATA PENGANTAR.....	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR	vii
DAFTAR TABEL	ix
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah	2
1.3. Tujuan.....	2
1.4. Batasan Masalah	2
1.5. Metode Pemecahan Masalah.....	3
1.6. Sistematika Penulisan.....	3
BAB II LANDASAN TEORI	5
2.1. Java	5
2.2. Extensible Markup Language (XML)	6
2.3. Android.....	6
2.3.1. Fitur Android	8
2.3.2. Arsitektur Android	9
2.3.3. Activity Lifecycle.....	13
2.3.4. Versi Android	15
2.4. Eclipse.....	21
2.5. Mikrotik.....	22
2.6. Winbox Loader.....	24
BAB III PERANCANGAN SISTEM	25
3.1. Analisa Masalah	25
3.2. Analisa Kebutuhan	25
3.2.1. Analisa Perangkat Keras (Hardware)	25
3.3.2. Analisa Perangkat Lunak (Software)	27
3.3. Spesifikasi Aplikasi.....	27
3.4. Perancangan Aplikasi.....	27

3.4.1. Perangcangan Form Aplikasi	27
3.5. Contex Diagram	35
3.6. Perancangan Sistem.....	36
BAB IV IMPLEMENTASI DAN PENGUJIAN	38
4.1. Implementasi Sistem	38
4.2. Hasil Implementasi.....	38
4.2.1. Tampilan Icon Aplikasi	38
4.2.2. Tampilan Menu Utama Aplikasi	39
4.2.3. Tampilan Login	39
4.2.4. Tampilan Main Menu	40
4.3. Pengujian	47
4.3.1. Pengujian Fungsional	47
4.3.2 Tabel Pengujian User.....	48
BAB V PENUTUP	49
5.1. Kesimpulan.....	49
5.2. Saran.....	49
DAFTAR PUSTAKA	50
LAMPIRAN	

DAFTAR GAMBAR

	Halaman	
Gambar 2.1	Logo Android	7
Gambar 2.2	Arsitektur Android	10
Gambar 2.3	Android Activity Lifecircle	13
Gambar 2.4	Android versi 1.0 pada HTC Dream.....	15
Gambar 2.5.	Tampilan layar utama Android versi 1.5	15
Gambar 2.6.	Fitur <i>Gesture</i> pada Android versi 1.6	16
Gambar 2.7.	Live wallpaper pada Android versi 2.1	17
Gambar 2.8	Tampilan layar utama Android versi 2.2	17
Gambar 2.9	Android versi 2.3	18
Gambar 2.10	Tampilan layar utama Android versi 3.0 pada <i>tablet</i>	19
Gambar 2.11	Tampilan layar utama dan menu pada Android versi 4.0	20
Gambar 2.12	Tampilan lembar kerja Eclipse	22
Gambar 3.1	Rancangan tampilan Awal	27
Gambar 3.2	Rancangan tampilan login	28
Gambar 3.3	Rancanga tampilan Interfaces	28
Gambar 3.4	Rancangan menu PPP	29
Gambar 3.5	Rancangan PPP profiles	29
Gambar 3.6	Rancangan tampilan IPV4 menu	30
Gambar 3.7	Rancangan tampilan IPv4 addres	30
Gambar 3.8	Rancangan tampilan current IPv4 router	31
Gambar 3.9	Rancangan tampilan current DNS	31
Gambar 3.10	Rancangan tampilan DNS cache	32
Gambar 3.11	Rancangan tampilan IPv4 firewall	32
Gambar 3.12	Rancangan tampilan IP Client	33
Gambar 3.13	Rancangan tampilan IPv4DHCP Server Menu	33
Gambar 3.14	Rancangan tampilan System menu	34
Gambar 3.15	Rancangan tampilan Tools Menu	34
Gambar 3.16	Diagram Contex	35
Gambar 3.17	Flowchart Sistem	36
Gambar 4.1	Tampilan <i>icon</i> pada menu utama	38
Gambar 4.2	Tampilan Menu Utama Aplikasi	39
Gambar 4.3	Tampilan Login	40

Gambar 4.4	Tampilan Main Menu	40
Gambar 4.5	Tampilan Interfaces	41
Gambar 4.6	Tampilan PPP	42
Gambar 4.7	Tampilan IP	42
Gambar 4.8	Tampilan Sub Menu Address	43
Gambar 4.9	Tampilan Sub Menu Route	43
Gambar 4.10	Tampilan Sub Menu Firewall	44
Gambar 4.11	Tampilan Sub Menu DNS	44
Gambar 4.12	Tampilan Sub Menu DHCP Server	45
Gambar 4.13	Tampilan IP Client	45
Gambar 4.14	System Menu	46
Gambar 4.15	Tampilan Tools Menu	46

DAFTAR TABEL

	Halaman
Tabel 3.1 Spesifikasi Samsung Galaxy Ace Duos.....	26
Tabel 4.1. Menunjukkan jika aplikasi winbox berbasis Android bisa berjalan Pada Platform Android tersebut	47
Tabel 4.2. Tabel Pengujian User Aplikasi Winbox Berbasis Android	48

BAB I

PENDAHULUAN

1.1. Latar Belakang

Di jaman yang serba modern sekarang ini banyak inovasi teknologi baru dan juga banyak terdapat program penunjang yang bisa di gunakan oleh berbagai kalangan masyarakat baik dari anak kecil maupun orang tua. Contoh nyatanya adalah pemanfaatan aplikasi android dan inovasi dari aplikasi android. Android merupakan merupakan generasi smartphone dimana aplikasi ini merupakan aplikasi yang memiliki berbagai macam fitur pendukung yang memudahkan kita dalam pemanfaatan aplikasi ini dengan baik dan cepat. Pesatnya pertumbuhan android karena android itu sendiri adalah platform yang sangat lengkap baik itu sistem aplikasinya, aplikasi Tool dan pengembangannya. Android dipuji sebagai “platform mobile pertama yang lengkap, terbuka dan bebas”.

Android merupakan generasi baru platform mobile, platform yang memberikan pengembang untuk melakukan pengembangan sesuai dengan yang diharapkan. Seperti yang dibahas pada skripsi ini yaitu penerapan aplikasi winbox pada android, dimana winbox sendiri adalah sebuah aplikasi untuk me-remote router-OS Mikrotik dengan mode tampilan GUI. Karna winbox merupakan salah satu kontribusi yang tepat pada proses konfigurasi jaringan tapi pada umunya winbox hanya berjalan pada PC, oleh karna itu untuk pengembangan aplikasi penulis menerapkan aplikasi winbox ke android. Dimana platform android merupakan system operation opensource sehingga memudahkan dalam mengembangkan aplikasi. Dengan winbox berbasis android user dapat dengan mudah memantau jaringan yang terhubung dengan OS Mikrotik seperti: memantau IP client yang mengakses jaringan, melihat interfaces dari jaringan, dan mengetahui informasi tentang nama host atau pun nama domain melalui tools DNS. Sedangkan Mikrotik sendiri adalah Operating

system (OS) yang dibuat khusus untuk router dan bisa digunakan dalam jaringan internet. Dengan sistem operasi android yang bersifat open source maka pengguna sistem operasi android di Indonesia dapat dimudahkan berinteraksi dengan mobile phone tersebut.

1.2. Rumusan Masalah

Adapun rumusan masalah yang akan dibahas adalah:

Bagaimana cara membuat aplikasi winbox yang dapat berjalan pada android

1.3 Tujuan

Tujuan pembuatan aplikasi ini adalah untuk lebih memudahkan dalam memantau jaringan yang terhubung dengan OS Mikrotik seperti:

1. Melihat IP Client
2. Melihat Interfaces dari jaringan
3. Mengetahui Informasi tentang nama host ataupun nama domain melalui tools DNS.

Hanya dengan menggunakan sistem operasi (Android)

1.4 Batasan Masalah

Batasan masalah pembahasan yang diambil agar sesuai dengan tujuan dan tidak terjadi penyimpangan maksud dan tujuan utama, maka ditentukan ruang lingkup pembahasan sebagai berikut:

1. aplikasi ini dapat berjalan pada system operasi android versi 2.3
2. Aplikasi dibangun dengan bahasa pemrograman java, xml dan Eclipse sebagai tools pengembangannya.
3. Aplikasi ini hanya bisa untuk menampilkan trafik jaringan
4. penggunaan aplikasi ini hanya bisa digunakan dengan koneksi internet wifi
5. aplikasi ini hanya menggunakan fungsi API dari mikrotik sebagai penghubung antara aplikasinya dengan mikrotik,dimana API yang aktif pada mikrotik yaitu port 8728.

1.5 Metode Pemecahan Masalah

Metode yang digunakan dalam pembahasan ini adalah:

1. Studi Literatur

Pengumpulan Data dilakukan dengan mencari bahan – bahan referensi dari berbagai sumber sebagai landasan teori yang berhubungan dengan permasalahan yang dijadikan objek penelitian.

2. Analisa kebutuhan Sistem

Data dan informasi yang telah diperoleh akan dianalisa agar didapatkan suatu kerangka yang digunakan untuk acuan perancangan sistem pembuatan program

3. Eksperimen dan Evaluasi

Pada tahap ini, sistem yang telah selesai dibuat akan diuji coba berdasarkan fungsional program dan akan dilakukan koreksi.

1.6 Sistematika Penulisan

Dalam penyusunan skripsi ini agar lebih mudah dipahami maka dibuatlah suatu sistematika penulisan sebagai berikut:

1. BAB I : Pendahuluan

Pada bab ini membahas tentang Latar Belakang, Tujuan dan Manfaat Penelitian, Batasan Masalah, dan Sistematika Penulisan.

2. BAB II : Landasan Teori.

Pada bab ini membahas tentang teori-teori ilmiah yang berhubungan dengan konsep-konsep yang akan dipakai dalam penelitian, pembahasan hasil-hasil kajian ilmiah lain yang berhubungan dengan penelitian, rangkuman hasil-hasil kajian teori

3. BAB III : Perancangan Sistem

Pada bab ini dibahas tentang perancangan dari sistem aplikasi yang dibuat, seperti: *Data Flow Diagram* (DFD), *flowchart*, instalasi aplikasi pendukung, dan lain-lain.

4. BAB IV : Implementasi dan Pengujian

Pada bab ini dibahas tentang tentang listing program, pembuatan tampilan layout aplikasi, dan pengujian aplikasi.

5. BAB V : Penutup

Pada bab ini membahas tentang kesimpulan dan saran.

BAB II

LANDASAN TEORI

Pada bab ini akan dijelaskan mengenai teori-teori yang berhubungan dengan penelitian ini, sehingga dapat dijadikan sebagai landasan berpikir dan akan mempermudah dalam hal pembahasan hasil utama pada bab berikutnya. Adapun teori-teori tersebut mencakup pengertian Java, XML, Android, Eclipse, dan Mikrotik.

2.1 Java^[8]

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk *mobile phone*. Bahasa ini dibuat oleh James Gosling saat masih bergabung di “Sun Microsystems” saat ini merupakan bagian dari “Oracle” dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana.

Aplikasi-aplikasi berbasis Java umumnya dikompilasi ke dalam p-code (*bytecode*) dan dijalankan pada *Java Virtual Machine* (JVM). Karena fungsionalitasnya yang memungkinkan aplikasi Java mampu berjalan di beberapa *platform*, Java memiliki slogan, "Tulis sekali, jalankan di mana pun". Saat ini Java dimanfaatkan dalam pengembangan berbagai jenis aplikasi berbasis *desktop* ataupun aplikasi berbasis web. Kelebihan Bahasa Pemrograman Java adalah sebagai berikut:

1. *Multiplatform*

Kelebihan utama dari Java ialah dapat dijalankan di beberapa *platform*. Dengan kelebihan ini *programmer* cukup menulis sebuah program Java dan *compile* sekali lalu hasilnya dapat dijalankan di atas beberapa *platform* tanpa perubahan. *Platform* yang didukung sampai saat ini adalah Microsoft Windows, Linux, Mac OS dan Sun Solaris.

2. Pemrogram Berorientasi Objek (*Object Oriented Programming*)

merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam *kelas*-

kelas atau *objek-objek*. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya.

3. Perpustakaan Kelas (*Class Libraries*) Yang Lengkap

Java terkenal dengan kelengkapan *library*/perpustakaan (kumpulan program-program yang disertakan dalam pemrograman Java).

4. Bergaya C++

Memiliki *source code* seperti bahasa pemrograman C++ sehingga menarik banyak pemrogram C++ untuk pindah ke Java.

5. Pengumpulan sampah otomatis (*automatic garbage collector*)

Memiliki fasilitas pengaturan penggunaan memori sehingga para *programmer* tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa C++ yang dipakai secara luas).

2.2 *Extensible Markup Language (XML)*

Extensible Markup Language (XML) adalah cara yang fleksibel untuk membuat format informasi yang umum dan dapat di-share baik format maupun data pada *World Wide Web*, *intranet*, atau di tempat lain. XML adalah “*extensible*” karena tidak seperti HTML, simbol *markup* yang dimiliki XML tidak terbatas dan bersifat *self-defining*. XML bersifat sederhana dan mudah dalam menggunakan sekumpulan *Standard Generalized Markup Language (SGML)* yang merupakan standar untuk membuat struktur dokumen.

2.3 *Android*^[4]

Android adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti *smartphone* dan *tablet PC*. Logo resmi Android berupa gambar robot berwarna hijau, seperti dalam Gambar 2.1.



Gambar 2.1. Logo Android

Android menyediakan *platform* terbuka bagi para *developer* untuk menciptakan aplikasi mereka sendiri. Awalnya, “Google Inc.” membeli “Android Inc.”, pendatang baru yang membuat *software* untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan *hardware*, *software*, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat perilisan perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan standar *open source* pada perangkat seluler. Di lain pihak, Google merilis *source code* Android di bawah lisensi Apache. Android memiliki karakteristik sebagai berikut:

1. Terbuka

Android dibangun untuk benar-benar terbuka sehingga sebuah aplikasi dapat memanggil salah satu fungsi inti perangkat seperti membuat panggilan, mengirim pesan teks, menggunakan kamera, dan lain-lain.

2. Semua aplikasi dibuat sama

Android tidak memberikan perbedaan terhadap aplikasi utama dari perangkat dan aplikasi yang dikembangkan oleh *user*. Semua aplikasi dibangun untuk memiliki akses yang sama terhadap kemampuan perangkat dalam menyediakan layanan dan aplikasi yang luas terhadap para *user*.

3. Pengembangan aplikasi yang cepat dan mudah

Android menyediakan akses yang sangat luas kepada *user* untuk menggunakan *library* yang diperlukan dan *tools* yang dapat digunakan untuk

membangun aplikasi yang semakin baik. Android memiliki sekumpulan *tools* yang dapat digunakan sehingga membantu para *developer* dalam meningkatkan produktivitas pada saat membangun aplikasi yang dibuat

Google Inc. sepenuhnya menjadikan Android *open source* sehingga para *developer* dapat menggunakan Android tanpa mengeluarkan biaya untuk lisensi dari Google dan dapat membangun Android tanpa adanya batasan-batasan. Android *Software Development Kit* (SDK) menyediakan alat dan *Application Programming Interface* (API) yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java dan XML.

2.3.1 Fitur Android

Fitur-fitur yang tersedia pada *platform* Android adalah sebagai berikut:

1. Framework aplikasi

Seperti pada umumnya, *framework* memiliki keuntungan dalam proses pengkodingan karena kita tidak perlu membuat *coding* untuk hal-hal yang pasti dilakukan seperti *coding* menampilkan gambar, *coding* koneksi database, dan lain-lain.

2. Dalvik Virtual Machine (DVM)

Andorid dibangun pada *Linux Kernel* dengan sebuah *machine virtual* untuk mengoptimalkan penggunaan memori dan *hardware* pada perangkat *mobile*. *Machine virtual* pada Android disebut dengan *Dalvik* yang bertugas sebagai *intrepreter* yang akan memetakan eksekusi memori sehingga program menjadi lebih efisien. Hasil dari Dalvik akan membentuk suatu file yang disebut dengan *Dalvik Executable (*.dex)*. Karakteristik dari *Dalvik Virtual Machine* berbasis register sehingga mampu membaca file *class* yang dibentuk dari kompilasi Java.

3. Integrated Browser

Terintegrasi dengan web *browser* berdasarkan *open source engine WebKit*.

4. Grafis

Dapat membuat aplikasi grafis 2D dan 3D karena Android memiliki *library* *openGL ES 1.0*.

5. SQLite

SQLite berperan dalam penyimpanan data (*database*)

6. Media Support

Fitur yang mendukung suara, video, dan gambar.

7. Bluetooth, EDGE, 3G, Wi-Fi

Fitur ini tidak selalu tersedia pada Android karena tergantung *hardware* atau *smartphone*.

8. Dukungan Perangkat Tambahan

Android dapat memanfaatkan kamera, layar sentuh, *accelerometer*, *magnetometers*, GPS, 2D *acceleration*, dan 3D *acceleration*.

9. Multi-touch

Dapat menggunakan dua jari atau lebih untuk berinteraksi dengan perangkat.

10. Lingkungan Development

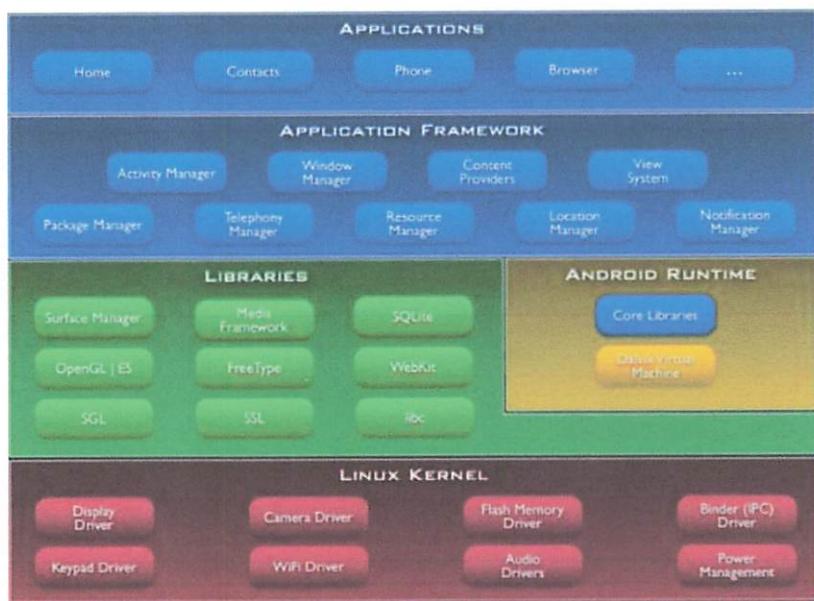
Memiliki fitur *emulator*, *tools* untuk *debugging*, profil dan kinerja memori, dan *plugin* untuk IDE Eclipse.

11. Market

Market pada Android merupakan katalog aplikasi yang dapat di-*download* dan di-*install* pada *handphone* melalui internet.

2.3.2 Arsitektur Android

Dalam paket sistem operasi Android terdiri dari beberapa unsur. Secara sederhana arsitektur Android merupakan sebuah kernel Linux dan sekumpulan pustaka/*library* C/C++ dalam suatu *framework* yang menyediakan dan mengatur alur proses aplikasi. Dalam Gambar 2.6, menjelaskan arsitektur dari Android atau komponen utama dari sistem operasi Android



Gambar 2.2. Arsitektur Android

Penjelasan dari arsitektur Android yang ditunjukkan dalam Gambar 2.2, adalah sebagai berikut:

1. *Application*

Android telah menyediakan aplikasi inti seperti: *email client*, SMS, kalender, peta, *browser*, kontak, dan lain-lain. Semua aplikasi ditulis dengan bahasa pemrograman java. Semua aplikasi berjalan pada layer aplikasi dengan menggunakan *library API* yang sama. Ini berarti semua aplikasi yang dibuat untuk Android akan memiliki akses yang setara dalam mengakses seluruh kemampuan perangkat.

2. *Application Framework*

Application framework merupakan peralatan yang digunakan oleh semua aplikasi, baik aplikasi bawaan dari ponsel, seperti daftar kontak, dan kontak SMS, maupun aplikasi yang ditulis oleh Google ataupun *Android developer*.

Developer bebas untuk mengambil keuntungan dari *hardware*, aksesoris lokasi informasi, menjalankan *background service*, mengatur *alarm*, menambahkan peringatan ke *status bar*, dan masih banyak lagi. *Developer*

memiliki akses yang penuh ke dalam *framework API* yang sama yang digunakan oleh semua pihak. Pada dasarnya, *application framework* memiliki beberapa komponen sebagai berikut:

a. *Activity Manager*

Mengatur siklus dari aplikasi dan menyediakan navigasi *backstack* untuk aplikasi yang berjalan pada proses yang berbeda.

b. *Package Manager*

Untuk melacak aplikasi yang diinstal pada perangkat.

c. *Windows Manager*

Merupakan abstraksi dari bahasa pemrograman Java pada bagian atas dari level *service* (pada level yang lebih rendah) yang disediakan oleh *Surface Manager*.

d. *Telephony Manager*

Berisi sekumpulan API yang diperlukan untuk memanggil aplikasi.

e. *Content Providers*

Digunakan untuk memungkinkan aplikasi mengakses data dari aplikasi lain (seperti kontak) atau untuk *sharing* data mereka sendiri.

f. *Resource Manager*

Digunakan untuk mengakses sumber daya yang bersifat bukan *code* seperti *string local*, *bitmap*, deskripsi dari *layout file*, dan bagian *eksternal* lain dari aplikasi.

g. *View System*

Digunakan untuk mengambil sekumpulan *button*, *list grid*, *text box*, dan lain sebagainya yang digunakan di dalam *user interface*.

h. *Notification Manager*

Digunakan untuk mengatur tampilan peringatan dan fungsi-fungsi lain.

3. Libraries

Android memiliki sekumpulan *library* C/C++ yang digunakan oleh berbagai komponen dalam sistem Android. Komponen-komponen ini digunakan oleh para *developer* melalui *application framework*. Beberapa dari *library* utama dijelaskan sebagai berikut:

a. System C Library

Merupakan implementasi turunan dari standar *system library C* (libc) yang diatur untuk peralatan berbasis *embedded Linux*.

b. Media Libraries

Merupakan *library* untuk memutar ulang dan menyimpan format suara, dan video, serta *static image file* seperti MPEG4, MP3, AAC, AMR, JPG, dan PNG.

c. Surface Manager

Mengatur akses ke dalam subsistem tampilan dan susunan grafis *layer* 2D dan 3D secara mulus dari beberapa aplikasi dan menyusun permukaan gambar yang berbeda pada layar ponsel.

d. LibWebCore

Merupakan *web browser modern* yang menjadi kekuatan bagi *browser* Android dan sebuah *embeddable web view*.

e. Scalable Graphics Library (SGL)

SGL mendasari mesin grafis 2D yang mengimplementasikan keseluruhan *graphics pipeline* dari Android.

f. 3D Libraries

Implementasi yang didasarkan pada OpenGL ES 1.0 APIs dimana *library* menggunakan baik *hardware 3D acceleration* (jika tersedia) ataupun yang disertakan, dengan *software 3D* yang sangat optimal.

g. FreeType Library

Digunakan untuk menghasilkan semua tulisan *bitmap* dan *vector*.

h. SQLite

Merupakan *relational database* yang kuat dan ringan serta tersedia untuk semua aplikasi.

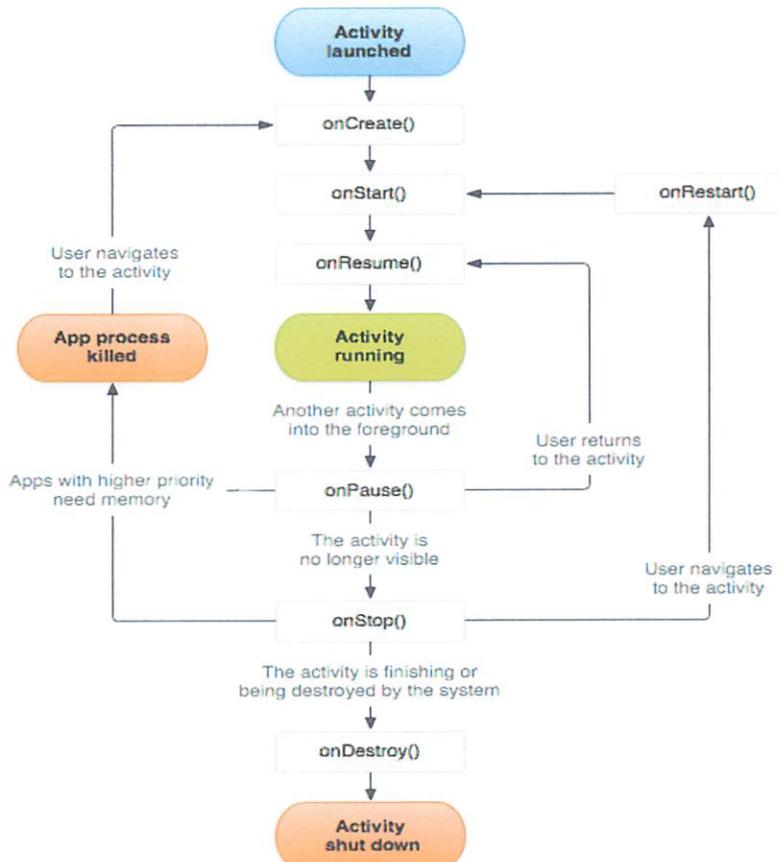
4. *Android Runtime*

Merupakan lokasi dimana komponen utama dari DVM ditempatkan.

5. *Linux Kernel*

Arsitektur Android berdasarkan pada *Linux Kernel* yang dapat digunakan untuk mengatur keamanan, manajemen memori, manajemen proses, *network stack*, dan *driver model*. Kernel juga bertindak sebagai lapisan abstrak antara *hardware* dan seluruh *software stack*.

2.3.3. Activity Lifecycle



Gambar 2.3. *Android Activity Lifecircle*

Activity dalam sistem dikenal sebagai *activity stack*. Berikut penjelasan masing-masing fase dalam Gambar 2.3.

1. OnCreate

Fase ini terjadi pada saat suatu *activity* dibuat. Pada bagian ini, tampilan yang berhubungan langsung dengan *user* dipanggil.

2. OnStart

Fase ini dipanggil tepat sebelum suatu *activity* terlihat di layar. Fase ini berakhir ketika *activity* yang dipanggil menjadi *foreground activity* di layar sehingga fase berpindah ke fase *onResume*.

3. OnResume

Pada fase ini, *activity* berjalan dan berinteraksi langsung dengan *user*. Pada fase ini, aplikasi menerima *input* dari *user* yang berasal dari *keyboard* dan *touch*.

4. OnPause

Fase ini dipanggil ketika aplikasi lain dalam satu sistem akan menuju ke fase *resume*. Pada fase ini, *activity* yang berjalan tidak memiliki akses ke layar.

5. OnStop

Fase ini dipanggil ketika *activity* tidak lagi terlihat karena *activity* lain sudah mengambil alih *foreground* atau karena *activity* kita sudah berakhir daur hidupnya.

6. OnRestart

Fase ini dipanggil ketika *activity* yang masih berjalan namun tidak terlihat pada layar kembali dijalankan, yang kemudian akan berpindah ke fase *onStart*.

7. OnDestroy

Fase ini merupakan fase berakhirnya suatu *activity*. Fase ini terjadi biasanya ketika aplikasi memanggil fungsi *finish()*.

2.3.4 . Versi Android

Android memiliki pembaharuan semenjak pertama kali diliris. Pembaharuan dilakukan untuk merperbaiki *bug* maupun penambahan fitur-fitur baru. Berikut merupakan versi-versi yang dimiliki Android:

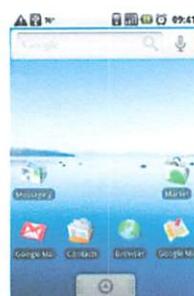
1. Android versi 1.0



Gambar 2.4. Android versi 1.0 pada HTC Dream

Dalam Gambar 2.4, ditunjukkan tampilan Android versi 1.0 pada HTC Dream. Dirilis pada tanggal 23 September 2008 dan menggunakan *Linux Kernel 2.6.25*. Adapun fitur-fitur pada Android versi 1.0, yakni akses *Google Account* langsung pada perangkat (*Google Contacts*, *Google Calendar*, *Google Maps*, *Google Sync*, *Google search*, *Google Talk*), *Android Market*, *Messaging*, *Media Player*, Konektifitas *Wi-Fi*, *Bluetooth*, dan lainnya.

2. Android versi 1.5 (*Cupcake*)



Gambar 2.5. Tampilan layar utama Android versi 1.5

Dalam Gambar 2.5, ditunjukkan tampilan layar utama Android versi 1.5. Versi ini dirilis pada tanggal 30 April 2009 dan menggunakan *Linux Kernel 2.6.27*. Adapun penambahan fitur dari versi sebelumnya, yakni:

- a. Merekam dan menonton video dengan modus kamera,
- b. *Upload* video ke *Youtube* dan gambar ke *Picasa*,
- c. Dukungan *Bluetooth Audio Distribution Profile* (A2DP) dan *Audio/Video Remote Control Profile* (AVRCP),
- d. Terhubung secara otomatis ke *headset Bluetooth* dalam jarak tertentu,
- e. *Widgets* dan *folder* yang dapat ditambahkan ke layar utama,
- f. Transisi animasi layar dan *keyboard* pada layar yang dapat disesuaikan dengan sistem.

3. Android versi 1.6 (*Donut*)



Gambar 2.6. Fitur *Gesture* pada Android versi 1.6

Dalam Gambar 2.6, ditunjukkan tampilan fitur *Gesture* untuk pengenalan objek dari garis atau gambar tangan pada layar. Android versi 1.6 dirilis pada tanggal 15 September 2009 dan menggunakan *Linux Kernel* 2.6.29. Pembaharuan fitur dari versi sebelumnya, yakni:

- a. Teknologi yang mendukung *Code Division Multiple Access/Evolution Data Only* (CDMA/EVDO), 802.1x, *Virtual Private Network* (VPN), *text-to-speech engine* serta kemampuan *dial contact*.
- b. Dukungan resolusi layar *Wide Video Graphics Array* (WVGA),
- c. Peralatan *development* untuk *gesture framework* dan *gesture builder*.

4. Android versi 2.0/2.1 (*Éclair*)



Gambar 2.7. *Live wallpaper* pada Android versi 2.1

Dalam Gambar 2.7, ditunjukkan tampilan fitur *live wallpaper* pada Android versi 2.1. Versi ini dirilis pada tanggal 26 Oktober 2009 dan menggunakan *Linux Kernel* 2.6.29. Pembaharuan fitur dari versi sebelumnya, yakni:

- a. Perubahan *user interface* dengan *browser* baru dan dukungan HTML5,
- b. *Microsoft exchange support*,
- c. Dukungan *flash* untuk kamera,
- d. *Digital zoom*,
- e. *Bluetooth* 2.1,
- f. *Live wallpaper*.
- g. Kelas *motionevent* untuk mendeteksi *event* yang digunakan dalam *multi touch*.

5. Android versi 2.2 (*Froyo*)



Gambar 2.8. Tampilan layar utama Android versi 2.2

Dalam gambar 2.8, ditunjukkan tampilan layar utama Android versi 2.2. Versi ini dirilis pada tanggal 20 Mei 2010 dan menggunakan *Linux Kernel* 2.6.32. Pembaharuan fitur dari versi sebelumnya, yakni:

- a. Perbaikan kecepatan aplikasi tambahan dalam implementasi *Just In Time (JIT)*,
 - b. Integrasi dari *JavaScript V8 Chrome* ke dalam aplikasi *browser*,
 - c. Fungsionalitas USB *tethering* dan *portable hostpot*,
 - d. Penambahan pilihan untuk menonaktifkan akses data pada jaringan ponsel,
 - e. Waktu peralihan yang singkat antara *autotext* dan kamus yang dimilikinya,
 - f. Pengiriman kontak melalui *bluetooth*,
 - g. Dukungan *password numeric* dan *alpha numeric*,
 - h. Dukungan untuk melakukan *upload file* pada aplikasi *browser*,
 - i. Dukungan Adobe Flash 10.1.
6. Android versi 2.3 (*Gingerbread*)



Gambar 2.9. Android versi 2.3

Dalam Gambar 2.9, ditunjukkan tampilan layar utama, fitur *edit text*, dan manajemen baterai dari *device* Android versi 2.3. Versi ini dirilis pada tanggal 6 Desember 2010 dan menggunakan *Linux Kernel* 2.6.35. Pembaharuan fitur dari versi sebelumnya, yakni:

- a. Mendukung layar dengan resolusi besar (WXGA),
- b. Peningkatan fungsi *copy-paste* yang mengijinkan *user* untuk menyeleksi suatu kata,

- c. Mendukung *Near Field Communication* (NFC),
- d. Penambahan efek suara seperti *reverb*, *equalization*, *headphone virtualization*, dan *bass boost*,
- e. Penambahan *Download Manager*, yang memberikan kemudahan pada *user* untuk mengakses *file* yang di-*download* dari *browser*, *e-mail*, dan aplikasi lainnya,
- f. Mendukung *webM/VP8 video playback*, dan *AAC audio encoding*,
- g. Peningkatan *power management* yang lebih efektif dalam mengelola aplikasi.
- h. Peralihan dari *Yet Another Flash File System* (YAFFS) ke *ext4* pada perangkat,
- i. Dukungan sensor lebih banyak (seperti *Gyroscopes* dan *Barometers*),
- j. Dukungan *voice* dan *video chat* menggunakan *Google Chat*.

7. Android versi 3.0 (*Honeycomb*)



Gambar 2.10 Tampilan layar utama Android versi 3.0 pada *tablet*

Dalam Gambar 2.10. ditunjukkan tampilan layar utama Android versi 3.0 pada *tablet*. Versi ini dirilis pada tanggal 22 Februari 2011 dan menggunakan *Linux Kernel* 2.6.36. Pembaharuan fitur dari versi sebelumnya, yakni:

- a. Dioptimasi untuk mendukung *tablet* dengan virtual baru dan *holographic user interface*,
- b. Penambahan *system bar*, fitur untuk mengakses *notifications*, dan tombol navigasi di bagian bawah layar,

- c. Peningkatan kecepatan akses untuk *camera exposure. Focus, flash, zoom, front-facing camera, time-lapse*, dan lainnya,
 - d. *Hardware acceleration*,
 - e. Mendukung *multi-core processors*,
 - f. Kemampuan untuk *encrypt* semua data *user*.
8. Android versi 4.0 (*Ice Cream Sandwich*)



Gambar 2.11. Tampilan layar utama dan menu pada Android versi 4.0

Dalam Gambar 2.11. ditunjukkan tampilan layar utama pada Android versi 2.2. Versi ini dirilis pada tanggal 19 Oktober 2011 dan menggunakan *Linux Kernel* 3.0.1. Pembaharuan fitur dari versi sebelumnya, yakni:

- a. Kemudahan untuk membuat *folder*, dengan *drag-and-drop*,
- b. Penambahan *visual voicemail* untuk kemampuan *voicemail* dengan cepat maupun lambat,
- c. Fungsi *pinch-to-zoom* pada kalender,
- d. *Offline search*,
- e. Integrasi *screenshot capture*,
- f. Penambahan *error correction* pada keyboard,
- g. *Face unlock*, fitur yang memungkinkan *user* untuk *unlock handset* menggunakan *facial recognition*,
- h. Penambahan aplikasi kamera dengan *zero shutter lag*, pengaturan *time lapse*, *panorama mode*, dan kemampuan untuk *zoom* ketika merekam,
- i. Penambahan *built-in photo editor*,

- j. Mendukung format gambar *WebP*,
- k. Memungkinkan perangkat *Wi-Fi* terhubung ke perangkat lainnya (*Wi-Fi Direct*),
- l. Mendukung resolusi QVGA.

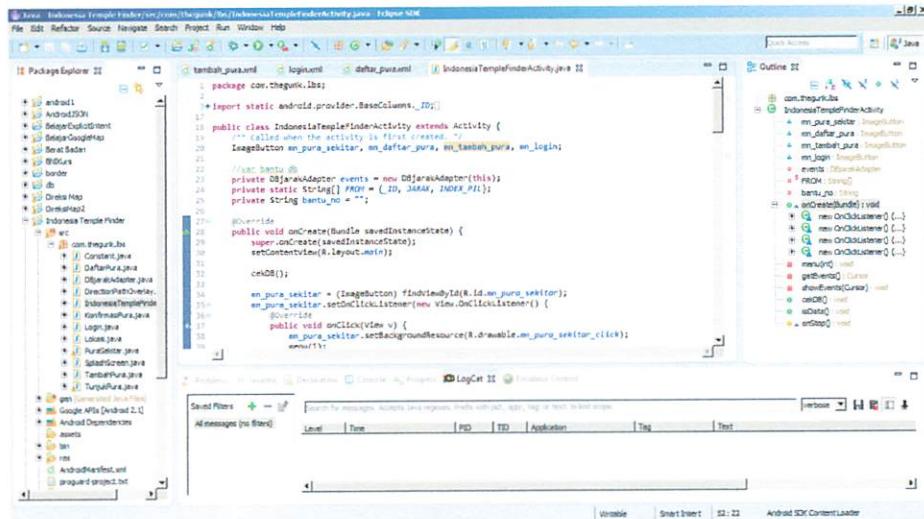
2.4 Eclipse^[3]

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent*). Berikut ini adalah sifat dari Eclipse:

1. **Multi-platform:** Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
2. **Mulit-language:** Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
3. **Multi-role:** Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.

Dalam Gambar 2.12. ditunjukkan tampilan lembar kerja Eclipse



Gambar 2.12. Tampilan lembar kerja Eclipse

2.5 Mikrotik^[11]

Mikrotik merupakan sistem operasi independen berbasiskan Linux yang difungsikan sebagai Router. Mikrotik didesain untuk memberikan kemudahan bagi penggunanya. Administrasinya bisa dilakukan melalui Windows application (WinBox). Selain itu instalasi dapat dilakukan pada Standard computer PC. PC yang akan dijadikan router mikrotikpun tidak memerlukan resource yang cukup besar untuk penggunaan standard, misalnya hanya sebagai gateway. Untuk keperluan beban yang besar (network yang kompleks, routing yang rumit). Mikrotik Router OS adalah sebuah sistem operasi berbasis text yang di peruntukkan pada computer network sebagai router jaringan. selain system mikrotik sendiri, OS ini dilengkapi dengan berbagai program tambahan seperti router, dhcp, ppp, wlan, hotspot, ipv6 dan banyak lagi, dimana keseluruhan program tersebut berfungsi dalam pengeturan jaringan komputer. dalam pembuatanya OS mokrotik dapat diinstal paa sebuah PC dengan spesifikasi minimum Pentium 2 dengan RAM minimum 64, adalah sesuatu yang sangat murah dalam membuatnya jika di bandingkan dengan kegunaannya pada jaringan komputer.

Secara umum / kebanyakan mikrotik digunakan sebagai pembatas akses secara langsung suatu hardware jaringan seperti modem, selain itu dengan adanya keberadaan sebuah router sirkulasi data pada jaringan akan lebih stabil dan cepat, karena router memiliki kemampuan membagi bandwith jaringan. jenis mikrotik ada 2 :

1. berbentuk software, jenis ini yang sering di pasang / di instal pada PC sebagai pengganti router Hardware
2. built-in Hardware mikrotik yang telah terpasang pada perangkat keras dan siap menerima setting dari administrator jaringan

Pada penerapannya orang selalu relatif dalam memilih jenis mikrotik yang ingin digunakan, banyak dari mereka yang memilih built-in hardware karena tidak perlu repot dengan pemasangan /instalasi-nya, namun tidak sedikit yang menggunakan software dan memasangnya sendiri pada sebuah PC, alasannya karena jika terjadi kerusakan mereka dapat memperbaikinya sendiri, tidak seperti yang built-in, saat rusak kita harus melirik tempat service yang mengerti tentang Built-in mikrotik.

2.6 Winbox Loader^[5]

Winbox Loader adalah Aplikasi yang berjalan pada PC dimana tugas utama winbox adalah untuk menyetting atau mengatur mikrotik dengan GUI, atau tampilan desktop. Winbox bisa mendeteksi mikrotik yang sudah diinstal asal masih dalam satu network yaitu dengan mendeteksi mac address dari ethernet yang terpasang di mikrotik.

Deskripsi tombol dari bidang layar winbox loader.

1. Menemukan dan menunjukkan MNDP (Mikrotik Neighbor Discovery Protocol) atau CDP (Cisco Discovery Protocol) perangkat.
2. Connect – hubungan ke router
3. Save – Menyimpan Alamat login, password dan catatan entri yang disimpan akan ditampilkan pada bagian bawah jendela loader.
4. Remove – menghapus enti yang dipilih dari daftar yang disimpan.
5. Tools – memungkinkan untuk menjalankan berbagai perangkat, menhapus semua item dari daftar, membersihkan cache pada disk lokal, alamat impor dari file WBX atau ekspor mereka untuk WBX file.
6. Connect to – tujuan IP atau MAC address dari router.
7. Login – username yang digunakan untuk otentikasi.

BAB III

PERANCANGAN SISTEM

Bab III ini akan diuraikan mengenai perancangan sistem dari aplikasi Gui Mikrotik. Dimana terdapat tahapan analisa dan perancangan umum, perancangan sistem dan lain sebagainya.

3.1 Analisa Masalah

Winbox merupakan salah satu aplikasi yang terus dikembangkan oleh para user di seluruh dunia karena aplikasi ini berbasis *OpenSource*. Hal ini juga diterapkan dalam pemrograman android oleh penulis, karena dengan penerapan pada system android banyak kelebihan yang dapat diambil dari manfaatnya seperti penggunaan yang lebih mudah dan lainnya. Oleh karena itu pilihan akan *Winbox berbasis android* merupakan salah satu kontribusi yang tepat pada proses konfigurasi jaringan.

3.2 Analisa Kebutuhan

Pada sub bab ini akan dibahas mengenai analisa kebutuhan dalam membangun sistem.

3.2.1 Analisa Perangkat Keras (*Hardware*)

Dalam membuat aplikasi winbox berbasis android diperlukan *Hardware* pendukung, diantaranya:

1. Notebook

Spesifikasi yang digunakan:

- a. Toshiba satellite U40D
- b. Windows 7 32-bit
- c. AMD Turion™ X2 Dual-Core Mobile RM-70 2.00 GHz
- d. 2GB RAM
- e. 250 GB HDD
- f. ATI Radeon 3100 Graphics

2. Perangakat Android

Dalam Tabel 3.1 ditunjukkan spesifikasi Samsung Galaxy Ace Duos

Tabel 3.1 Spesifikasi Samsung Galaxy Ace Duos

<i>General</i>	<i>2G Network</i>	GSM 850 / 900 / 1800 / 1900 - SIM 1 & SIM 2
	<i>3G Network</i>	HSDPA 900 / 2100
<i>Display</i>	<i>Type</i>	TFT capacitive touchscreen, 256K colors
	<i>Size</i>	320 x 480 pixels, 3.5 inches (~165 ppi pixel density)
<i>Memory</i>	<i>Internal</i>	3 GB storage, 512 MB
	<i>Eksternal</i>	microSD, up to 32 GB
<i>Data</i>	<i>GPRS</i>	85.6 kbps
	<i>EDGE</i>	236.8 kbps
	<i>Speed</i>	HSDPA, 7.2 Mbps
	<i>WLAN</i>	Wi-Fi 802.11 b/g/n, Wi-Fi hotspot
	<i>USB</i>	Yes, microUSB v2.0
<i>Features</i>	<i>OS</i>	Android OS, v2.3 (Gingerbread)
	<i>Chipset</i>	Qualcomm MSM7227
	<i>CPU</i>	832 MHz
	<i>GPS</i>	Yes, with A-GPS support
	<i>Java</i>	Yes, via Java MIDP emulator
<i>Battery</i>	<i>Type</i>	Li-Ion 1300 mAh battery
	<i>Stand-by</i>	Up to 570 h (2G) / Up to 420 h (3G)

3.2.2 Analisa Perangkat Lunak (*Software*)

Selain *hardware*, dalam pembuatan aplikasi diperlukan beberapa *software* pendukung, yaitu:

1. Sistem Operasi Windows 7 Ultimate 32-bit
2. Eclipse
3. JDK 6 Windows i586

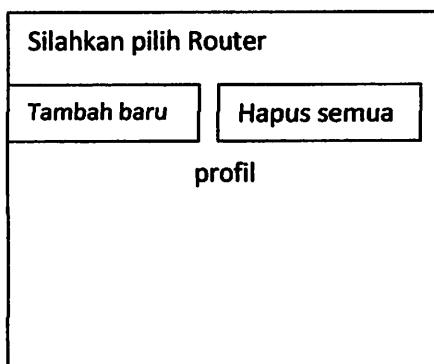
3.3 Spesifikasi Aplikasi

Sesuai dengan tujuan pembuatan aplikasi, spesifikasi aplikasi ini adalah sebagai berikut:

1. Aplikasi dibuat dengan menggunakan bahasa pemrograman *Java* dan *XML*,
2. *IDE* yang digunakan adalah *Eclipse*,
3. Aplikasi berjalan di perangkat Android versi 2.3.
4. Aplikasi ini membutuhkan koneksi internet.

3.4 Perancangan Aplikasi

3.4.1 Perancangan Form Aplikasi



Gambar 3.1. Rancangan tampilan Awal

Dalam gambar 3.1. menunjukkan rancangan tampilan awal dengan beberapa tombol menu.

Silahkan masukan data login	
IP	port
username	
password	

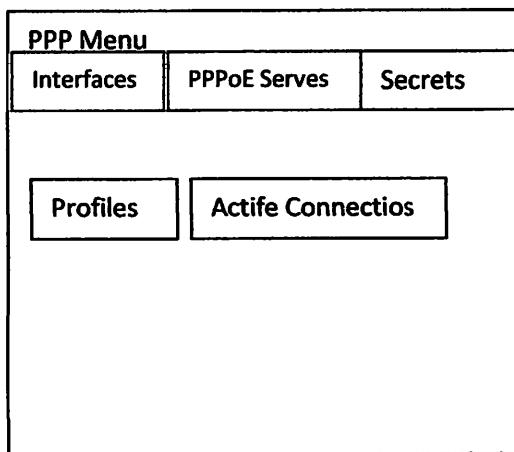
Gambar 3.2. Rancangan tampilan login

Dalam gambar 3.2. menunjukkan rancangan tampilan login untuk *admin*. Terdapat *textbox* untuk memasukkan *IP*, *port*, *username* dan *password*.

Interfaces	
aktif	Alamat IP

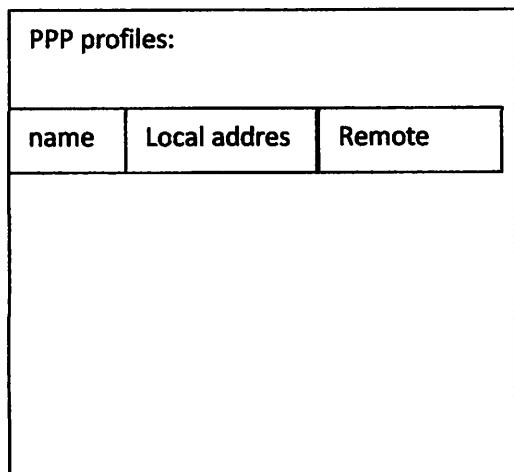
gambar 3.3. Rancangan tampilan Interfaces

Dalam gambar 3.3. menunjukkan rancangan tampilan interfaces dimana terdapat interfaces yang aktif dan non aktif serta alamat ip.



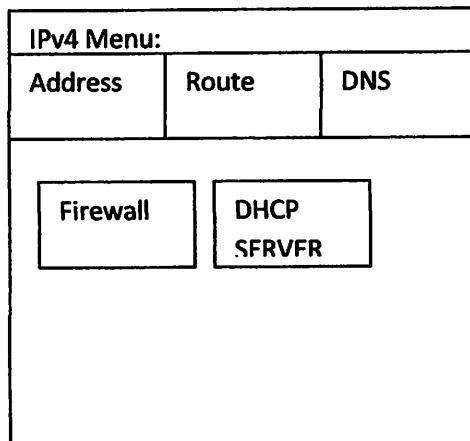
Gambar 3.4. rancangan menu PPP

Dalam gambar 3.4. menunjukkan rancangan dari menu PPP



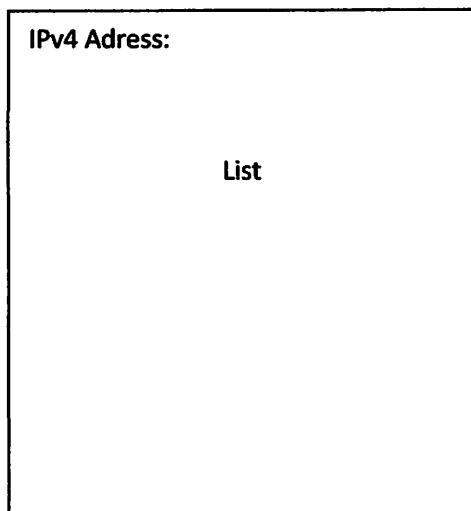
Gambar 3.5. rancangan PPP profiles

Dalam gambar 3.5. menunjukkan rancangan profil PPP dimana terdapat name.local address dan remote address



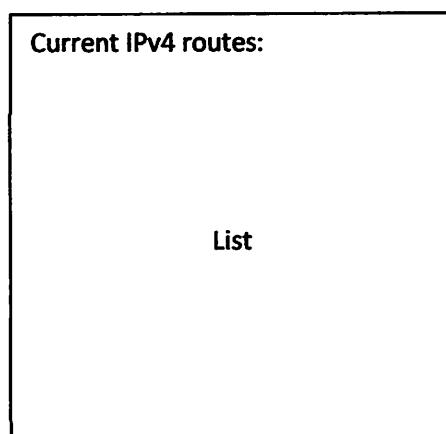
Gambar 3.6. rancangan tampilan IPV4 menu

Dalam gambar 3.6. menunjukkan rancangan tampilan IPv4 dimana terdapat Address, Route, DNS, Firewall, dan DHCP server.



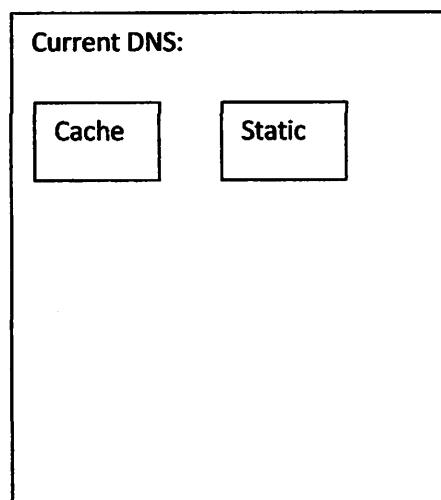
Gambar 3.7. rancangan tampilan IPv4 adres

Dalam gambar 3.7 menunjukkan rancangan tampilan address IPv4 dimana terdapat daftar dari IPv4 address



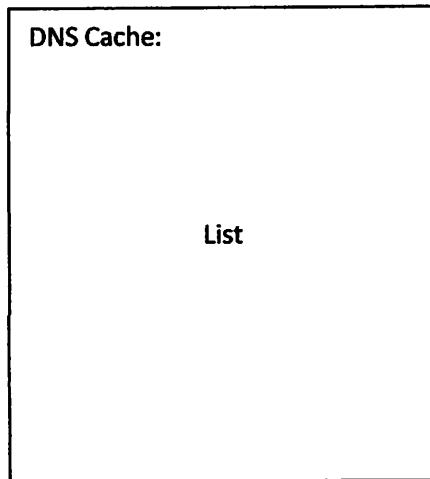
Gambar 3.8. rancangan tampilan current IPv4 router

Dalam gambar 3.8. menunjukkan rancangan tampilan current IPv4 router.



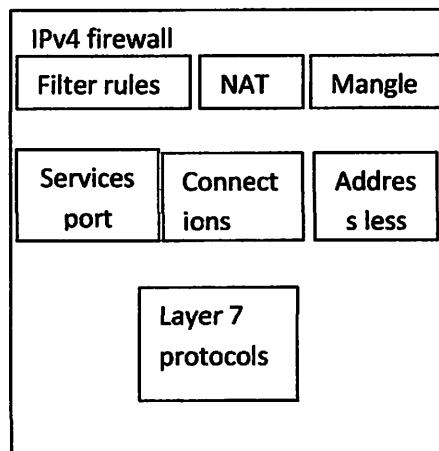
Gambar 3.9. rancangan tampilan current DNS

Dalam gambar 3.9. menunjukkan rancangan tampilan current DNS, dimana terdapat dua tombol Cache dan Static.



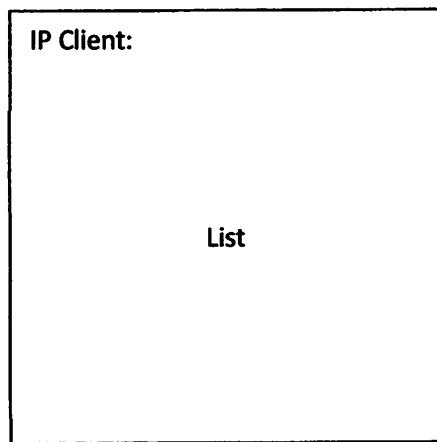
Gambar 3.10. rancangan tampilan DNS cache

Dalam gambar 3.10 menunjukkan rancangan DNS cache



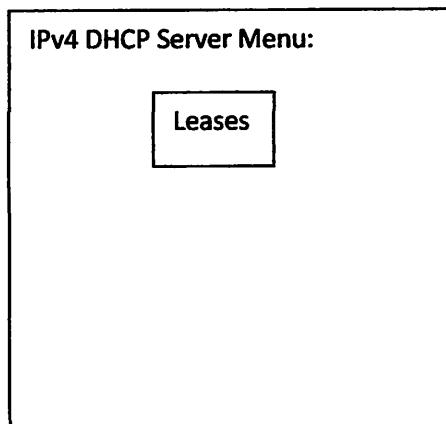
Gambar 3.11. rancangan tampilan IPv4 firewall

Dalam gambar 3.11. menunjukkan rancangan tampilan IPv4 firewall, dimana terdapat filter rules, NAT, mangle, services port, connections, address less, dan layer7 protocols.



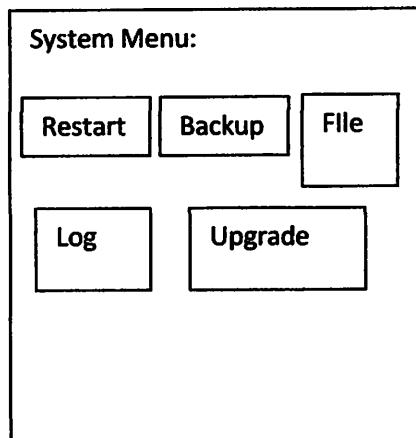
Gambar 3.12. rancangan tampilan IP Client

Dalam gambar 3.12. menunjukkan rancangan tampilan daftar IP client.



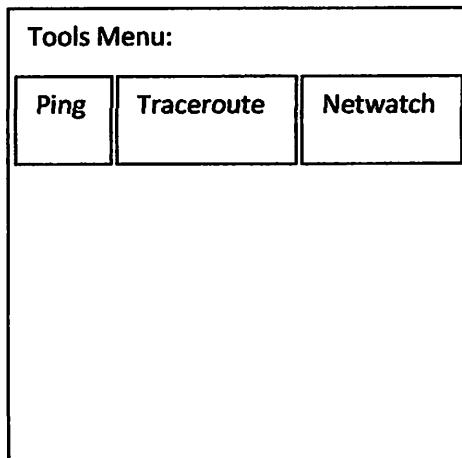
Gambar 3.13. rancangan tampilan IPv4DHCP Server Menu

Dalam gambar 3.13. menunjukkan rancangan tampilan IPv4DHCP Server Menu.



Gambar 3.14. rancangan tampilan System menu

Dalam gambar 3.14. menunjukkan rancangan tampilan System Menu, dimana terdapat Restart, Backup, File, Log, Dan Upgrade.

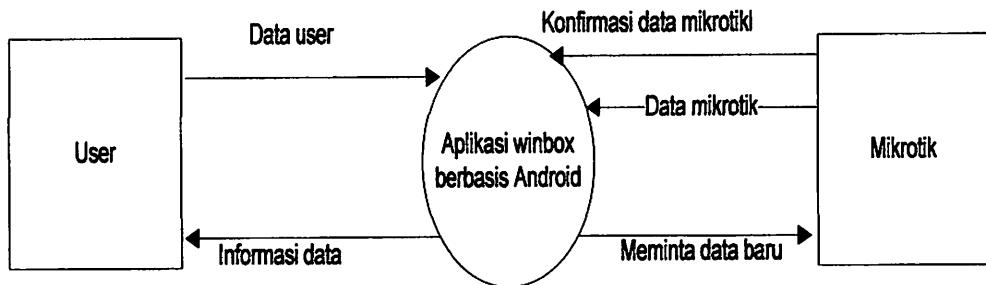


Gambar 3.15. rancangan tampilan Tools Menu

Dalam gambar 3.15. menunjukkan rancangan tampilan system menu, dimana terdapat Ping, Traceroute, dan Netwatch.

3.5 Contex Diagram

Context diagram merupakan diagram pertama dalam rangkaian suatu *Data Flow Diagram* (DFD) yang menggambarkan entitas-entitas yang berhubungan dengan suatu sistem.

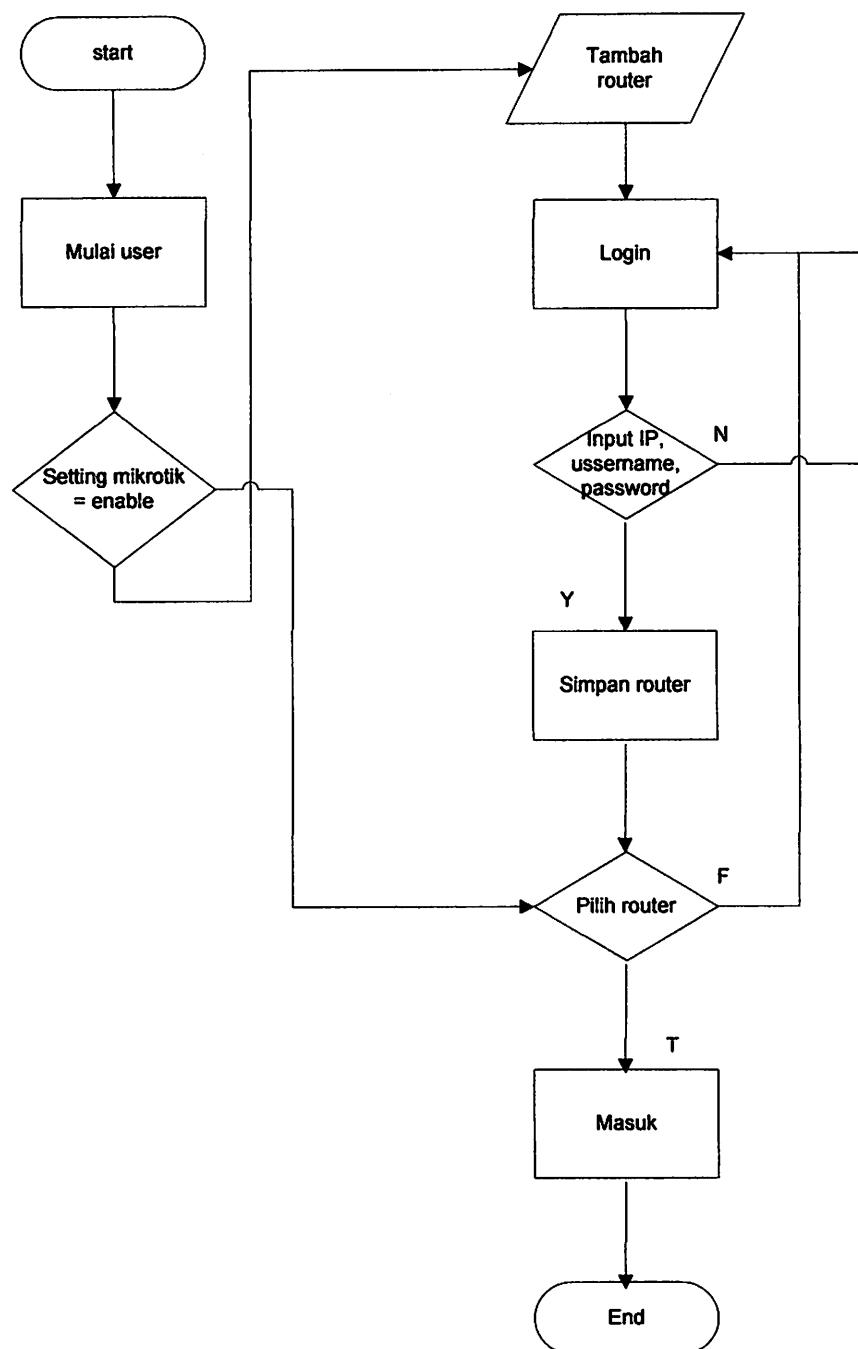


Gambar 3.16. Diagram Contex

Dalam Contex diagram gambar 3.16. terdapat dua entitas yang berkaitan dengan sistem, yaitu *user* yang berinteraksi dan menjalankan sistem untuk melakukan penambahan data ke aplikasi winbox, kemudian aplikasi winbox meminta data baru ke mikrotik dan mendapatkan informasi mikrotik. Entitas yang kedua *mikrotik* mengkonfirmasi data baru yang diminta aplikasi winbox dan memberikan informasi data ke *user*.

3.6 Perancangan Sistem

Flowchart



Gambar 3.17 Flowchart Sistem

Penjelasan dari alur sistem flowchart adalah sebagai berikut:

1. User mulai
2. User menyeting mikrotik dengan mengenablekan.
3. User menambahkan router
4. User login menginputkan IP, username, dan password mikrotik
5. Simpan router
6. Apabila mikrotik telah enable,pilih router yang telah ditambahkan
7. Apabila karakter yang diinputkan valid pada penambahan router selanjutnya data akan terbaca.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

Dalam Bab IV akan dijelaskan tentang implementasi dan pengujian terhadap kinerja aplikasi yang telah dibuat. Dengan demikian akan diketahui tingkat kekurangan dan tingkat kelebihan dari sistem atau aplikasi yang telah dibuat dalam program aplikasi ini.

4.1 Implementasi Sistem

Tahap implementasi pengembangan aplikasi merupakan proses pengubahan spesifikasi sistem menjadi sistem yang dapat dijalankan. Tahap ini merupakan lanjutan dari proses perancangan, yaitu proses pemrograman perangkat lunak sesuai dengan spesifikasi dan desain sistem. Dalam hal ini dapat dilihat cara kerja program lebih detail tentang fitur yang disediakan.

4.2 Hasil Implementasi

Tampilan aplikasi Winbox berbasis android meliputi tampilan *menu utama*, tampilan *login*, tampilan *main menu*, tampilan interfaces tampilan *PPP menu*, tampilan *IP v4 menu*, tampilan *IP v4 DHCP menu*, tampilan *system menu*, dan tampilan *Tools menu*.

4.2.1 Tampilan *Icon* Aplikasi

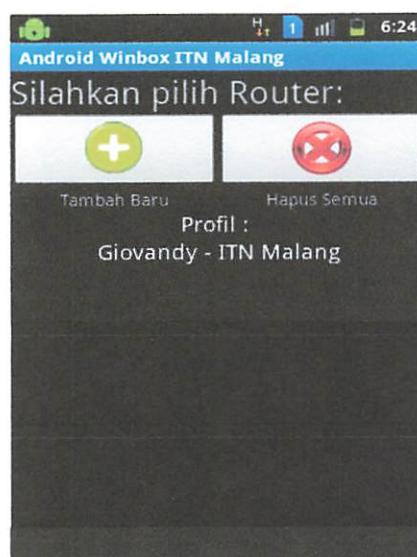
Saat aplikasi ter-*install* di perangkat Android, *icon* aplikasi akan muncul di menu utama Android seperti yang ditunjukkan dalam Gambar 4.1



Gambar 4.1. Tampilan *icon* pada menu utama

4.2.2 Tampilan Menu Utama Aplikasi

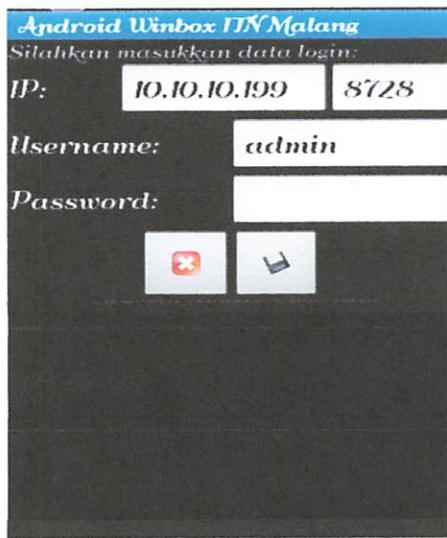
Setelah masuk ke aplikasi akan ditampilkan menu utama dari aplikasi, terdapat menu Add RB untuk menambahkan router mikrotik yang akan di remoute, Setting untuk mengatur font size, Import WBX untuk mengambil dan membuka (import) daftar isian alamat dari file wbx atau menyimpannya (export) dalam file wbx, dan Clear All untuk menghapus semua isian di daftar bagian bawah winbox. Seperti yang ditunjukkan dalam Gambar 4.2



Gambar 4.2. *Tampilan Menu Utama Aplikasi*

4.2.3 Tampilan Login

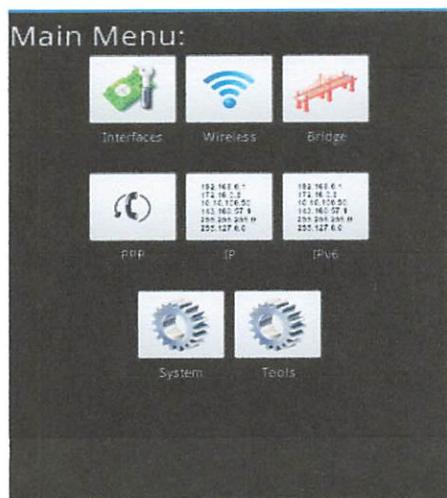
Menu *Login* akan tampil jika kita akan menambahkan router mikrotik dengan menekan menu Add RB. Digunakan user untuk *login* agar mendapat hak akses masuk ke mikrotik. *User* diminta harus melengkapi *IP*, *Username* dan *Password* yang *valid*. Tampilan *Login* ditunjukkan dalam Gambar 4.3



Gambar 4.3. Tampilan Login

4.2.4 Tampilan Main Menu

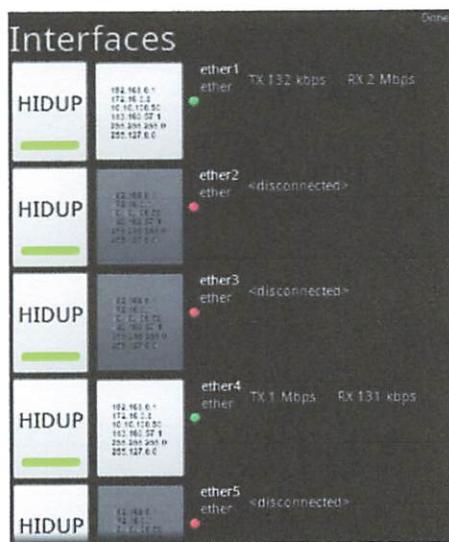
Menu main menu akan tampil jika user telah melakukan login yang valid untuk mendapatkan hak akses dari router mikrotik dan pada main menu tersebut terdiri beberapa fitur menu seperti: Interface, Bridge, PPP, IP, IPv6, System, Tools. Dalam Gambar 4.4 ditunjukkan tampilan dari main menu.



Gambar 4.4. Tampilan Main Menu

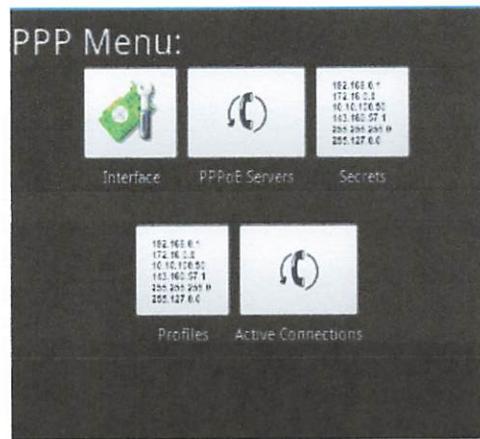
1. Interfaces, merupakan gerbang trafik keluar atau masuk ke mikrotik. Secara default mikrotik hanya mengenali interfaces yang secara fisik memang ada. Kita dapat merubah nama interfaces tersebut dengan tujuan untuk

memudahkan dalam mengidentifikasi fungsi. Interfaces dapat dilihat pada gambar 4.5.



Gambar 4.5. Tampilan Interfaces

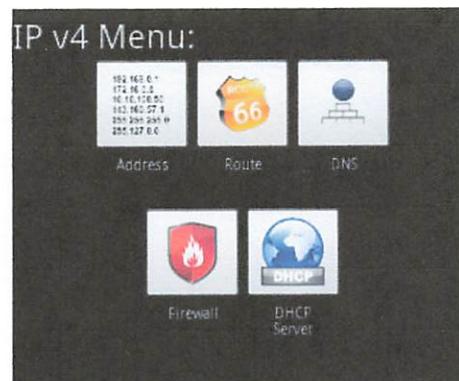
2. Wireless, menampilkan jaringan yg terhubung.
3. Bridge, menghubungkan 2 Plant Network terpisah jarak dan waktu secara Point to Point
4. PPP (Point-to-Point Protocol), merupakan paket yang memuat protokol PPP. Paket ini diperlukan untuk fitur komunikasi serial dengan menggunakan PPP, ISDN PPP, L2TP, dan PPTP serta komunikasi PPP on Ethernet (PPPoE). Paket PPP digunakan untuk komunikasi *Wide Area Network* dengan menggunakan komunikasi serial mode *asynchronous* maupun mode *synchronous*. PPP dapat dilihat pada gambar 4.6.



Gambar 4.6 Tampilan PPP

5. IP, Menu IP adalah menu utama dengan berbagai pilihan yang berhubungan dengan konfigurasi Internet Protocol. Dalam mengkonfigurasi manajemen bandwidth sub menu yang sering digunakan yaitu address, route, firewall, DNS, DHCP Server.

Tampilan menu IP seperti pada gambar 4.7.



Gambar 4.7. Tampilan IP

- a. *Sub menu Address*, Sub menu ini adalah bagian utama yang digunakan untuk membuat router bekerja. Mikrotik saat ini hanya mendukung ipv4 dengan subnet mask. Mikrotik dapat menggunakan alamat Ip secara static atau pun dynamic. Sub menu address dapat dilihat pada gambar 4.8.

IPv4 Addresses:		
hotspot network		
192.168.1.3/24	ether2	192.168.1.0
DWHD		
192.168.16.13/27	ether4	192.168.10.0
lant		
192.168.30.1/24	ether3	192.168.30.0

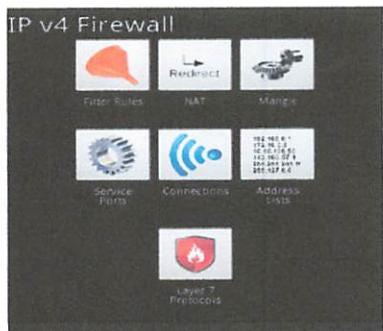
Gambar 4.8. Tampilan Sub Menu Address

- b. *Sub menu route*, Sub menu ini menampilkan kondisi table routing baik aktif maupun yang cadangan. Daftar routing ini bersifat permanen (read only), static dan dynamic. Tampilan route dapat dilihat pada gambar 4.9.

Current IPv4 Routes:			
AD	dst-address	gateway	gateway-status
AD	192.168.30.0/24	ether3	ether3 unreachable
AD	192.168.10.0/27	ether4	ether4 reachable
AD	192.168.1.0/24	ether2/ether1	ether2 unreachable ether1 re
ADS	6.0.0.6/0	192.168.1.1	192.168.1.1 reachable ether

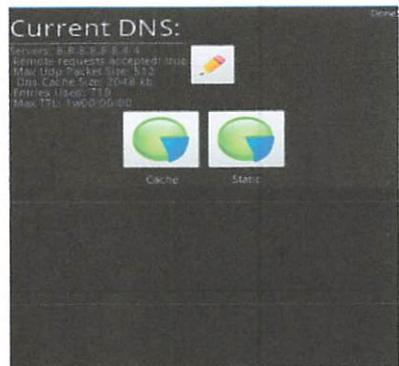
Gambar 4.9. Tampilan Sub Menu Route

- c. Sub menu firewall, Sub Menu *Firewall* ini berisi konfigurasi packet filter dan fitur mengatur fungsi keamanan untuk mengatur arus data dari dan ke router. Fungsi *Network Address Translation* juga merupakan *tools* yang termasuk digunakan untuk pembatasan *access* secara langsung dan melindungi *traffic* yang akan keluar dari router. Tampilan menu firewall seperti pada gambar 4.10.



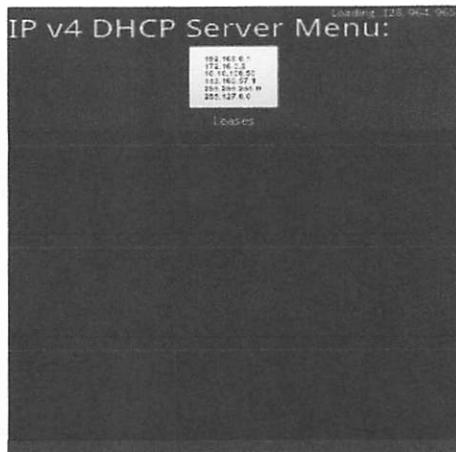
Gambar 4.10. Tampilan Sub Menu Firewall

- d. Sub menu DNS, Sub menu ini digunakan untuk mengurangi trafik DNS ke internet dan mempercepat waktu yang reselove dapat digunakan fungsi DNS cache. Mikrotik DNS cache dapat menggunakan dns server primary dan secondary. Menu DNS dapat dilihat pada gambar 4.11.



Gambar 4.11. Tampilan Sub Menu DNS

- e. Sub menu DHCP server, Tempat menyeting DHCP Server. Jika kita menginginkan tiap komputer dalam jaringan mendapatkan IP, DNS dan Gateway secara otomatis. Tampilan DHCP server dapat dilihat pada gambar 4.12.



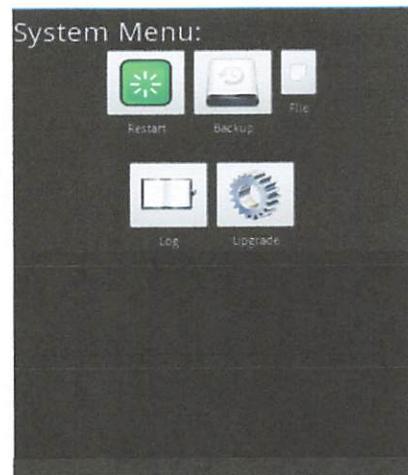
Gambar 4.12. Tampilan Sub Menu DHCP Server

- f. Pada menu IP kita bisa Memantau IP client yang mengakses jaringan yang terhubung dengan router mikrotik seperti pada Gambar 4.13

04:39:E5 70:E1:71 192.168.10.34 Active:04:39:E5 70:E1:71 192.168.10.34 0:04:39:e5:70:e1:71	00:19:E0:83:10:E7 192.168.10.25 Active:00:19:E0:83:10:E7 192.168.10.25 1:0:19:e0:83:10:e7	0C:EE:E6:98:7C:84 192.168.10.18 Active:0C:EE:E6:98:7C:84 192.168.10.18 0:0:c:ee:e6:98:7c:84	00:48:7A:88:41:33 192.168.10.3 Active:00:48:7A:88:41:33 192.168.10.3 1:0:48:7a:88:41:33	44:6D:52:09:30:1F 192.168.10.23 Active:44:6D:52:09:30:1F 192.168.10.23 1:44:6d:52:09:30:1f	00:30:18:A5:27:20 192.168.10.27 Active:00:30:18:A5:27:20 192.168.10.27 1:0:30:18:a5:27:20	00:DF:9A:17:99:5B 192.168.10.30 Active:00:DF:9A:17:99:5B 192.168.10.30 1:0:df:9a:17:99:5b	18:D1:11:11:11:64 192.168.10.29 Active:18:D1:11:11:11:64 192.168.10.29 1:18:d1:11:11:11:64	1C:48:D8:87:EC:92 192.168.10.12 Active:1C:48:D8:87:EC:92 192.168.10.12 1:1c:48:d8:87:ec:92
--	---	---	---	--	---	---	--	--

Gambar 4.13. Tampilan IP Client

- g. Sistem menu,ada pun tools dalam sistem menu digunakan untuk mengatur mikrotik tersebut seperti restart, backup file, log, dan upgrade. Tampilan system menu seperti pada gambar 4.14.



Gambar 4.14. *System Menu*

h. Tools Menu, Menu ini digunakan untuk menguji jaringan maupun merekam kondisi suatu jaringan. Tampilan menu tools dapat dilihat pada gambar 4.15.



Gambar 4.15. *Tampilan Tools Menu*

4.3 Pengujian

Pada bagian ini dilakukan pengujian pada aplikasi Winbox berbasis Android. Pengujian yang dilakukan adalah pengujian fungsional sistem.

4.3.1 Pengujian fungsional

Pengujian fungsional sistem yang dilakukan terdiri dari beberapa pengujian. Penjelasan dari setiap pengujian sistem dapat dilihat dalam Tabel 4.1

Tabel 4.1. Menunjukkan jika apliaksi winbox berbasis Android bisa berjalan Pada Platform Android tersebut.

Fungsi	Hasil	Keterangan
Menampilkan IP client yang mengakses jaringan	<i>User</i> dapat melihat IP client yang sedang mengakses jaringan	Sesuai
Menampilkan Interfaces	<i>User</i> dapat melihat Interfaces dari jaringan yang terhubung	Sesuai
Menampilkan DNS Cache	<i>User</i> dapat melihat DNS dari Client	Sesuai

4.3.2 Tabel Pengujian Pada User

Dari hasil uji pada 9 user yang menggunakan aplikasi winbox berbasis android, Di dapatkan hasil 70% mengatakan menampilkan IP client mudah, 10% mengatakan menampilkan IP client cukup, dan 20% mengatakan menampilkan IP client Baik. Menampilkan interfaces 70% mengatakan mudah, 20% mengatakan menampilkan interfaces baik dan 10% mengatakan menampilkan interfaces cukup. Menampilkan DNS chace 70% mengatakan mudah, 20% mengatakan baik, dan 10% mengatakan cukup

Tabel 4.2 Hasil Pengujian aplikasi winbox berbasis android pada pengguna

No	Pertanyaan	Baik		Cukup		Mudah	
		Jumlah	%	Jumlah	%	Jumlah	%
1	Menampilkan IP client	3 Orang	20	2 Orang	10	4 Orang	70
2	Menampilkan Interfaces	3 Orang	20	2 Orang	10	4 Orang	70
3	Menampilkan DNS chace	3 Orang	20	2 Orang	10	4 Orang	70

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil-hasil pengujian fungsional yang telah dilakukan, aplikasi Winbox berbasis Android memiliki karakteristik-karakteristik sebagai berikut :

1. Aplikasi winbox berbasis android ini memungkinkan orang-orang yang ingin mengkonfigurasi mikrotik dapat melakukannya melalui *smartphone* Android 2.3 Gingerbread.
2. Fitur yang disediakan dalam aplikasi winbox berbasis android ini seperti menampilkan IP, menampilkan Interfaces memudahkan serta membantu *user* saat mengontrol jaringan mikrotik.
3. Aplikasi ini memungkinkan penggunaanya mengontrol jaringan tertentu tanpa harus menggunakan PC lg.

5.2 Saran

Untuk pengembangan kedepannya yang lebih baik untuk aplikasi winbox berbasis android ini, ada beberapa saran yang dapat digunakan, yaitu:

1. Dalam pengaplikasian aplikasi winbox berbasis android ini hanya sebatas menampilkan trafik jaringan, untuk kedepannya bisa dikembangkan dengan fungsi winbox pada dasarnya seperti mengkonfigurasi mikrotik.
2. Kedepannya untuk penggunaan aplikasi ini oleh user bisa lebih dikembangkan dengan penambahan user dalam mengontrol jaringan.

DAFTAR PUSTAKA

- [1].Herlambang, Moch Linto. Asis Catur L. 2008. Panduan Lengkap Menguasai Router Masa Depan Menggunakan Mikrotik RouterOS TM. Yogyakarta
- [2].Meier, Reto, 2009. “*Professional Android Application Development*”, Indianapolis, Indiana, Wiley Publishing, Inc.
- [3].Nugroho, Adi, 2008. Pemrograman Java Menggunakan IDE ECLIPSE CALLISTO. Salatiga.
- [4].Pratama, Widianto, 2011. Pengenalan Android, Depok
- [5].Pratamacomputer.com, 2009. Petunjuk Penggunaan Winbox, Pontianak.
- [6].Rogers, Ricks. Lombardo, John. Mednieks, Zigurd, & Blake Meike, 2009. “*Android Application Development*”, United States of America, O'Reilly Media, Inc.
- [7].Wardhana, Asoka. 2006. *MODUL BASIC MIKROTIK ROUTERS OS*. Jakarta
- [8].[Http://belajar-komputer-mu.com/pengertian-pemrograman-java-kelebihan-dan-kekurangan/](http://belajar-komputer-mu.com/pengertian-pemrograman-java-kelebihan-dan-kekurangan/)<http://id.m.wikipedia.org/wiki/java>
- [9].<http://communities.softwareag.com/ecosystem/communities/public/Developer/webmethods/products/tamino/faq/XMLStarter/XMLBasics.html>
- [10].<http://www.eclipse.org/downloads/>

LAMPIRAN



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Karanglo Km. 2 Malang

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama : Gio Vandy
NIM : 08.18.095
Masa Bimbingan : 17 Oktober 2012 – 17 April 2013
Judul Skripsi : **Rancang Bangun Aplikasi GUI Mikrotik Berbasis Android**

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Sabtu
Tanggal : 20 Februari 2013
Nilai : 70,00 (B)

Panitia Ujian Skripsi

Ketua Majelis Penguji

Joseph Dedy Irawan, ST, MT
NIP. 19740416 2005011002

Anggota Penguji

Penguji I

Ali Mahmudi, BEng, PhD
NIP.P. 1031000429

Penguji II

Yosep Agus Pranoto, ST
NIP.P. 1031000432



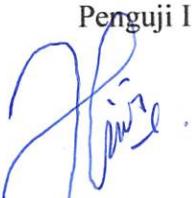
FORMULIR PERBAIKAN SKRIPSI

Nama : Gio Vandy
NIM : 08.18.095
Masa Bimbingan : 17 Oktober 2012 – 17 April 2013
Judul Skripsi : Rancang Bangun Aplikasi GUI Mikrotik Berbasis Android

TANGGAL	PENGUJI	URAIAN	PARAF
20 Februari 2013	I	<ol style="list-style-type: none">Perbaiki daftar isi, gambar, tabel, lampiranPerbaiki sistematika penulisanPerlu demo pakai mikrotikBagian 3.2 tidak pakai kebutuhan non fungsionalSurat pernyataan bermaterai	
	II	<ol style="list-style-type: none">Perbaiki daftar isi, gambar, lampiran.Perancangan aplikasi mengenai setting eclipse tidak usah ditampilkanDemo ulang pake mikrotikNama tabel diatasTabel pengujian sertakan kusioner	

Anggota Penguji

Penguji I



Ali Mahmudi, BEng, PhD
NIP.P. 1031000429

Penguji II

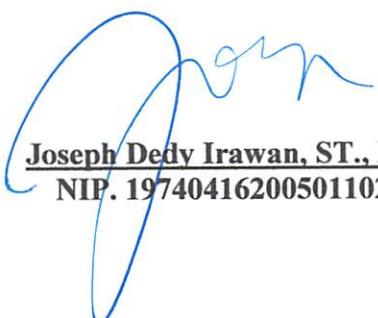


Yosep Agus Pranoto, ST
NIP.P. 1031000432

Mengetahui,

Dosen Pembimbing I

Dosen Pembimbing II



Joseph Dedy Irawan, ST., MT.
NIP. 197404162005011022



Sonny Prasetyo, ST., MT.
NIP. P. 1031000433



FORMULIR BIMBINGAN SKRIPSI

Nama : Gio vandy
Nim : 08.18.095
Masa Bimbingan : 17 Oktober 2013 – 17 April 2013
Judul Skripsi : Rancang Bangun Aplikasi GUI Mikrotik Berbasis Android

No	Tanggal	Uraian	Paraf Pembimbing
1	7-Januari-2013	Konsultasi Rancang Bangun Aplikasi GUI Mikrotik Berbasis Android	J
2	13-Januari-2013	Konsultasi bab 1, bab 2.	R
3	17-Januari-2013	Konsultasi bab 3.	J
4	3-Februari-2013	Konsultasi Makalah Seminar Hasil	A
5	4-Februari-2013	Hasil Revisi serta ACC Makalah Seminar Hasil	O
6	10-Agustus-2013	Konsultasi Laporan Ujian Skripsi	A
7	19-Februari-2013	ACC Laporan Ujian Skripsi	A
8			
9			
10			

Malang, 20 September 2013

Dosen Pembimbing I

Joseph Dedy Irawan, ST, MT.
NIP. 197404162005011022



FORMULIR BIMBINGAN SKRIPSI

Nama : Gio vandy
Nim : 08.18.095
Masa Bimbingan : 17 Oktober 2013 – 17 April 2013
Judul Skripsi : Rancang Bangun Aplikasi GUI Mikrotik Berbasis Android.

No	Tanggal	Uraian	Paraf Pembimbing
1	9-Januari-2013	Konsultasi BAB I, BAB II, dan BAB III	
2	14-Januari-2013	ACC BAB I, BAB II, dan BAB III	
3	17 Januari - 2013	Revisi Makalah Semhas	
4	28-Februari-2013	ACC Makalah Semhas	
5	28-Januari	Revisi BAB IV dan BAB IV	
6	29-Januari-2013	Revisi Buku Kompre	
7	19-Februari-2013	ACC Kompre	
8			
9			
10			

Malang, 20 September 2013
Dosen Pembimbing II

Sonny Prasetyo, ST, MT.
NIP.P. 1031000433

Form S-4b



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG

INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI

FAKULTAS TEKNIK SIPIL DAN PERENCANAAN

PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145

Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Nomor : ITN-107/T.INF/TA/2012

17 Oktober

2012

Lampiran :

Perihal : Bimbingan Skripsi

Kepada : Yth. Sdr. Joseph Dedy Irawan, ST, MT
Dosen Pembimbing Program Studi Teknik Informatika S1
Institut Teknologi Nasional
M a l a n g

Dengan hormat .

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

Nama : GIO VANDY
Nim : 0818095
Prodi : Teknik Informatika S1
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu 6 (enam) bulan, terhitung mulai tanggal ;

17 Oktober 2012 – 17 April 2013

Sebagai satu syarat untuk menempuh Ujian Sarjana Teknik, Program Studi Teknik Informatika S1.

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui
Program Studi Teknik Informatika S1
Ketua,

Joseph Dedy Irawan, ST, MT
NIP. 497404162005021002

Form S-4a



PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Nomor : ITN-107/T.INF/TA/2012
2012
Lampiran : -
Perihal : Bimbingan Skripsi

17 Oktober

Kepada : Yth. Sdr. Sonny Prasetyo, ST, MT.
Dosen Pembimbing Program Studi Teknik Informatika S1
Institut Teknologi Nasional
M a l a n g

Dengan hormat

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

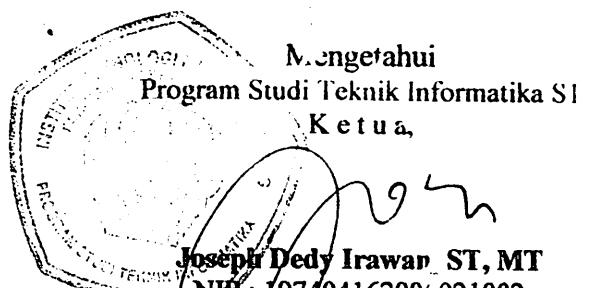
Nama : GIO VANDY
Nim : 0818095
Prodi : Teknik Informatika S1
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu 6 (enam) bulan, terhitung mulai tanggal ;

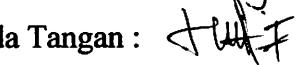
17 Oktober 2012 – 17 April 2013

Sebagai satu syarat untuk menempuh Ujian Sarjana Teknik, Program Studi Teknik Informatika S1.

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.



Form S-4a

Nama : Tommy William
Tanda Tangan : 

KUISIONER RANCANG BANGUN APLIKASI GUI MIKROTIK BERBASIS ANDROID

Mohon kesediaannya untuk mengambil waktu sejenak, dalam mengisi evaluasi berikut ini selengkap mungkin.Untuk masukan yang saudara berikan, berarti ikut membantu meningkatkan kualitas aplikasi ini.

Berilah tanda centang (✓) pada kotak jawaban yang saudara pilih.

1. Menampilkan IP client ? B C M
2. Menampilkan Interfaces ? B C M
3. menampilkan DNS client? B C M

<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
--------------------------	-------------------------------------	--------------------------

<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
--------------------------	-------------------------------------	--------------------------

<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
-------------------------------------	--------------------------	--------------------------

Keterangan :

B =Baik

C = Cukup

M = Mudah

Nama : James Mangesa
Tanda Tangan : 

KUISIONER RANCANG BANGUN APLIKASI GUI MIKROTIK BERBASIS ANDROID

Mohon kesediaannya untuk mengambil waktu sejenak, dalam mengisi evaluasi berikut ini selengkap mungkin.Untuk masukan yang saudara berikan, berarti ikut membantu meningkatkan kualitas aplikasi ini.

Berilah tanda centang (✓) pada kotak jawaban yang saudara pilih.

1. Menampilkan IP client ? B C M
2. Menampilkan Interfaces ? B C M
3. menampilkan DNS client? B C M

✓		
---	--	--

✓		
---	--	--

	✓	
--	---	--

Keterangan :

B =Baik

C = Cukup

M = Mudah

Nama : Venal

Tanda Tangan : 

KUISIONER RANCANG BANGUN APLIKASI GUI MIKROTIK BERBASIS ANDROID

Mohon kesediaannya untuk mengambil waktu sejenak, dalam mengisi evaluasi berikut ini selengkap mungkin.Untuk masukan yang saudara berikan, berarti ikut membantu meningkatkan kualitas aplikasi ini.

Berilah tanda centang (✓) pada kotak jawaban yang saudara pilih.

1. Menampilkan IP client ? B C M
2. Menampilkan Interfaces ? B C M
3. menampilkan DNS client? B C M

✓		
---	--	--

	✓	
--	---	--

		✓
--	--	---

Keterangan :

B =Baik

C = Cukup

M = Mudah

Nama : Januardo Arung
Tanda Tangan : 

KUISIONER RANCANG BANGUN APLIKASI GUI MIKROTIK BERBASIS ANDROID

Mohon kesediaannya untuk mengambil waktu sejenak, dalam mengisi evaluasi berikut ini selengkap mungkin.Untuk masukan yang saudara berikan, berarti ikut membantu meningkatkan kualitas aplikasi ini.

Berilah tanda centang (✓) pada kotak jawaban yang saudara pilih.

1. Menampilkan IP client ? B C M
2. Menampilkan Interfaces ? B C M
3. menampilkan DNS client? B C M

	✓	
		✓
✓		

Keterangan :

B =Baik

C = Cukup

M = Mudah

Nama : Made Wira Christiano
Tanda Tangan : 

KUISIONER RANCANG BANGUN APLIKASI GUI MIKROTIK BERBASIS ANDROID

Mohon kesediaannya untuk mengambil waktu sejenak, dalam mengisi evaluasi berikut ini selengkap mungkin.Untuk masukan yang saudara berikan, berarti ikut membantu meningkatkan kualitas aplikasi ini.

Berilah tanda centang (✓) pada kotak jawaban yang saudara pilih.

1. Menampilkan IP client ? B C M
2. Menampilkan Interfaces ? B C M
3. menampilkan DNS client? B C M

<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
--------------------------	--------------------------	-------------------------------------

<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
--------------------------	--------------------------	-------------------------------------

<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
-------------------------------------	--------------------------	--------------------------

Keterangan :

B =Baik

C = Cukup

M = Mudah

Nama : Narian Massudi
Tanda Tangan :

KUISIONER RANCANG BANGUN APLIKASI GUI MIKROTIK BERBASIS ANDROID

Mohon kesediaannya untuk mengambil waktu sejenak, dalam mengisi evaluasi berikut ini selengkap mungkin.Untuk masukan yang saudara berikan, berarti ikut membantu meningkatkan kualitas aplikasi ini.

Berilah tanda centang (✓) pada kotak jawaban yang saudara pilih.

1. Menampilkan IP client ? B C M
2. Menampilkan Interfaces ? B C M
3. menampilkan DNS client? B C M

		✓
✓		
	✓	

Keterangan :

B =Baik

C = Cukup

M = Mudah

Nama : Novembrie

Tanda Tangan :



KUISIONER RANCANG BANGUN APLIKASI GUI MIKROTIK BERBASIS ANDROID

Mohon kesediaannya untuk mengambil waktu sejenak, dalam mengisi evaluasi berikut ini selengkap mungkin.Untuk masukan yang saudara berikan, berarti ikut membantu meningkatkan kualitas aplikasi ini.

Berilah tanda centang (✓) pada kotak jawaban yang saudara pilih.

1. Menampilkan IP client ? B C M
2. Menampilkan Interfaces ? B C M
3. menampilkan DNS client? B C M

✓		
✓		
		✓

Keterangan :

B =Baik

C = Cukup

M = Mudah

Nama : Siska

Tanda Tangan : 

KUISIONER RANCANG BANGUN APLIKASI GUI MIKROTIK BERBASIS ANDROID

Mohon kesediaannya untuk mengambil waktu sejenak, dalam mengisi evaluasi berikut ini selengkap mungkin.Untuk masukan yang saudara berikan, berarti ikut membantu meningkatkan kualitas aplikasi ini.

Berilah tanda centang (✓) pada kotak jawaban yang saudara pilih.

1. Menampilkan IP client ? B C M
2. Menampilkan Interfaces ? B C M
3. menampilkan DNS client? B C M

<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
--------------------------	--------------------------	-------------------------------------

<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
-------------------------------------	--------------------------	--------------------------

<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
--------------------------	--------------------------	-------------------------------------

Keterangan :

B =Baik

C = Cukup

M = Mudah

Nama : Brian
Tanda Tangan : Brian

KUISIONER RANCANG BANGUN APLIKASI GUI MIKROTIK BERBASIS ANDROID

Mohon kesediaannya untuk mengambil waktu sejenak, dalam mengisi evaluasi berikut ini selengkap mungkin.Untuk masukan yang saudara berikan, berarti ikut membantu meningkatkan kualitas aplikasi ini.

Berilah tanda centang (✓) pada kotak jawaban yang saudara pilih.

1. Menampilkan IP client ? B C M
2. Menampilkan Interfaces ? B C M
3. menampilkan DNS client? B C M

<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Keterangan :

B =Baik

C = Cukup

M = Mudah

Lampiran 1. Sintak Program Winbox

```
import android.content.Intent;
import android.text.Editable;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.View.OnFocusChangeListener;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.Spinner;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TableRow.LayoutParams;
import android.widget.TextView;
import java.util.ArrayList;
import java.util.HashMap;

public class FormGenericEditor extends WolfForm
    implements View.OnClickListener, IDataReceiver, IForm,
View.OnFocusChangeListener
{
    private HashMap<String, View> ControlsCollection = new HashMap();
    private boolean TrapFlag;
    private String columnFormatString = "";
    private String genEditTitle = "Generic Edit";
    private Integer iTagSave;
    private String myDataString;
    private String myEditCommand;
    private String myId = "";

    public FormGenericEditor(WolfActivity paramWolfActivity, IForm
paramIForm, Integer paramInt)
    {
        super(paramWolfActivity, paramIForm, paramInt);
    }

    private void ExecuteSave()
    {
        String[] arrayOfString1 = this.myEditCommand.split("\n=");
        String str1 = "";
        int i = 0;
        String[] arrayOfString2;
        String str6;
        if (i >= arrayOfString1.length)
        {
            if (str1.length() > 0)
            {
```

```

// arrayOfString2 = UtilsUI.ProcessDataFormat(str1,
this.myEditCommand.replace("[.id]", this.myId)).split("\n");
// str6 = arrayOfString2[0];
}

//String str2;
// View localView1;

// str2 = arrayOfString1[i].split("=")[1].replace("[", "").replace("]", "");
// localView1 = (View)this.ControlsCollection.get(str2);
/* if (localView1 != null)
{
    str3 = "";
    // str4 = UtilsConnection.getPropertyValue(this.myDataString, str2);
    str5 = localView1.getClass().getName();
    if (!str5.contains("EditText"))
        break label263;
    str3 = ((EditText)localView1).getText().toString();
} */
}

private void createControls()
{
    String[] arrayOfString2;
    int j;
    if (this.columnFormatString != null)
    {
        arrayOfString2 = this.columnFormatString.split("\\\\[");

        j = 0;
        if (j < arrayOfString2.length);
    }
    while (true)
    {
        return;
    }
}

private void createcontrol(String paramString)
{
    if (paramString.contains(".id"))
        return;
    String str1 = paramString;
    if (paramString.contains("|"))

```

```

    str1 = paramString.split("\\\\\"")[0];
    // String str2 = UtilsConnection.getPropertyValue(this.myDataString, str1);
    Object localObject1 = paramString.substring(0, 1).toUpperCase() +
paramString.substring(1);
    TableLayout localTableLayout
    TableLayout this.mainActivity.findViewById(R.id.tblGenericEdit);
    localTableLayout.setGravity(3);
    TableRow localTableRow = new TableRow(this.mainActivity);
    localTableRow.setGravity(3);
    TextView localTextView = new TextView(this.mainActivity);
    localTableRow.addView(localTextView);
    Object localObject2;
}

private View makeDropDown(ArrayList<String> paramArrayList, String paramString1, String paramString2)
{
    Spinner localSpinner = new Spinner(this.mainActivity);
    localSpinner.setAdapter(new ArrayAdapter(this.mainActivity,
R.layout.wolfspinner, paramArrayList));
    if ((paramArrayList.contains(paramString2)) && (paramString2 != ""))
        localSpinner.setSelection(paramArrayList.indexOf(paramString2));
    if (paramArrayList.contains(paramString1))
        localSpinner.setSelection(paramArrayList.indexOf(paramString1));
    return localSpinner;
}

private EditText makeEditText(String paramString)
{
    EditText localEditText = new EditText(this.mainActivity);
    localEditText.setText(paramString);
    localEditText.setTextSize(this.FontSize);
    localEditText.setLayoutParams(new TableRow.LayoutParams(-1, -2));
    localEditText.setOnFocusChangeListener(this);
    return localEditText;
}

public void hide()
{
}

public void onActivityResult(int paramInt1, int paramInt2, Intent paramIntent)
{
}

```

```
public void onClick(View paramView)
{
    if (paramView.getId() == R.id.btnSaveGenericEdit)
        ExecuteSave();
    while (true)
    {
        super.onClick(paramView);
        return;
        // navigateBack();
    }
}

public void onFocusChange(View paramView, boolean paramBoolean)
{
    if (paramView.getTag() != null)
    {
        String str = paramView.getTag().toString();
        if ((str != null) && (str.contains("IPADDRESS")) && (!paramBoolean))
        {
            EditText localEditText = (EditText)paramView;
            if
(!localEditText.getText().toString().matches("\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}\\d{1,2}"))
                localEditText.setError("IP!");
        }
    }
}

protected void processMessage(Integer paramInt, String paramString)
{
    if (paramInt == this.iTagSave)
    {
        if (paramString.contains("!trap"))
        {
            // UtilsUI.Toast(UtilsConnection.getPropertyValue("Error:" + paramString,
"message"), this.mainActivity);
            this.TrapFlag = true;
        }
        if ((paramString.contains("!done")) && (!this.TrapFlag))
            navigateBack();
        return;
    }
    if (paramString.contains("!done"))
    {
        // UtilsUI.Toast("Command Completed", this.mainActivity);
        return;
    }
    // UtilsUI.Toast(paramString, this.mainActivity);
}
```

```
}

public void setColumns(String paramString)
{
    this.columnFormatString = paramString;
}

public void setDataString(String paramString)
{
    this.myDataString = paramString;
// this.myId = UtilsConnection.getPropertyValue(this.myDataString, ".id");
}

public void setEditCommand(String paramString)
{
    this.myEditCommand = paramString;
}

public void setTitle(String paramString)
{
    this.genEditTitle = paramString;
}

public void show()
{
    super.show();
    if (this.genEditTitle != null)
    {
        TextView localTextView
        localTextView = (TextView)this.mainActivity.findViewById(R.id.txtGenericEditTitle);
        localTextView.setText(this.genEditTitle);
        localTextView.setTextSize((float)(2.4D * this.FontSize));
    }
    createControls();
    ImageButton localImageButton1
    localImageButton1 = (ImageButton)this.mainActivity.findViewById(R.id.btnSaveGenericEdit);
    ImageButton localImageButton2
    localImageButton2 = (ImageButton)this.mainActivity.findViewById(R.id.btnCancelGenericEdit);
    localImageButton1.setOnClickListener(this);
    localImageButton2.setOnClickListener(this);
}

import android.content.Intent;
import android.text.Editable;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.View.OnFocusChangeListener;
```

```
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.Spinner;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TableRow.LayoutParams;
import android.widget.TextView;
import java.util.ArrayList;
import java.util.HashMap;

public class FormGenericEditor extends WolfForm
    implements View.OnClickListener, IDataReceiver, IForm,
View.OnFocusChangeListener
{
    private HashMap<String, View> ControlsCollection = new HashMap();
    private boolean TrapFlag;
    private String columnFormatString = "";
    private String genEditTitle = "Generic Edit";
    private Integer iTagSave;
    private String myDataString;
    private String myEditCommand;
    private String myId = "";

    public FormGenericEditor(WolfActivity paramWolfActivity, IForm
paramIForm, Integer paramInt)
    {
        super(paramWolfActivity, paramIForm, paramInt);
    }

    private void ExecuteSave()
    {
        String[] arrayOfString1 = this.myEditCommand.split("\n=");
        String str1 = "";
        int i = 0;
        String[] arrayOfString2;
        String str6;
        if (i >= arrayOfString1.length)
        {
            if (str1.length() > 0)
            {
                // arrayOfString2 = UtilsUI.ProcessDataFormat(str1,
                this.myEditCommand.replace("[.id]", this.myId)).split("\n");
                // str6 = arrayOfString2[0];
            }
        }
        //String str2;
```

```

// View localView1;

// str2 = arrayOfString1[i].split("=')[1].replace("[", "").replace("]", "");
// localView1 = (View)this.ControlsCollection.get(str2);
/* if (localView1 != null)
{
    str3 = "";
    // str4 = UtilsConnection.getPropertyValue(this.myDataString, str2);
    str5 = localView1.getClass().getName();
    if (!str5.contains("EditText"))
        break label263;
    str3 = ((EditText)localView1).getText().toString();
} */
}

private void createControls()
{
    String[] arrayOfString2;
    int j;
    if (this.columnFormatString != null)
    {
        arrayOfString2 = this.columnFormatString.split("\\\\[");
        j = 0;
        if (j < arrayOfString2.length);
    }
    while (true)
    {
        return;
    }
}

private void createcontrol(String paramString)
{
    if (paramString.contains(".id"))
        return;
    String str1 = paramString;
    if (paramString.contains("|"))
        str1 = paramString.split("\\|")[0];
    // String str2 = UtilsConnection.getPropertyValue(this.myDataString, str1);
    Object localObject1 = paramString.substring(0, 1).toUpperCase() +
paramString.substring(1);
    TableLayout localTableLayout = (TableLayout)this.mainActivity.findViewById(R.id.tblGenericEdit);
    localTableLayout.setGravity(3);
}

```

```
TableRow localTableRow = new TableRow(this.mainActivity);
localTableRow.setGravity(3);
TextView localTextView = new TextView(this.mainActivity);
localTableRow.addView(localTextView);
Object localObject2;

}

private View makeDropDown(ArrayList<String> paramArrayList, String paramString1, String paramString2)
{
    Spinner localSpinner = new Spinner(this.mainActivity);
    localSpinner.setAdapter(new ArrayAdapter(this.mainActivity,
R.layout.wolfspinner, paramArrayList));
    if ((paramArrayList.contains(paramString2)) && (paramString2 != ""))
        localSpinner.setSelection(paramArrayList.indexOf(paramString2));
    if (paramArrayList.contains(paramString1))
        localSpinner.setSelection(paramArrayList.indexOf(paramString1));
    return localSpinner;
}

private EditText makeEditText(String paramString)
{
    EditText localEditText = new EditText(this.mainActivity);
    localEditText.setText(paramString);
    localEditText.setTextSize(this.FontSize);
    localEditText.setLayoutParams(new TableRow.LayoutParams(-1, -2));
    localEditText.setOnFocusChangeListener(this);
    return localEditText;
}

public void hide()
{
}

public void onActivityResult(int paramInt1, int paramInt2, Intent paramIntent)
{
}

public void onClick(View paramView)
{
    if (paramView.getId() == R.id.btnSaveGenericEdit)
        ExecuteSave();
    while (true)
    {
        super.onClick(paramView);
```

```

return;
// navigateBack();
}
}

public void onFocusChange(View paramView, boolean paramBoolean)
{
if (paramView.getTag() != null)
{
    String str = paramView.getTag().toString();
    if ((str != null) && (str.contains("IPADDRESS")) && (!paramBoolean))
    {
        EditText localEditText = (EditText)paramView;
        if
(!localEditText.getText().toString().matches("\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}\\d{1,2}"))
            localEditText.setError("IP!");
    }
}
}

protected void processMessage(Integer paramInt, String paramString)
{
if (paramInt == this.iTagSave)
{
    if (paramString.contains("!trap"))
    {
        // UtilsUI.Toast(UtilsConnection.getPropertyValue("Error:" + paramString,
"message"), this.mainActivity);
        this.TrapFlag = true;
    }
    if ((paramString.contains("!done")) && (!this.TrapFlag))
        navigateBack();
    return;
}
if (paramString.contains("!done"))
{
    // UtilsUI.Toast("Command Completed", this.mainActivity);
    return;
}
// UtilsUI.Toast(paramString, this.mainActivity);
}

public void setColumns(String paramString)
{
    this.columnFormatString = paramString;
}

```

```

public void setDataString(String paramString)
{
    this.myDataString = paramString;
    // this.myId = UtilsConnection.getPropertyValue(this.myDataString, ".id");
}

public void setEditCommand(String paramString)
{
    this.myEditCommand = paramString;
}

public void setTitle(String paramString)
{
    this.genEditTitle = paramString;
}

public void show()
{
    super.show();
    if (this.genEditTitle != null)
    {
        TextView localTextView
        = (TextView)this.mainActivity.findViewById(R.id.txtGenericEditTitle);
        localTextView.setText(this.genEditTitle);
        localTextView.setTextSize((float)(2.4D * this.FontSize));
    }
    createControls();
    ImageButton localImageButton1
    = (ImageButton)this.mainActivity.findViewById(R.id.btnSaveGenericEdit);
    ImageButton localImageButton2
    = (ImageButton)this.mainActivity.findViewById(R.id.btnCancelGenericEdit);
    localImageButton1.setOnClickListener(this);
    localImageButton2.setOnClickListener(this);
}
}

import android.content.DialogInterface;
import android.content.DialogInterface.OnClickListener;
import android.content.Intent;
import android.text.Editable;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageButton;

public class FormRouterDetails extends WolfForm
    implements View.OnClickListener, IDataReceiver, IForm
{
    private int RbNumber = -1;
}

```

```
public FormRouterDetails(WolfActivity paramWolfActivity, IForm
paramIForm, Integer paramInt)
{
    super(paramWolfActivity, paramIForm, paramInt);
}

public int getRbNumber()
{
    return this.RbNumber;
}

public void hide()
{
}

public void onActivityResult(int paramInt1, int paramInt2, Intent paramInt)
{
}

public void onClick(View paramView)
{
    if (paramView.getId() == R.id.btnCancelRouterDetails)
        new FormRouterList(this.mainActivity, null,
Integer.valueOf(R.layout.routerlist)).show();
    String str1;
    String str2;
    String str3;
    String str4;
    if (paramView.getId() == R.id.btnSaveRouterDetails)
    {
        str1 = ((EditText)this.mainActivity.findViewById(R.id.txtPort)).getText().toString();
        str2 = ((EditText)this.mainActivity.findViewById(R.id.txtIp)).getText().toString();
        str3 = ((EditText)this.mainActivity.findViewById(R.id.txtUsername)).getText().toString();
        str4 = ((EditText)this.mainActivity.findViewById(R.id.txtPassword)).getText().toString();
        if (!str1.contentEquals("8728"))
        {
        }
    }
    else
    {
        return;
    }
}
```

```
        }

    // UtilsUI.editRouterToSettings(str2, this.RbNumber, str1, str3, str4,
this.settings);
    this.parentForm.show();
}

protected void processMessage(Integer paramInt, String paramString)
{
}

public void setData(String paramString1, String paramString2, String paramString3, String paramString4)
{
    View localView = this.mainActivity.findViewById(R.id.Layout111);

}

public void setRbNumber(int paramInt)
{
    this.RbNumber = paramInt;
}

public void show()
{
    super.show();
    ImageButton           localImageButton1      =
(localImageButton) this.mainActivity.findViewById(R.id.btnCancelRouterDetails);
    if (localImageButton1 != null)
        localImageButton1.setOnClickListener(this);
    ImageButton           localImageButton2      =
(localImageButton) this.mainActivity.findViewById(R.id.btnSaveRouterDetails);
    if (localImageButton2 != null)
        localImageButton2.setOnClickListener(this);

}
```

Lampiran 2.Sintak Layout Winbox

```
android:layout_width="fill_parent" android:layout_height="fill_parent"
android:weightSum="1.0"
    xmlns:android="http://schemas.android.com/apk/res/android">
        <LinearLayout android:id="@+id/linearLayout1"
            android:layout_width="fill_parent" android:layout_height="wrap_content"
            android:weightSum="1.0" android:layout_weight="0.05">
            <ToggleButton android:id="@+id/ifEnabled"
                android:layout_width="wrap_content" android:layout_height="fill_parent"
                android:checked="true" android:text="@string/togglebutton" />
            <ImageButton android:id="@+id/btnTorch"
                android:layout_width="wrap_content" android:layout_height="fill_parent"
                android:src="@drawable/ip" android:maxWidth="3.0dip"
                android:maxHeight="3.0dip" />
            <LinearLayout android:orientation="vertical"
                android:id="@+id/linearLayout3" android:layout_width="wrap_content"
                android:layout_height="43.0dip">
                <TextView android:textAppearance="?android:textAppearanceSmall"
                    android:id="@+id/textView2" android:layout_width="wrap_content"
                    android:layout_height="wrap_content" android:text="@string/spaces"
                    android:layout_weight="0.12" />
                <ImageView android:id="@+id/imageView1"
                    android:layout_width="wrap_content" android:layout_height="15.0dip"
                    android:src="@drawable/greenbutton" android:layout_weight="0.12"
                    android:contentDescription="@string/green_button" />
            </LinearLayout>
            <LinearLayout android:orientation="vertical"
                android:id="@+id/linearLayout2" android:layout_width="wrap_content"
                android:layout_height="fill_parent">
                <TextView android:textAppearance="?android:textAppearanceMedium"
                    android:id="@+id/ifName" android:layout_width="wrap_content"
                    android:layout_height="wrap_content" android:text="@string/textview" />
                <TextView android:textAppearance="?android:textAppearanceSmall"
                    android:id="@+id/ifType" android:layout_width="wrap_content"
                    android:layout_height="wrap_content" android:text="@string/textview" />
            </LinearLayout>
            <TextView android:textAppearance="?android:textAppearanceLarge"
                android:textSize="5.0pt" android:id="@+id/textView3"
                android:layout_width="wrap_content" android:layout_height="wrap_content"
                android:text="@string/morespaces" />
            <LinearLayout android:orientation="vertical"
                android:id="@+id/linearLayout4" android:layout_width="wrap_content"
                android:layout_height="fill_parent">
                <TextView android:textAppearance="?android:textAppearanceSmall"
                    android:id="@+id/ifComment" android:layout_width="wrap_content"
                    android:layout_height="wrap_content" android:text="@string/textview" />
```

```
<TextView android:textAppearance="?android:textAppearanceSmall"  
        android:textSize="10.0sp" android:id="@+id/txtRx"  
        android:layout_width="wrap_content" android:layout_height="wrap_content"  
        android:text="@string/morespaces" />  
<TextView android:textAppearance="?android:textAppearanceSmall"  
        android:textSize="10.0sp" android:id="@+id/txtMac"  
        android:layout_width="wrap_content" android:layout_height="wrap_content"  
        android:text="@string/morespaces" />  
<TextView android:textAppearance="?android:textAppearanceSmall"  
        android:textSize="10.0sp" android:id="@+id/txtSignal"  
        android:layout_width="wrap_content" android:layout_height="wrap_content"  
        android:text="@string/morespaces" />  
    </LinearLayout>  
</LinearLayout>  
<LinearLayout android:orientation="vertical" android:id="@+id/llAdd"  
        android:layout_width="fill_parent" android:layout_height="wrap_content"  
        android:layout_weight="0.05" />  
</LinearLayout>
```

```
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical" >
```

```
<TextView  
    android:id="@+id/textView1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="IP V4 Menu"  
    android:textAppearance="?android:attr/textAppearanceLarge" />
```

```
<Button  
    android:id="@+id/button1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Button" />
```

```
<Button  
    android:id="@+id/button2"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Button" />
```

```
</LinearLayout>  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical" >
```

```
<TextView
```

```
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Main Menu"
    android:textAppearance="?android:attr/textAppearanceLarge" />

<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="-" />

<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="-" />

<Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="-" />

</LinearLayout>
```