

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Jaringan komputer adalah sebuah sistem yang terdiri atas komputer-komputer yang saling terhubung yang didesain dengan tujuan agar dapat berbagi sumber daya, berkomunikasi dan dapat mengakses informasi melalui situs web. Oleh karena itu, suatu jaringan komputer harus dijaga kestabilan operasionalnya. Jaringan komputer dibangun dari beberapa komponen penting, salah satunya yaitu router dimana router ini merupakan media penghubung dan pengatur pada 2 buah jaringan atau lebih yang berguna dalam meneruskan data dari satu jaringan ke jaringan yang lain. Mikrotik merupakan sebuah sistem operasi router yang digunakan untuk menjalankan dan mengatur segala aktivitas jaringan secara menyeluruh. Mikrotik didesain untuk keperluan administrasi jaringan komputer seperti merancang dan membangun sebuah sistem jaringan komputer skala kecil hingga yang kompleks sekalipun.

Manajemen jaringan adalah suatu pekerjaan untuk memelihara jaringan agar tetap dalam keadaan yang baik. Salah satu dari manajemen jaringan adalah manajemen performa jaringan, manajemen ini dimulai dengan melakukan monitoring pada jaringan, salah satunya adalah memonitoring besaran *bandwidth* kemudian jika ada *client* yang melebihi batas *bandwidth* dan menyebabkan ketidakstabilan jaringan maka akan dilakukan tindakan agar jaringan bisa kembali pada performa yang baik. Manajemen bandwidth sendiri dapat diterapkan pada jaringan skala kecil seperti pada Lab Informatika Untan. Saat ini di Lab Informatika Untan manajemen yang dilakukan masih secara manual yaitu admin atau asisten lab jaringan harus masuk pada web mikrotik untuk melakukan manajemen bandwidth secara langsung.

Aplikasi *mobile* adalah program komputer yang dirancang untuk berjalan pada peranti bergerak seperti ponsel/ *tablet* dan merupakan salah satu hasil dari perkembangan teknologi di dunia. Adapun kebutuhan aplikasi *mobile* berbasis android untuk memudahkan pekerjaan semakin meningkat. Aplikasi *mobile* android yang lebih fleksibel penggunaannya dinilai lebih memudahkan dalam membantu

pekerjaan yang memerlukan internet. Menurut Kementrian Komunikasi dan Informatika tingginya minat untuk memanfaatkan aplikasi *mobile* android ini dikarenakan manusia lebih sering membawa *smartphone* dari pada membawa laptop, sehingga dengan memanfaatkan aplikasi *mobile* android dinilai lebih memudahkan pekerjaan karena bisa di akses dimanapun dengan hanya menggunakan *smartphone*.

Berdasarkan pembahasan diatas, maka diperlukan aplikasi yang bisa melakukan manajemen jaringan pada mikrotik dengan memanfaatkan teknologi android *mobile* pada Lab Informatika Untan, sehingga bisa membuat kerja dari administrator atau asisten lab jaringan untuk melakukan manajemen jaringan pada mikrotik menjadi lebih mudah.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang diatas, rumusan masalah yang dapat diambil adalah bagaimana merancang dan membangun sebuah aplikasi berbasis android yang dapat melakukan manajemen jaringan pada mikrotik untuk membantu admin jaringan lebih mudah memantau besaran *bandwith client* dan melakukan tindakan jika ada besaran *bandwith* yang tidak sesuai.

## **1.3 Tujuan Penelitian**

Tujuan dari penelitian ini adalah membuat aplikasi manajemen jaringan pada mikrotik yang diharapkan dapat membantu admin jaringan untuk lebih mudah melakukan manajemen performa jaringan pada mikrotik yaitu memantau besaran *bandwith client* dan melakukan tindakan yaitu memutus koneksi pada *client*.

## **1.4 Pembatasan Masalah**

Pembatasan masalah dari penelitian yang akan dilakukan adalah sebagai berikut.

1. Aplikasi ini di buat untuk Lab Informatika Untan.
2. Manajemen jaringan yang dilakukan adalah manajemen performa yaitu memonitoring besaran *bandwith* dan melakukan tindakan memutuskan koneksi pada *client*.

3. Tidak membahas *firewall*.
4. Aplikasi dibuat berbasis android *mobile*.
5. Jaringan yang dimanajemen adalah jaringan pada mikrotik.

### 1.5 Sistematika Penulisan Skripsi

Adapun sistematika penulisan skripsi tugas akhir ini tersusun atas lima bab, yang terdiri dari Bab I Pendahuluan, Bab II Tinjauan Pustaka, Bab III Metodologi Penelitian dan Perancangan Sistem, Bab IV Hasil dan Analisis, serta Bab V Penutup.

**Bab I Pendahuluan** adalah bab yang berisi latar belakang penelitian, perumusan masalah, tujuan penelitian, pembatasan masalah, dan sistematika penulisan

**Bab II Tinjauan Pustaka** adalah bab yang berisi uraian sistematis tentang hasil-hasil penelitian yang didapat oleh peneliti terdahulu dan landasan teori yang berhubungan dengan penelitian yang akan dilakukan.

**Bab III Metodologi Penelitian dan Perancangan Sistem** adalah bab yang berisi tentang bahan penelitian, alat yang digunakan, metode penelitian, variable atau data, serta diagram alir penelitian.

**Bab IV Hasil dan Analisis** adalah bab yang berisi data hasil perancangan, pengujian dan sebagainya yang telah dirancang pada Bab III. Setiap hasil yang disajikan akan dilakukan analisis untuk mengarah kepada suatu kesimpulan.

**Bab V Penutup** adalah bab yang berisi kesimpulan dari penelitian yang telah dilakukan dan saran/rekomendasi untuk perbaikan, pengembangan atau kesempurnaan/kelengkapan penelitian yang telah dilakukan.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Manajemen Jaringan**

Manajemen jaringan merupakan pengaturan, pengurusan atau pengelolaan elemen pada jaringan agar jaringan tetap dapat digunakan untuk kedepannya (Subramanian, 2000).

Manajemen jaringan dapat dikategorikan menjadi lima macam, yaitu:

1. *Configuration management* (manajemen konfigurasi)

Meliputi pengaturan alamat (*Address*) dan perubahan konfigurasi dari jaringan dan komponen-komponen di dalamnya.

2. *Fault management* (manajemen kesalahan)

Meliputi deteksi masalah, pengisolasian kesalahan, dan perbaikan sehingga jaringan dapat kembali ke operasi normal.

3. *Performance management* (manajemen performa)

Meliputi pengaturan tingkah laku performa dari jaringan yang sedang berjalan dimana performa jaringan ini memantau kinerja jaringan dan melakukan tindakan untuk mengembalikan performa jaringan jika terjadi penurunan.

4. *Security management* (manajemen keamanan)

Mengatur keamanan fisik jaringan, akses ke sumber daya jaringan, dan keamanan komunikasi yang terjadi di dalam jaringan.

5. *Accounting management* (manajemen akunting)

Meliputi manajemen keuangan yang dikeluarkan dalam pengelolaan jaringan.

#### **2.2 Aplikasi Mobile**

Pengertian aplikasi menurut Reito Meiler (Profesional *Android Application Development* 2009:2) adalah program yang digunakan orang untuk melakukan sesuatu pada sistem komputer. *Mobile* dapat diartikan sebagai perpindahan yang mudah dari satu tempat ke tempat yang lain, misalnya telepon *mobile* berarti bahwa terminal telepon yang dapat berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi. Sistem aplikasi *mobile* merupakan aplikasi yang dapat digunakan walaupun pengguna berpindah dengan

mudah dari satu tempat ketempat lain lain tanpa terjadipemutusan atau terputusnya komunikasi. Aplikasi ini dapat diakses melalui perangkat nirkabel seperti pager, seperti telepon seluler dan PDA. Adapun karakteristik perangkat *mobile* yaitu:

- Ukuran yang kecil : perangkat *mobile* memiliki ukuran yang kecil. Konsumen menginginkan perangkat yang terkecil untuk kenyamanan dan mobilitas mereka.
- Memori yang terbatas : perangkat *mobile* juga memiliki *memory* yang kecil, yaitu *primary* (RAM) dan *secondary* (disk).
- Daya proses yang terbatas : sistem *mobile* tidaklah setangguh rekan mereka yaitu *desktop*.
- Mengkonsumsi daya yang rendah : perangkat *mobile* menghabiskan sedikit daya dibandingkan dengan mesin *desktop*
- Kuat dan dapat diandalkan : karena perangkat *mobile* selalu dibawa kemana saja, mereka harus cukup kuat untuk menghadapi benturan-benturan, gerakan, dan sesekali tetesan-tetesan air.
- Konektivitas yang terbatas : perangkat *mobile* memiliki bandwith rendah, beberapa dari mereka bahkan tidak tersambung.
- Masa hidup yang pendek : perangkat-perangkat konsumen ini menyala dalam hitungan detik kebanyakan dari mereka selalu menyala.

### 2.3 Android

Menurut Nasruddin Safaat (Pemrograman aplikasi *mobile smartphone* dan tablet PC berbasis android, 2012:1) Android adalah sebuah sistem operasi pada *handphone* yang bersifat terbuka dan berbasis pada sistem operasi linux. Android bisa digunakan oleh setiap orang yang ingin menggunakannya pada perangkat mereka. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk bermacam peranti bergerak.

Android merupakan *software environment* dibuat untuk perangkat *mobile*. Android menggunakan OS kernel linux, UI yang berkualitas, *end-user application*, *code libraries*, *framework* aplikasi, mendukung multimedia dan banyak lagi. Aplikasi yang dibuat untuk android menggunakan bahasa pemrograman java.

Ada dua cara untuk membangun atau membuat aplikasi berbasis android, yaitu memiliki perangkat telepon seluler yang berbasis android langsung dan menggunakan emulator yang sudah disediakan oleh google. Sebelum memulai membangun aplikasi berbasis android, diperlukan beberapa perangkat, antara lain (Dimarzio, 2008):

- Android Studio
- *Sun's Java Development Kit* (JDK)
- *The Android Software Developer's Kit* (SDK).

Windows, linux, dan mac OS X merupakan sistem operasi yang dapat digunakan untuk pengembangan pembuatan aplikasi berbasis android dengan memanfaatkan android SDK (Brady, 2008).

Awalnya, google Inc. membeli android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, 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 terbuka pada perangkat seluler. Di lain pihak, google merilis kode-kode android dibawah lisensi apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler.

Pada awal peluncurannya, google meyakini bahwa *platform* perangkat *mobile* android memiliki kesempatan yang sangat besar dalam pengembangan aplikasi. Google mengumumkan *Open Handset Alliance* (OHA) dan *platform* android pada November 2007, dan meluncurkan android *Software Development Kit* (SDK) pertama yang masih dalam versi beta di waktu yang sama. Dalam waktu yang tidak lama, lebih dari satu juta orang mengunduh android SDK dari website 14 google. Di Amerika Serikat, T-Mobile mengumumkan perangkat *mobile* android bergelar G1 pada Oktober 2008, dan diperkirakan ratusan ribu perangkat G1 terjual pada akhir tahun yang sama. Android memiliki potensi yang besar untuk menghilangkan batasan dan kendala yang selama ini muncul dalam mengembangkan suatu perangkat lunak versi *mobile phone*. Dari berbagai hal yang telah disiapkan google, android berhasil mengembangkan pasar aplikasi perangkat

*mobile*, serta memberikan kesempatan besar untuk para pengembang perangkat lunak untuk mengambil keuntungan dari aplikasi yang dibuatnya (Andry, 2011).

## 2.4 Jaringan Komputer

Jaringan dapat diartikan sebagai interkoneksi dari beberapa komputer. Komputer-komputer ini dapat dihubungkan secara bersama-sama untuk kepentingan yang berbeda dan menggunakan berbagai jenis kabel yang terpisah. Sehingga mengarah ke pengurangan pemakaian waktu dan meningkatnya produktifitas. (Pavani, Chandrika, Krishna, Vol.1, 2012).

Menurut Forouzan (2008) Jaringan adalah seperangkat *devices* (biasanya disebut sebagai *nodes*) yang dihubungkan melalui *communication links*. Pada dasarnya tujuan daripada pembuatan jaringan adalah untuk:

1. Dapat menghemat *hardware* seperti berbagi pemakaian *printer* dan CPU.
2. Melakukan komunikasi, contohnya surat elektronik, *instant messaging*, *chatting*.
3. Mendapatkan akses informasi dengan cepat, contohnya *web browsing*.
4. Melakukan *sharing* data.

Secara sederhana jaringan komputer mempunyai gambaran seperti berikut ini. Dua buah komputer atau lebih yang masing-masing memiliki kartu jaringan (NIC= *Network Interface Card*), kemudian dihubungkan dengan menggunakan kabel atau nirkabel sebagai media transmisi data. Dalam jaringan tersebut terdapat sebuah perangkat lunak (*software*) sistem operasi jaringan yang dapat menghubungkan komputer-komputer tersebut. Rangkaian sederhana seperti di atas sudah bisa dikatakan sebagai sebuah jaringan komputer, tentunya sebuah jaringan komputer yang sederhana. Setiap komputer atau perangkat jaringan yang terhubung dengan jaringan disebut node. Sebuah jaringan komputer dapat memiliki dua, puluhan, ribuan atau bahkan jutaan node. Apabila ingin membuat jaringan komputer yang lebih luas lagi jangkauannya, maka diperlukan peralatan tambahan seperti hub, bridge, switch, router, gateway sebagai peralatan interkoneksinya.

Dengan adanya jaringan komputer ini banyak keuntungan yang akan kita dapatkan, contohnya pertukaran data antar komputer atau perangkat dalam sebuah jaringan, *sharing* perangkat, *sharing* program, dan lain sebagainya. Selain beberapa

contoh di atas tentu masih banyak lagi kegunaan dari sebuah jaringan komputer (Forouzan, 2008).

Jaringan komputer adalah sebuah sistem yang terdiri atas komputer dan perangkat jaringan yang bekerja bersama-sama untuk mencapai tujuan yang sama (Wahana Komputer, 2010).

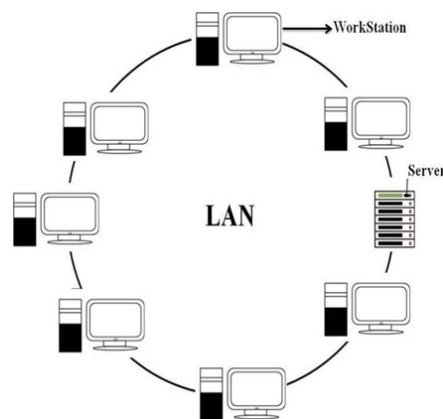
Tujuan dari jaringan komputer adalah :

- a. Membagi fungsi sumber daya seperti berbagi pemakaian *printer*, CPU, RAM, *harddisk*.
- b. Komunikasi : contohnya email, *instant messaging*, *chatting*.
- c. Akses informasi : contohnya web *browsing*.

Agar dapat mencapai tujuan yang sama, setiap bagian dari jaringan komputer meminta dan memberikan layanan. *Client* sebagai yang menerima layanan. Arsitektur seperti ini sering disebut dengan sistem *client server*, dan digunakan pada hampir seluruh aplikasi jaringan komputer.

Jenis jaringan komputer ada 4 jenis jaringan komputer, yaitu:

- a. *Local Area Network* (LAN), adalah jaringan yang dibatasi oleh area yang relatif kecil, umumnya dibatasi oleh area lingkungan seperti sebuah perkantoran di sebuah gedung atau sebuah sekolah dan tidak jauh dari sekitar 1 km persegi (Madcoms, 2016).

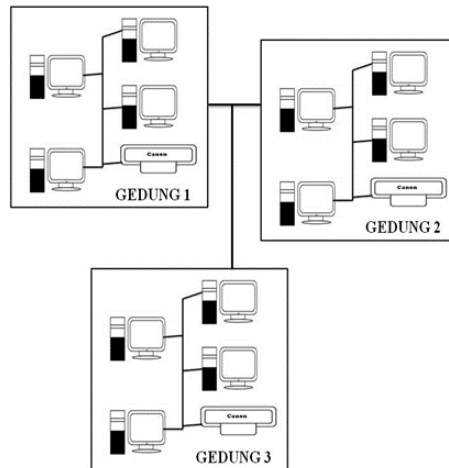


**Gambar 2.1** Jaringan LAN

- b. *Metropolitan Area Network* (MAN), meliputi area yang lebih besar dari LAN, misalnya antar wilayah dalam satu provinsi. Dalam hal ini jaringan menghubungkan beberapa buah jaringan-jaringan kecil ke dalam lingkungan area yang lebih besar, sebagai contoh yaitu jaringan bank dimana beberapa kantor

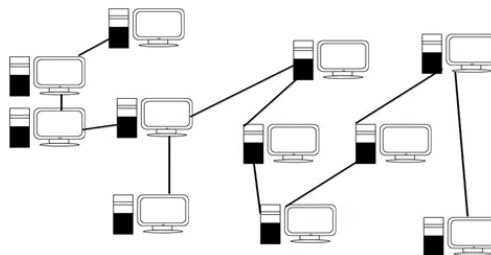


cabang sebuah bank di dalam sebuah kota besar dihubungkan antara satu dengan lainnya (Madcoms, 2016).



**Gambar 2.2** Jaringan MAN

c. *Wide Area Network* (WAN), adalah jaringan yang lingkupnya sudah menggunakan media satelit atau kabel bawah laut, sebagai contoh keseluruhan jaringan bank yang ada di Indonesia atau yang ada di negaranegara lain (Madcoms, 2016).



**Gambar 2.3** Jaringan WAN

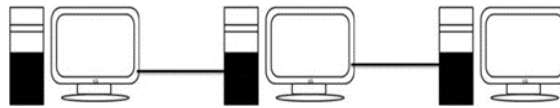
d. *Wireless Local Area Network* (WLAN), adalah jaringan komputer yang menggunakan gelombang sinyal radio sebagai transmisi data. Data ditransfer dari satu perangkat ke perangkat yang lain tanpa menggunakan kabel sebagai perantara (Madcoms, 2016).

Tipe-tipe dari jaringan komputer sebagai berikut:

a. *Peer to peer*

Menurut Alan Nur Aditya (2011 : 21), pada jaringan ini tidak ada komputer *client* maupun *computer server* karena semua komputer dapat melakukan pengiriman maupun penerimaan informasi sehingga semua komputer berfungsi sebagai *client* sekaligus sebagai *server*. Sistem operasi jaringan model *peer to peer*

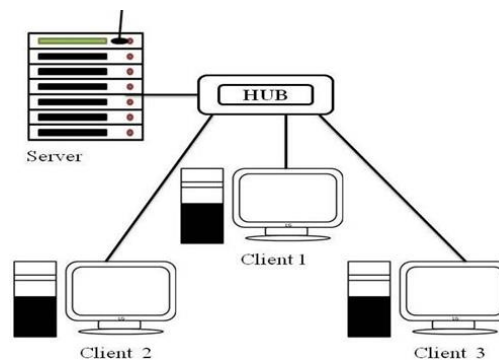
memungkinkan seorang user membagi sumber daya yang ada dikomputernya, baik itu file data, *printer*, dan peripheral lain. Namun, model ini tidak mempunyai sebuah file server atau sumber daya yang terpusat.



**Gambar 2.4** Jaringan *Peer to Peer*

b. *Client Server*

Menurut Alan Nur Aditya (2011 : 21) Pada Jaringan *client server* terdapat satu atau beberapa komputer server dan komputer *client*. *Computer* yang akan menjadi komputer *client* dan diubah-ubah melalui *software* jaringan pada protokolnya. Komputer *client* merupakan perantara untuk dapat mengakses data pada komputer server, sedangkan komputer server merupakan penyedia informasi yang diperlukan oleh komputer *client*.



**Gambar 2.5** Jaringan *Client Server*

## 2.5 Router

Router adalah perangkat jaringan komputer yang berfungsi untuk menghubungkan beberapa jaringan, dan meneruskan paket data dari satu jaringan ke jaringan lain, baik jaringan yang menggunakan teknologi yang sama atau yang berbeda. Router mengirimkan paket data melalui sebuah jaringan atau internet menuju tujuannya, melalui sebuah proses yang dikenal sebagai *routing*. ( Cisco Networking Academy Program, 2011 ).

Router mengirimkan paket data melalui sebuah jaringan atau internet menuju tujuannya, melalui sebuah proses yang dikenal sebagai *routing*. Proses *routing* terjadi pada lapisan 3 (lapisan jaringan seperti internet protocol) dari stack

protokol tujuh lapis OSI. Router dapat digunakan untuk menghubungkan banyak jaringan kecil ke sebuah jaringan yang lebih besar, yang disebut dengan *internetwork*, atau untuk membagi sebuah jaringan besar ke dalam beberapa *subnetwork* untuk meningkatkan kinerja dan juga mempermudah manajemennya. Router juga kadang digunakan untuk mengoneksikan dua buah jaringan yang menggunakan media yang berbeda (seperti halnya router wireless yang pada umumnya selain dapat menghubungkan komputer dengan menggunakan radio, juga mendukung penghubungan komputer dengan kabel UTP), atau berbeda arsitektur jaringan, seperti halnya dari ethernet ke token ring.

Jenis-jenis router sebagai berikut.

- a. Router aplikasi, adalah sebuah aplikasi yang bisa anda instal pada sistem operasi komputer, sehingga sistem operasi *computer* tersebut dapat bekerja seperti router, misalnya aplikasi WinGate, WinProxy Winroute, SpyGate dll.
- b. Router *hardware*, adalah sebuah *hardware* yang memiliki kemampuan seperti router, maka dengan *hardware* tersebut *IP Address* dapat dibagi, misalnya dari router ini adalah access point, wilayah yang mendapat *Ip Address* dan koneksi internet disebut *Hot Spot Area*.
- c. Router PC, adalah sebuah komputer yang dimodifikasi sedemikian rupa sehingga dapat digunakan sebagai router. PC Router pada umumnya memiliki fungsi seperti router, yaitu untuk mengirim paket data melalui sebuah jaringan atau internet menuju suatu tujuan sebagai penghubung dua buah jaringan atau lebih pada jaringan yang berbeda. Sistem operasi yang populer untuk router PC saat ini adalah mikrotik.

## 2.6 Mikrotik

Mikrotik adalah sistem operasi dan perangkat lunak yang dapat digunakan untuk menjadikan komputer biasa menjadi router *network* yang handal, mencakup berbagai fitur yang dibuat untuk IP jaringan dan jaringan wireless. Fitur-fitur tersebut diantaranya : Firewall & Nat, *Routing*, Hotspot, Point to Point Tunneling Protocol, DNS server, DHCP Server, Hotspot, dan masih banyak lagi fitur lainnya. Mikrotik dapat digunakan dalam 2 tipe, yaitu dalam bentuk perangkat keras dan perangkat lunak. Dalam bentuk perangkat keras, mikrotik biasanya sudah

diinstalasi pada suatu *board* tertentu, sedangkan dalam bentuk perangkat lunak, mikrotik merupakan satu distro linux yang memang dikhususkan untuk fungsi router (Cisco, *Networking Academy Program: Second-Year Companion Guide 2nd Edition*, 2011).

## 2.7 Java

Java adalah bahasa pemrograman serbaguna. Java dapat digunakan untuk membuat suatu program sebagaimana anda membuatnya dengan bahasa seperti Pascal atau C++. Yang lebih menarik, java juga mendukung sumber daya internet yang saat ini populer, yaitu *World Wide Web* (WWW) atau yang sering di sebut web saja. Java juga mendukung aplikasi klien/server, baik dalam jaringan *local* (LAN) maupun jaringan berskala luas (WAN).

Bahasa ini awalnya 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 yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana. Aplikasi-aplikasi berbasis Java umumnya dikompilasi ke dalam *p-code* (*bytecode*) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/ non-spesifik (*general purpose*), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi Java mampu berjalan di beberapa *platform* sistem operasi yang berbeda, Java dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun". Saat ini java merupakan bahasa pemrograman yang paling populer digunakan dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web (Kadir, 2005).

## 2.8 JSON (*JavaScript Object Notation*)

JSON (*JavaScript Object Notation*) adalah sebuah standar yang berbasis teks yang didesain untuk pertukaran data yang mudah dibaca manusia (Sriparasa, 2013).

Kegunaan dari JSON:

- Digunakan saat menulis aplikasi berbasis *JavaScript* yang memerlukan ekstensi *browser* dan website.
- Format JSON digunakan untuk mentransmisikan data yang berstruktur melalui jaringan.
- Biasanya digunakan untuk mentransmisikan data antara server dan aplikasi web.
- *Web service* dan API menggunakan format JSON untuk menyediakan data publik.
- Dapat digunakan pada bahasa pemrograman modern.

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari bahasa pemrograman *JavaScript*, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh *programmer* keluarga C termasuk C, C++, C#, Java, *JavaScript*, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data.

JSON terbuat dari dua struktur sebagai berikut.

- Kumpulan pasangan nama/ nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau associative array.
- Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemrograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini.

## 2.9 PHP

PHP (*Hypertext Preprocessor*) merupakan salah satu yang banyak digunakan sebagai *server-side scripting language* untuk mengembangkan aplikasi berbasis web. PHP adalah *script* yang digunakan untuk membuat halaman website yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu baru atau up to date. Semua *script* PHP dieksekusi pada server dimana *script* tersebut dijalankan (Anhar, 2010). Dalam membangun aplikasi berbasis Android, aplikasi akan berhubungan dengan server melalui web *service* yang dibuat menggunakan bahasa pemrograman PHP.

## 2.10 UML (*Unified Modeling Language*)

Membantu dalam pengembangan perangkat lunak dikenal istilah pemodelan. Salah satu pemodelan yang saat ini paling banyak digunakan oleh pengembang perangkat lunak adalah UML (*Unified Modeling Language*). UML adalah standar bahasa yang sering digunakan dalam bidang industri untuk mendefinisikan requirement (kebutuhan), membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

UML terbagi atas 3 (tiga) kategori, yaitu diagram struktur (*structure diagram*), diagram kelakuan sistem (*behaviour diagram*), dan diagram interaksi (*interaction diagram*) (Sukamto dan Shalahuddin, 2014).

- a. *Structure diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. *Structure diagram* terdiri dari *class diagram*, *object diagram*, *component diagram*, *composite diagram*, *package diagram*, dan *deployment diagram*.
- b. *Behaviour diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem. *Behaviour diagram* terdiri dari *use case diagram*, *activity diagram*, *state machine system*.
- c. *Interaction diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem. *Interaction diagram* terdiri dari *sequence*

diagram, *communication* diagram, *timing* diagram, *interaction* overview diagram.

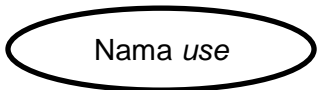
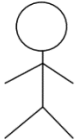

UML akan digunakan untuk merancang dan menggambarkan model dari aplikasi berbasis android dan berbasis web yang akan dibangun. Beberapa diagram yang akan digunakan adalah *use case* diagram, *activity* diagram, *class* diagram, dan *sequence* diagram.

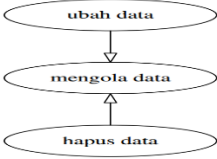

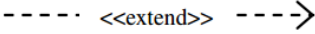
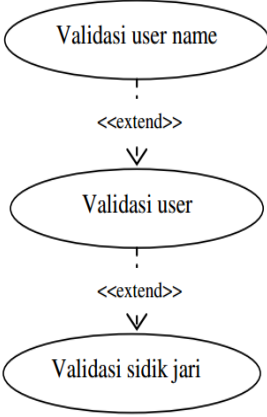
### 2.10.1. Use case Diagram

*Use case* diagram merupakan pemodelan untuk kelakuan (*behaviour*) sistem yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Berikut adalah simbol-simbol yang ada pada diagram *use case*.

**Tabel 2.1** Simbol-simbol *Use case* Diagram

No	Simbol	Deskripsi
1.	<i>Use case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal fase nama <i>use case</i> .
2.	Aktor/ <i>actor</i> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dibuat di luar sistem yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3.	Generalisasi/ <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i>

No	Simbol	Deskripsi
		<p>dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya :</p>  <p>Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).</p>
4.	<p>Asosiasi/<i>association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
5.	<p>Ekstensi/<i>extend</i></p> 	<p>Relasi <i>use case</i> tambahan keesebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal</p>  <p>Arah panah mengarah pada <i>use case</i> yang ditambahkan, biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang</p>



No	Simbol	Deskripsi
		sama dengan <i>use case</i> yang menjadi induknya.






Sumber: (Sukamto & Shalahuddin, 2014).

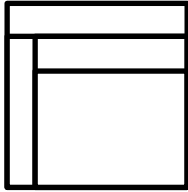
### 2.10.2. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu di perhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas.

**Tabel 2.2** Simbol-simbol *Activity Diagram*

No	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status akhir 	Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

No	Simbol	Deskripsi
6.	<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber: (Sukamto & Shalahuddin, 2014).




### 2.10.3. Class Diagram



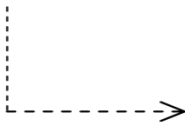
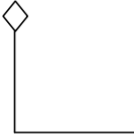
Diagram kelas atau *class* diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan *method* atau operasi. Berikut penjelasan atribut dan *method*.

- Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
- Operasi atau *method* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Berikut adalah simbol-simbol yang ada pada diagram kelas:

**Tabel 2.3** Simbol-simbol *Class* Diagram

No	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktur sistem.
2.	Antarmuka/ <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.	Asosiasi/ <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .

No	Simbol	Deskripsi
4.	Asosiasi berarah/ <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
6.	Kebergantungan / <i>dependensi</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
7.	Agregasi/ <i>aggregation</i> 	Relasi antar kelas dengan makna semua bagian ( <i>whole-part</i> ).

Sumber: (Sukamto & Shalahuddin, 2014).

#### 2.10.4. Sequence Diagram

*Sequence Diagram* adalah salah satu dari diagram - diagram yang ada pada UML, *sequence diagram* ini adalah diagram yang menggambarkan kolaborasi dinamis antara sejumlah *object*. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*. Sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

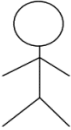
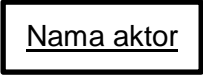

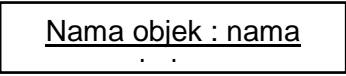
*Sequence Diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan message yang dikirimkan dan diterima antar objek. Menggambar diagram sekuen harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya


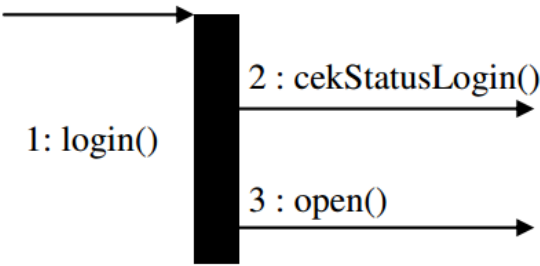
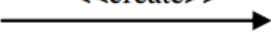

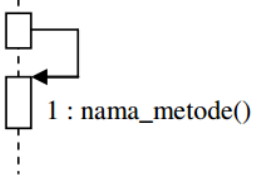
pesan sudah dicakup dalam diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

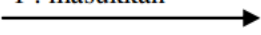
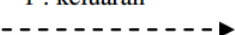
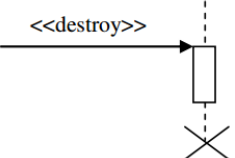
Dalam *sequence* diagram, setiap *object* hanya memiliki garis yang digambarkan garis putus-putus ke bawah. Pesan antar *object* digambarkan dengan anak panah dari *object* yang mengirimkan pesan ke *object* yang menerima pesan.

Berikut adalah simbol-simbol yang ada pada diagram sekuen.

**Tabel 2.4** Simbol-simbol *Sequence* Diagram

No	Simbol	Deskripsi
1.	<p>Aktor</p>  <p>Atau</p>  <p>Tanpa waktu aktif</p>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dibuat diluar sistem yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda diawal frase nama aktor.
2.	<p>Garis hidup/<i>lifeline</i></p> 	Menyatakan kehidupan suatu objek
3.	<p>Objek</p> 	Menyatakan objek yang berinteraksi pesan

No	Simbol	Deskripsi
4.	Waktu aktif 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>  <p>Maka cekStatusLogin() dan open() dilakukan didalam metode login(). Aktor tidak memiliki waktu aktif.</p>
5.	Pesan tipe <i>create</i> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>
6.	Pesan tipe <i>call</i> 	 <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.</p>

No	Simbol	Deskripsi
7.	Pesan tipe <i>send</i> 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	Pesan tipe <i>return</i> 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
9.	Pesan tipe <i>destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>destroy</i> .

Sumber: (Sukamto & Shalahuddin, 2014).

## 2.11 Pengujian *Black box*

Metode *black box* yaitu data pengujian dipilih berdasarkan spesifikasi masalah tanpa memperhatikan detail internal dari program (Sukamto, 2009), untuk memeriksa apakah program dapat berjalan dengan benar. Pemilihan data pengujian paling tidak harus dipilih dengan cara berikut (Sukamto, 2009).

- *Extreme values*, banyak program *error* pada suatu batas range dari aplikasi.
- *Illegal values*, yaitu suatu data atau nilai yang tidak diperbolehkan maupun data yang tidak berguna.
- *Values*, yaitu data yang mudah diperiksa.
- *Typical realistic value*, yaitu mencoba program dengan data pengujian untuk melihat bagaimana program melakukannya. Data ini harus cukup sederhana sehingga hasilnya dapat dihitung secara manual.

Menurut Sukamto (2009), terdapat beberapa teknik pengujian secara *black box*, adapun teknik pengujian tersebut adalah sebagai berikut.

- a. *Boundary Value Analyst (BVA)/Limit Testing* adalah banyak kesalahan terjadi pada kesalahan masukan. BVA mengijinkan untuk menyeleksi kasus uji yang menguji batasan nilai input. BVA merupakan komplemen dari equivalence partitioning. Lebih pada memilih elemen-elemen di dalam kelas ekivalen pada bagian sisi batas dari kelas.
- b. *Comparison Testing* adalah pengujian pada beberapa aplikasi reliability dari sebuah perangkat lunak. Redundansi perangkat keras dan perangkat lunak mungkin digunakan untuk meminimalisir kesalahan (*error*). Untuk redundansi perangkat lunak, gunakan tim yang terpisah untuk mengembangkan setiap versi perangkat lunak yang independen. Uji setiap versi dengan data yang sama untuk memastikan semua versi menghasilkan keluaran yang sama. Jalankan semua versi dengan paralel dan perbandingan keluaran secara real-time.
- c. *Sample Testing* adalah pengujian yang melibatkan beberapa nilai yang terpilih dari sebuah kelas ekivalen. Mengintegrasikan nilai pada kasus uji dan nilai-nilai yang terpilih mungkin dipilih dengan urutan tertentu atau interval tertentu.
- d. *Robustness Testing* adalah pengujian dengan data input dipilih diluar spesifikasi yang telah didefinisikan. Tujuan dari pengujian ini adalah membuktikan bahwa tidak ada kesalahan jika masukan tidak valid.
- e. *Behavior Testing* adalah pengujian dengan hasil uji tidak dapat dievaluasi jika hanya melakukan pengujian sekali, tapi dapat dievaluasi jika pengujian dilakukan beberapa kali, misalnya pada pengujian struktur data stack.
- f. *Requirement Testing* adalah spesifikasi kebutuhan yang terasosiasi dengan perangkat lunak (input, output, fungsi, performansi) diidentifikasi pada tahap spesifikasi kebutuhan dan desain. *Requirement Testing* melibatkan pembuatan kasus uji untuk setiap spesifikasi kebutuhan yang terkait dengan program.
- g. *Performance Testing* adalah mengevaluasi kemampuan program untuk beroperasi dengan benar dipandang dari sisi acuan kebutuhan misalnya: aliran data, ukuran pemakaian memori, kecepatan eksekusi dan lain-lain. Untuk mencari tahu beban kerja atau kondisi konfigurasi program dan dapat digunakan untuk menguji batasan lingkungan program.

- h. Uji Ketahanan (*Endurance Testing*) adalah melibatkan kasus uji yang diulang-ulang dengan jumlah tertentu dengan tujuan untuk mengevaluasi program apakah sesuai dengan spesifikasi kebutuhan.
- i. *Equivalence Partitioning* adalah membagi input menjadi kelas-kelas data yang dapat digunakan untuk menggenerasi kasus uji dengan tujuan untuk menemukan kelas-kelas kesalahan. Selain itu, *equivalence partitioning* berdasarkan pada kesamaan kelas-kelas kondisi input. Sebuah kelas yang ekuivalen merepresentasikan kumpulan status/kondisi yang valid atau tidak valid. Sebuah kondisi input dapat berupa nilai numerik yang spesifik, rentang nilai, kumpulan nilai yang berkaitan, atau kondisi boolean.
- j. Uji Sebab-Akibat (*Cause-Effect Relationship Testing*) adalah teknik yang merupakan suplemen dari *equivalence Testing* dengan menyediakan cara untuk memilih kombinasi data input dan melibatkan kondisi input (*cause*) dan kondisi output (*effect*) untuk mencegah pendefinisian kasus uji yang terlalu banyak.
- k. *Acceptance Testing* adalah pengujian terakhir sebelum sistem dipakai oleh user yang melibatkan pengujian dengan data dari pengguna sistem. *User acceptance test* adalah uji terima perangkat lunak yang dilakukan ditempat pengguna oleh user perangkat lunak tersebut. Tujuan pengujian ini adalah untuk menguji apakah sistem sudah sesuai dengan apa yang tertuang dalam spesifikasi fungsional sistem (*validation*).

## 2.12 BitBar Testing

BitBar *Testing* dibuat oleh perusahaan BitBar Technologies Limited, yang menyediakan *automation Testing* dan *manual Testing* menggunakan perangkat Android dan iOS asli. Bitbar *Testing* menyediakan alat pengujian berbasis cloud yang fleksibel. Pengujian dengan menggunakan Bitbar *Testing* mempunyai kelebihan konkurensi tidak terbatas sehingga dapat melakukan pengujian perangkat dengan berbagai jumlah perangkat, dari sepuluh sampai seribu, dan secara paralel menggunakan perangkat dalam jumlah besar yang tersedia online.

Hasil pengujian didapatkan secara cepat dengan semua data, logs, videos, *screenshots*, serta grafik CPU dan *memory usage* hasil pengujian, sehingga pengembang dapat mendeteksi dan menyelesaikan permasalahan lebih cepat dan



sesuai waktu untuk menghasilkan aplikasi yang lebih baik. Pada penelitian ini Bitbar *Testing* digunakan sebagai pengujian aplikasi dari segi fungsionalitas, kompatibilitas, dan performa.

Pada segi fungsionalitas dilakukan pengujian terhadap fungsi-fungsi atau *methods* yang terdapat pada aplikasi berjalan tanpa *error* atau tidak seperti saat dilakukan pengiriman data, dan sebagainya. Pada segi kompatibilitas dilakukan pengujian aplikasi yang dipasang pada beberapa *devices* yang dipilih oleh pengguna pada sistem BitBar. Pada segi performa dilakukan pengujian terhadap penggunaan CPU dan Memori pada setiap *devices* yang telah dipilih.

Metode ini lebih efektif dan mendalam karena pengujian ini mencakup pengujian *black box* yaitu fungsionalitas, ditambah dengan pengujian kompatibilitas, dan performa yang dilakukan secara cepat dan dalam beberapa perangkat sekaligus.

### 2.13 Pengujian Kuesioner

Kuesioner atau angket merupakan suatu daftar pertanyaan atau pernyataan tentang topik tertentu yang diberikan kepada subyek, baik secara individual atau kelompok untuk mendapatkan informasi tertentu. Angket dibedakan menjadi dua jenis yaitu angket terbuka dan angket tertutup. Angket terbuka yaitu angket yang disajikan dalam bentuk sederhana sehingga responden dapat memberikan isian sesuai dengan kehendak dan keadaannya. Sedangkan angket tertutup ialah angket yang disajikan dalam bentuk sedemikian rupa sehingga responden diminta untuk memilih satu jawaban yang sesuai dengan karakteristik dirinya dengan cara memberikan tanda silang atau tanda *checklist* ( Nasution, 2006).

Dalam proses pembuatan kuesioner yang perlu dilakukan yaitu :

- Konsep
- Dari konsep itu dibuat definisi operasionalnya
- Mengembangkan definisi operasional menjadi dimensi dimensi, sebagai petunjuk untuk menentukan jumlah indicator atau item-item pertanyaan.
- Merumuskan ke dalam item-item pertanyaan atau pernyataan.
- Menentukan jenjang skor.
- Melakukan seleksi item.

### 2.14 Kajian Terkait

Dalam penelitian ini beberapa pustaka sebagai bahan referensi dan acuan. Beberapa penelitian diantaranya:

**Tabel 2.5** Tabel Perbandingan Penelitian

No	Penulis	Judul	Keterangan
1.	Muhammad Himmi Syahrul Abidin dan Yusriel Ardian (2013), Universitas Kanjuruhan	Rancang Bangun Aplikasi Monitoring <i>Network</i> Berbasis Web Menggunakan HTML5 pada Dinas Pendidikan Kabupaten Blitar	<ul style="list-style-type: none"> <li>- Aplikasi ini dibangun menggunakan HTML 5 berbasis web.</li> <li>- Melakukan monitoring jaringan menggunakan protocol SNMP.</li> <li>- Menggunakan pengujian <i>black box</i></li> </ul>
2.	Andi Amrurrahman Maula (2015), Universitas Islam Negeri Syarif Hidayatullah Jakarta	Aplikasi Monitoring Jaringan Berbasis <i>Mobile</i> Web dengan Sistem Notifikasi Berbasis SMS Gateway	<ul style="list-style-type: none"> <li>- Aplikasi dibangun berbasis web <i>mobile</i>.</li> <li>- Dibangun untuk Sekolah Islam Al-Azhar Bumi Serpong Damai.</li> <li>- Menggunakan sistem notifikasi berbasis sms gateway</li> </ul>
3.	Zahria Ihram Gavitra (2018), Universitas Islam Indonesia	Aplikasi Remote Mikrotik Berbasis Android	<ul style="list-style-type: none"> <li>- Aplikasi berbasis android <i>mobile</i>.</li> <li>- Melakukan remote pada mikrotik untuk melakukan konfigurasi mikrotik.</li> </ul>

**Tabel 2.6** Tabel Penelitian yang Akan Dilakukan

No	Penulis	Judul	Keterangan
1.	Iqbal Tri Hartanto (2019), Universitas Tanjungpura Pontianak	Rancang Bangun Aplikasi Manajemen Jaringan pada Mikrotik Berbasis Android	<ul style="list-style-type: none"> <li>- Aplikasi dibangun berbasis android.</li> <li>- Dibuat untuk Lab Informatika Untan.</li> <li>- Melakukan onitoring besaran <i>bandwith</i> pada <i>client</i></li> <li>- Melakukan remote pada mikrotik untuk memutus koneksi <i>client</i></li> </ul>

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Bahan Penelitian**

Bahan penelitian yang digunakan berupa data dari *router mikrotik* yang ada di Lab Informatika unta, yaitu data yang dibutuhkan untuk pembuatan aplikasi manajemen jaringan mikrotik.

#### **3.2 Alat yang Digunakan**

Adapun alat yang digunakan yaitu alat penelitian, perangkat lunak dan perangkat keras.

##### **3.2.1 Alat Penelitian**

Alat penelitian yang digunakan dalam penelitian ini adalah:

1. Diagram alir sistem, digunakan untuk menggambarkan aliran data dari sebuah proses atau sistem.
2. Unified Modelling Language (UML). UML digunakan untuk menggambarkan batasan sistem dan fungsi-fungsi sistem secara umum.
3. Use Case Diagram, untuk mengetahui fungsi apa saja yang ada di dalam sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

##### **3.2.2 Perangkat Keras**

Perangkat keras yang digunakan untuk merancang aplikasi dalam penelitian ini adalah:

1. Satu unit laptop Asus VivoBook A442U, dengan spesifikasi Intel Core i5-8250U, up to 3.4GHz, 8.00GB RAM DDR4, Harddisk 1000GB.
2. Smartphone Realme 3Pro dengan spesifikasi Qualcomm Snapdragon 712 Octa-Core, 128GB internal memory, 6GB RAM, Android 9.
3. MikroTik Routerboard RB941-2nD-TC.

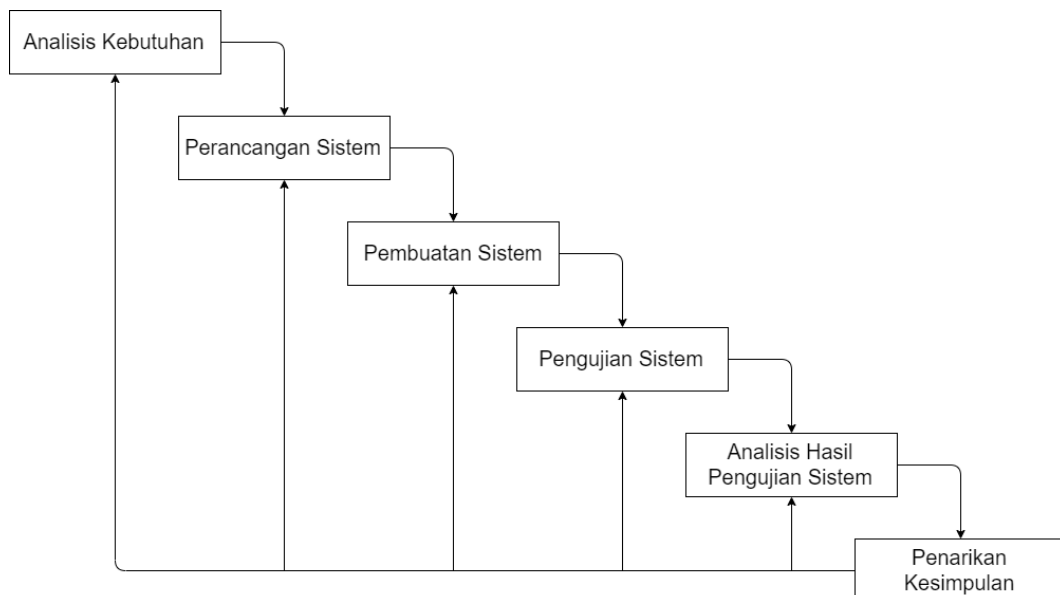
### 3.2.3 Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian ini adalah:

1. Sistem Operasi Ubuntu 16.04.6
2. Android Studio 3.2.1 sebagai aplikasi untuk mengembangkan aplikasi platform Android.
3. Visual Studio Code sebagai aplikasi untuk text editor.
4. phpMyAdmin 4.6.5.2 sebagai pengelola basis data MySQL.
5. MySQL 5.14.1 sebagai pengelola sistem database.
6. PHP Vesi 7.1.1 sebagai bahasa pemrograman yang digunakan.
7. Mozilla Firefox sebagai browser untuk menjalankan aplikasi.
8. Postman version 7.0.4 sebagai aplikasi untuk menguji coba REST API.
9. Menggunakan tools RouterOSClient dan RouterOSQuery

### 3.3 Metode Penelitian

Metode penelitian merupakan beberapa tahapan yang dirancang dan dijadikan sebagai panduan dalam melakukan penelitian. Pada penelitian ini menggunakan metode *Waterfall* dalam membangun sistem. Tahapan pada penelitian yang akan dilakukan dijelaskan pada Gambar 3.1 sebagai berikut.



**Gambar 3.1** Diagram Alir Penelitian

### **3.3.1 Analisis Kebutuhan**

Analisis kebutuhan dilakukan dengan memahami masalah dan menganalisa masalah. Analisa kebutuhan sistem dilakukan dengan admin atau aslab jaringan yang ada di Lab Informatika Untan.

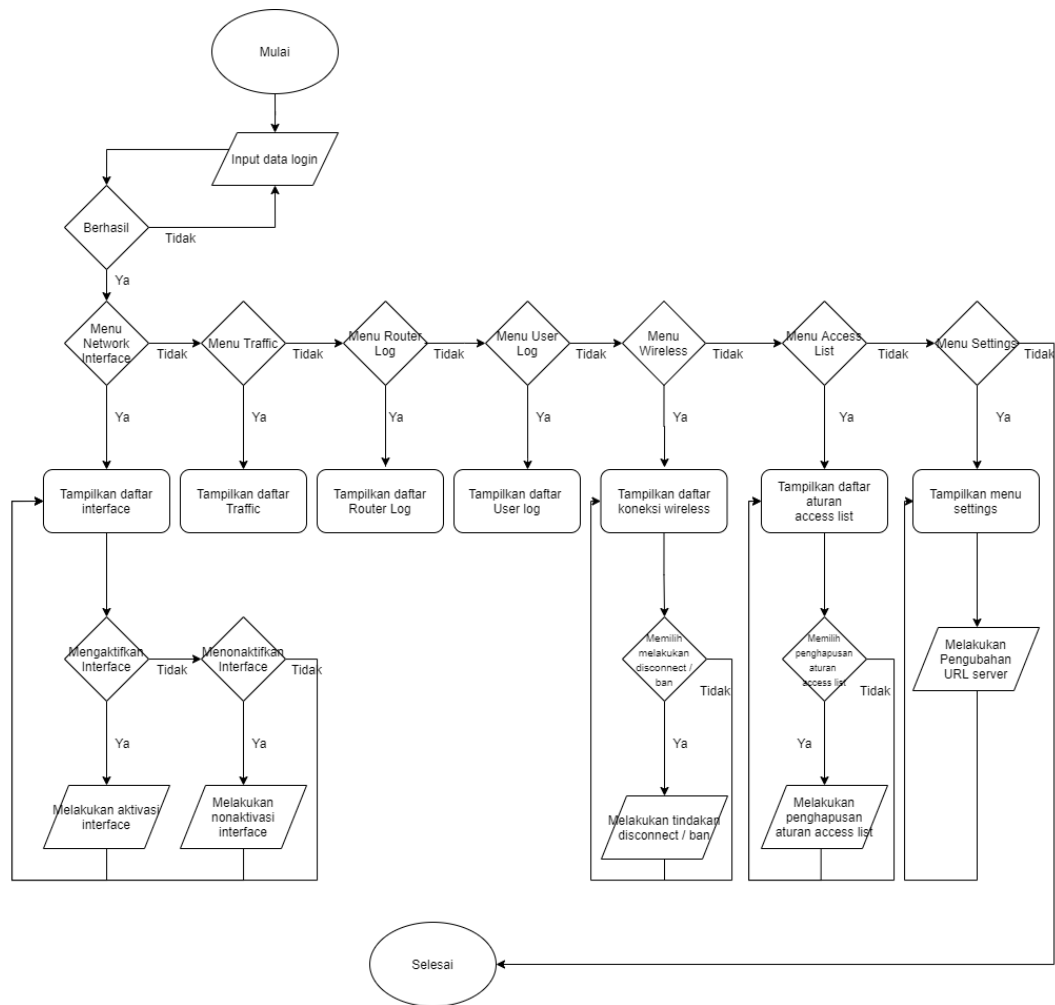
Aplikasi Manajemen jaringan yang akan dibuat akan dirancang sesuai dengan kebutuhan. Pada aplikasi ini akan memudahkan admin atau aslab jaringan untuk melakukan pemantauan *bandwith* pengguna, memutuskan koneksi jaringan pada pengguna, melihat *log history*, dan memantau berapa banyak pengguna yang sedang terhubung.

### **3.3.2 Perancangan Sistem**

Dalam melakukan perancangan sistem terdapat tahapan-tahapan yang dilakukan yaitu:

#### **3.3.2.1 Perancangan Diagram Alir Sistem**

Diagram alir sistem merupakan diagram yang menggambarkan sistem yang akan dibangun. Adapun diagram alir sistem yang akan dibangun dapat dilihat pada Gambar 3.2 dan Gambar 3.3 berikut.

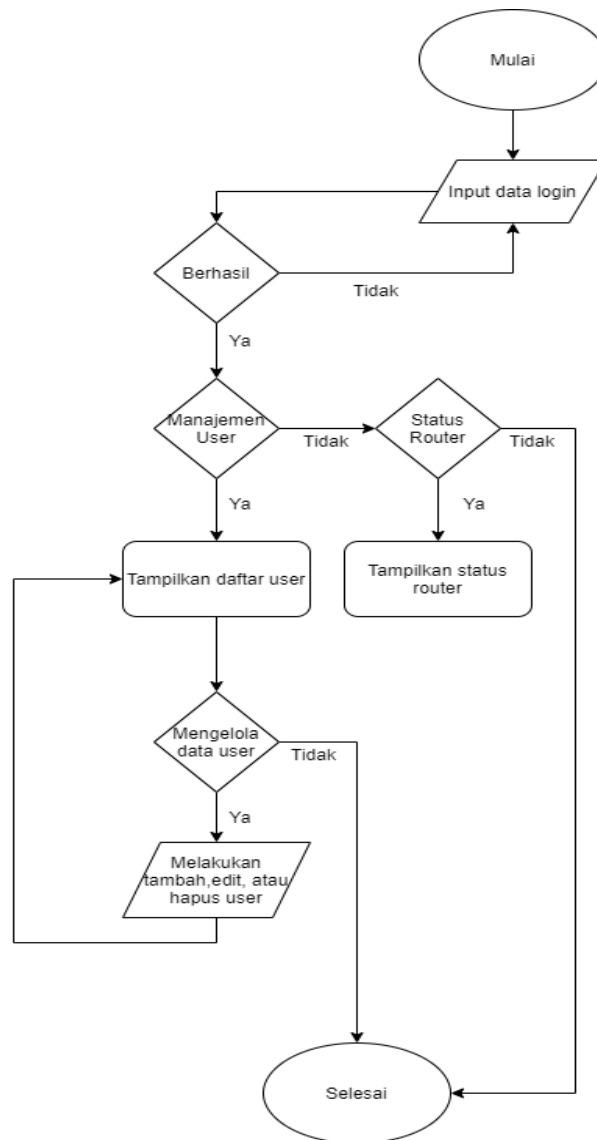


**Gambar 3.2** Diagram Alir Sistem Aplikasi Android

Administratur jaringan menjalankan aplikasi manajemen jaringan mikrotik dengan melakukan *login* dan jika berhasil maka akan masuk ke aplikasi, kemudian akan menampilkan data dari *network interace* yaitu daftar *client* yang terhubung dari port pada mikrotik. Kemudian jika administrator jaringan ingin mengecek *log* atau *history* dari aplikasi administrator jaringan dapat *swipe* ke kiri untuk masuk ke halaman *log* aplikasi. Untuk mengecek *client* yang terhubung pada *wireless* mikrotik administrator jaringan dapat pindah ke halaman *wireless* dengan *swipe* ke kiri atau menekan tulisan *WIRELESS*.

Administrator jaringan dapat memutuskan koneksi pada *client* yang terhubung pada mikrotik dengan menekan tombol *disable* pada *neteork interface* dan tombol *disconect/ban* untuk *wireless*. Untuk masuk ke halaman *ban list* administrator jaringan bisa menekan tulisan *ban list* pada aplikasi atau *swipe* ke kiri

untuk berpindah halaman. Jika administrator jaringan ingin mengubah URL *server* maka administrator jaringan tinggal masuk pada halaman *settings* dengan menekan tulisan *settings* atau *swipe* ke kiri untuk berpindah kehalaman *settings*, jika sudah masuk pada halaman *settings* administrator jaringan dapat mengubah URL *server* dengan mengubah alamat ip kemudian menekan tombol *save settings* maka URL *server* akan berubah.



**Gambar 3.3** Diagram Alir Sistem Website

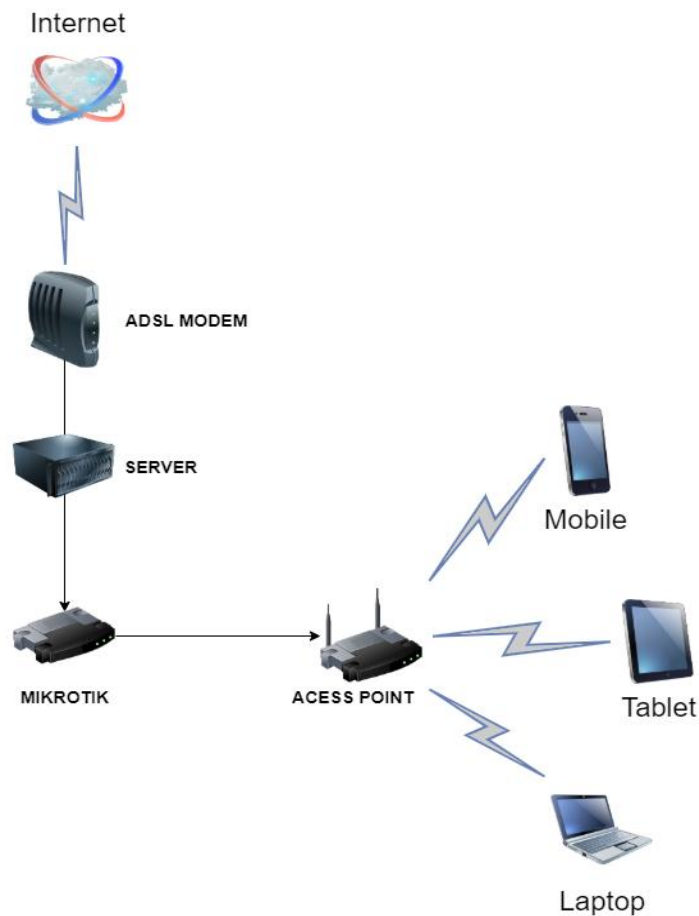
Super admin masuk ke website dengan melakukan *login* terlebih dahulu dan jika berhasil maka akan masuk pada website dan menampilkan halaman manajemen user yaitu menampilkan daftar data user atau hak akses yang dapat *login* ke aplikasi dan website dari manajemen mikrotik. Super admin dapat menambah,



mengubah, dan menghapus data *user*. Kemudian jika super admin ingin melihat status *user* maka super admin tinggal menekan tombol status *router* maka akan menampilkan halaman status *router*.

### 3.3.2.2 Topologi jaringan

Secara umum gambaran topologi jaringan yang akan digunakan seperti di gambar 3.3.

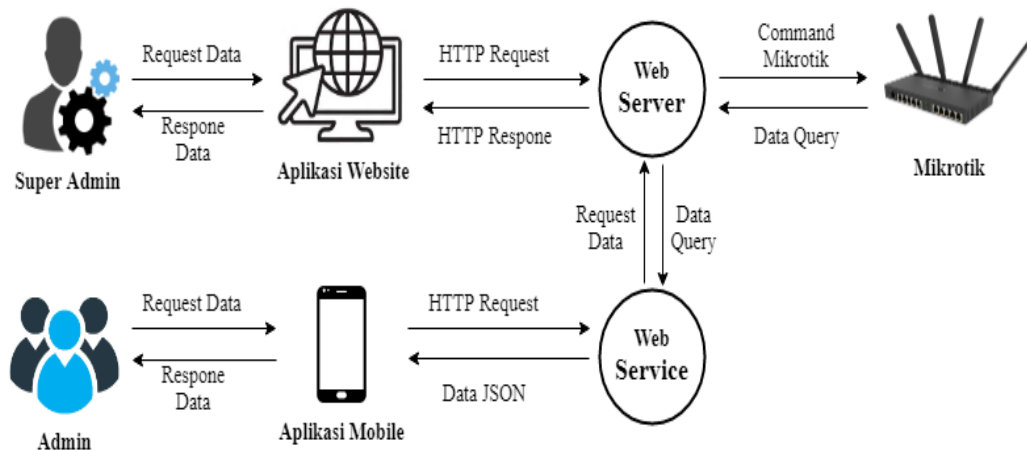


**Gambar 3.4** Topologi Jaringan

Berdasarkan topologi jaringan pada gambar 3.3 proses yang terjadi adalah dimulai dari sumber internet yang masuk ke AD SL Modem yaitu UPT TIK Untan, kemudian disalurkan ke *server* yang ada di Lab Informatika Untan dan diteruskan ke *mikrotik* kemudian disebarakan menggunakan *access point* ke pengguna.

### 3.3.2.3 Perancangan Arsitektur Sistem

Secara umum gambaran sistem dapat dibuat rancangan awal arsitektur sistem seperti Gambar 3.4.



**Gambar 3.5** Arsitektur Sistem

Berdasarkan arsitektur sistem pada Gambar 3.3 Proses yang terjadi adalah: Sistem yang dibangun memiliki dua aktor yaitu super admin dan admin. Super admin adalah kepala lab jaringan informatika untan memiliki akses untuk mengatur siapa saja yang bisa mengakses aplikasi web dan aplikasi *mobile* yang dibuat. Pada aplikasi web admin hanya dapat melihat status *router* dan tidak bisa membuka halaman manajemen *user*, sedangkan super admin dapat melihat status *router* dan dapat mengakses halaman manajemen *user*. Pada halaman menejemen *user* yang ada pada aplikasi web super admin dapat mengolah data hak akses dari admin atau siapa saja yang punya hak akses masuk pada aplikasi manajemen jaringan mikrotik, kemudian aplikasi web akan mengirim *HTTP request* ke server kemudian server yang terhubung dengan *router mikrotik* menggunakan *socket TCP* di port 8287 akan memberi comand pada *mikrotik*, kemudian *router mikrotik* akan membalas dengan memberikan data yang di minta dan kemudian dikembalikan ke web oleh server kemudian web akan menampilkan data yang didapat. Pada aplikasi *mobile android* admin memberikan *request* ke aplikasi *mobile* kemudian aplikasi *mobile* akan mengirim *HTTP request* ke web service kemudian akan diteruskan oleh web service ke server yang terhubung dengan *router mikrotik* menggunakan *socket TCP* di port 8287 dan kemudian server akan memberikan *comand* pada *mikrotik* kemudian

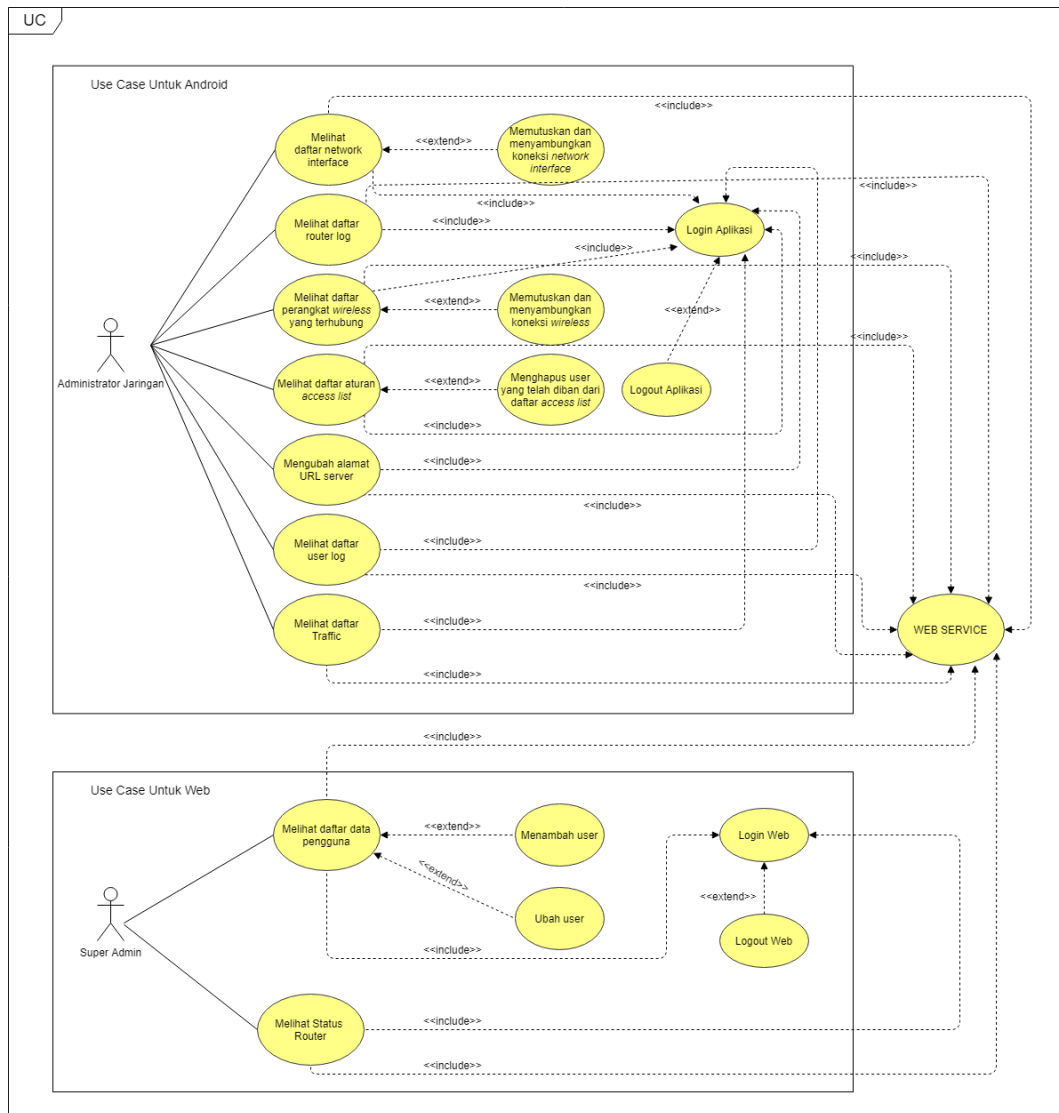
*mikrotik* akan membalas dengan mengirim data yang diminta ke *server*, *server* akan mengirim data yang diperoleh dari *mikrotik* ke *web service* dan akan diubah data yang diperoleh yang berupa data *query* ke data *json* kemudian akan dikirim ke aplikasi android.

#### **3.3.2.4 Perancangan Use Case Diagram**

Diagram use case merupakan pemodelan untuk kelakuan (behaviour) sistem yang akan dibuat. Diagram ini digunakan untuk mendeskripsikan interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Use case digunakan untuk mengetahui fungsi apa yang sudah ada didalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

a. **Diagram Use Case**

Berikut ini adalah diagram use case dari aplikasi yang dapat dilihat pada Gambar 3.5.



**Gambar 3.6** Diagram Use Case

b. Definisi Aktor

Berikut adalah deskripsi pendefinisian aktor pada sistem layanan pengaduan masyarakat.

**Tabel 3.1** Definisi Aktor

No.	Aktor	Deskripsi
1.	Administrator Jaringan	Admin dalam sistem ini adalah aslab jaringan di lab informatika untan.
2.	Super Admin	Super admin dalam sistem ini adalah kepala lab jaringan di lab informatika untan

c. Definisi Use Case

Berikut adalah deskripsi pendefinisian use case untuk Android dan Web yang digunakan oleh administrator jaringan dan super admin.

**Tabel 3.2** Definisi Use Case untuk Android

No.	Use case	Deskripsi
UC01	<i>Login Aplikasi Mobile</i>	Merupakan proses melakukan <i>login</i> bagi pengguna pada aplikasi <i>mobile</i>
UC02	<i>Logout Aplikasi Mobile</i>	Merupakan proses melakukan <i>logout</i> bagi pengguna pada aplikasi <i>mobile</i>
UC03	Melihat daftar <i>network interface</i>	Merupakan proses menampilkan data <i>client</i> yang terhubung pada <i>network interface</i>
UC04	Memutuskan koneksi <i>network interface</i>	Merupakan proses untuk mematikan atau <i>disable</i> yang ada di <i>network interface</i>
UC05	Melihat daftar <i>Router log</i>	Merupakan proses menampilkan data <i>log</i> atau history dari <i>router</i>
UC06	Melihat daftar <i>wireless</i> yang terhubung	Merupakan proses menampilkan data <i>client</i> yang terhubung pada <i>wireless</i>
UC07	Memutuskan koneksi <i>wireless</i>	Merupakan proses untuk memutuskan atau <i>ban client</i> yang tersambung pada <i>wireless</i>
UC08	Melihat daftar aturan <i>ban list</i>	Merupakan proses melihat data daftar <i>client</i> yang di <i>ban</i>
UC09	Menghapus <i>user</i> yang <i>banned</i> pada <i>ban list</i>	Merupakan proses untuk menghapus atau mengembalikan <i>user</i> yang <i>baned</i> agar dapat terhubung ke mikrotik kembali
UC10	Mengubah alamat URL server	Merupakan proses mengubah alamat URL server
UC11	Melihat daftar <i>user log</i>	Merupakan proses menampilkan data <i>log</i> atau history dari aplikasi
UC12	Melihat daftar traffic	Merupakan proses menampilkan data <i>traffic</i> atau kecepatan <i>upload</i> dan <i>download</i> dari mikrotik
UC13	<i>Login Web</i>	Merupakan proses melakukan <i>login</i> bagi

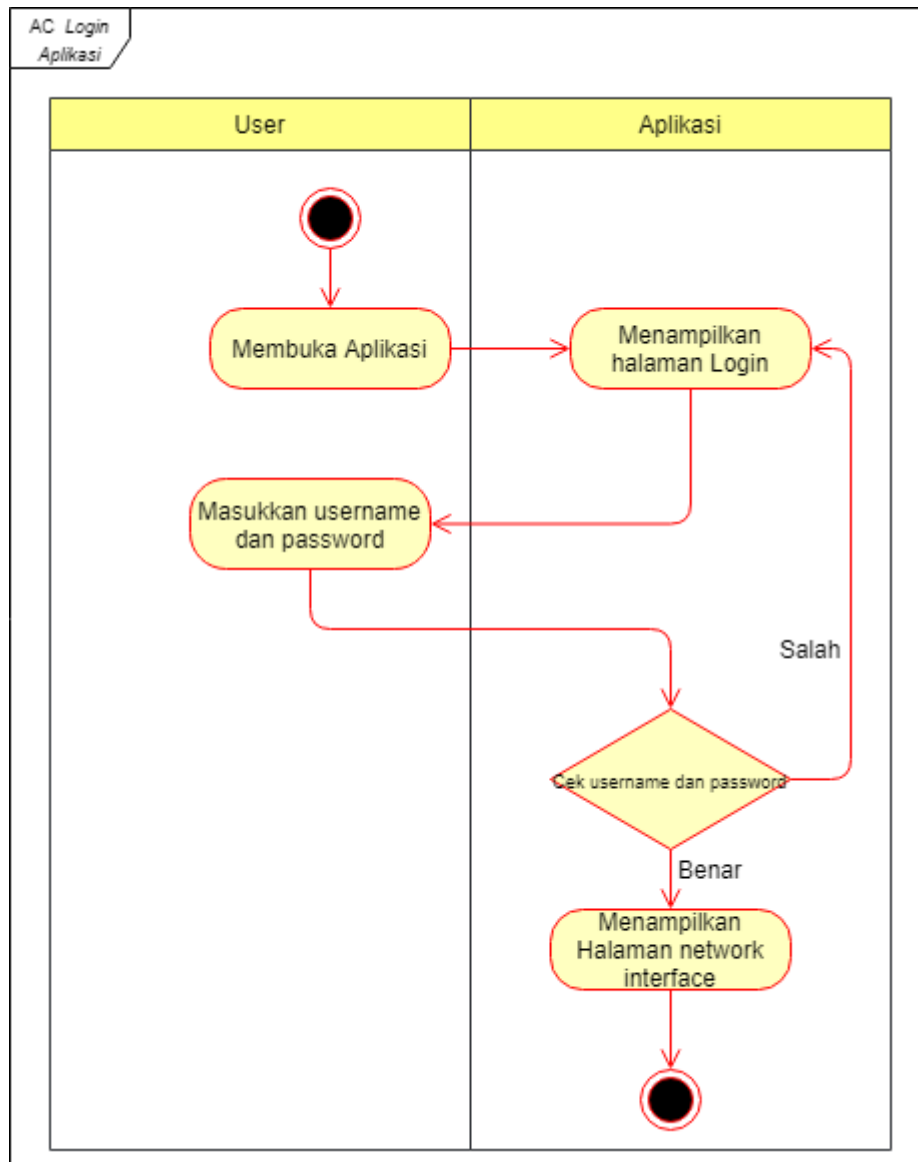
No.	Use case	Deskripsi
		pengguna pada <i>web</i>
UC14	<i>Logout Web</i>	Merupakan proses melakukan <i>logout</i> bagi pengguna pada <i>web</i>
UC15	Melihat daftar data pengguna	Merupakan proses menampilkan data pengguna yang dapat mengakses aplikasi manajemen jaringan dan <i>web</i>
UC16	Menambah <i>User</i>	Merupakan proses menambah data pengguna yang dapat mengakses aplikasi manajemen jaringan dan <i>web</i>
UC17	Ubah <i>User</i>	Merupakan proses mengubah data dari user yang telah ada
UC18	Melihat status <i>router</i>	Merupakan proses menampilkan status <i>router</i>

### 3.3.2.5 Perancangan Activity Diagram

Activity diagram menggambarkan alur kerja atau aktivitas pada sistem yang bertujuan untuk melihat alur proses sistem secara bertahap. Activity diagram dibuat berdasarkan sebuah atau beberapa use case pada use case diagram.

#### a. Activity Diagram Login Aplikasi Mobile

Berikut adalah activity diagram Login Aplikasi Mobile dapat dilihat pada Gambar 3.6.

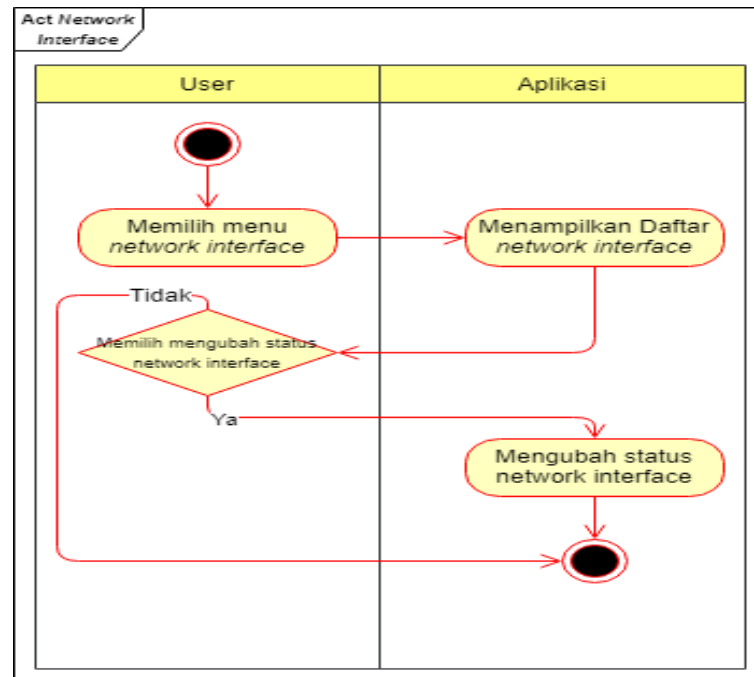


**Gambar 3.7** Activity Diagram Login Aplikasi

Pada Gambar 3.6, dapat dilihat bahwa pada saat login, Admin memasukkan username dan password, kemudian aplikasi akan mengecek jika username dan password benar maka aplikasi akan menampilkan halaman network interface, jika salah maka akan kembali menampilkan halaman login.

b. Activity Diagram Network Interface

Berikut adalah activity diagram network interface dapat dilihat pada Gambar 3.7.

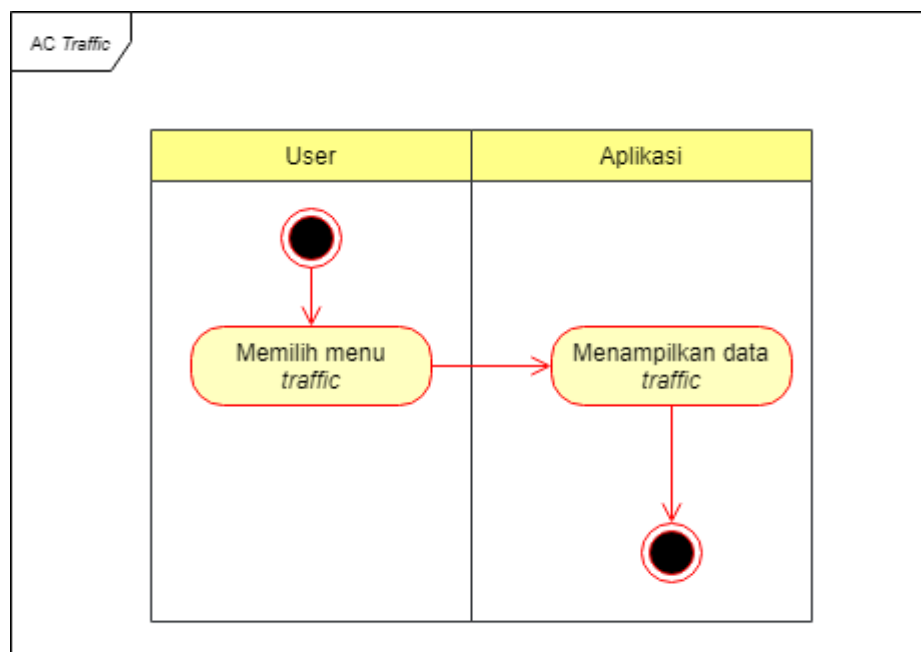


**Gambar 3.8** Activity Diagram Network Interface

Pada Gambar 3.7, dapat dilihat bahwa pada halaman network interface pengguna dapat melihat daftar network interface dan dapat mengubah status menjadi disable atau dimatikan koneksinya dari mikrotik.

c. Activity Diagram Traffic

Berikut adalah activity diagram traffic dapat dilihat pada Gambar 3.8.



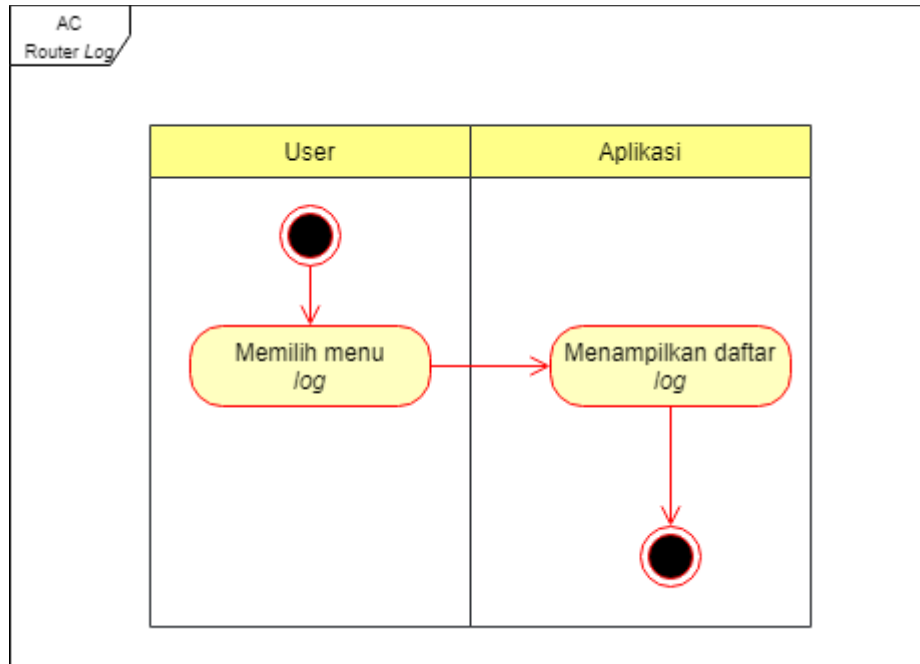
**Gambar 3.9** Activity Diagram Traffic



Pada Gambar 3.8, dapat dilihat bahwa pada halaman traffic pengguna dapat melihat kecepatan upload dan download dari router mikrotik.

d. Activity Diagram Router Log

Berikut adalah activity diagram router log dapat dilihat pada Gambar 3.9.

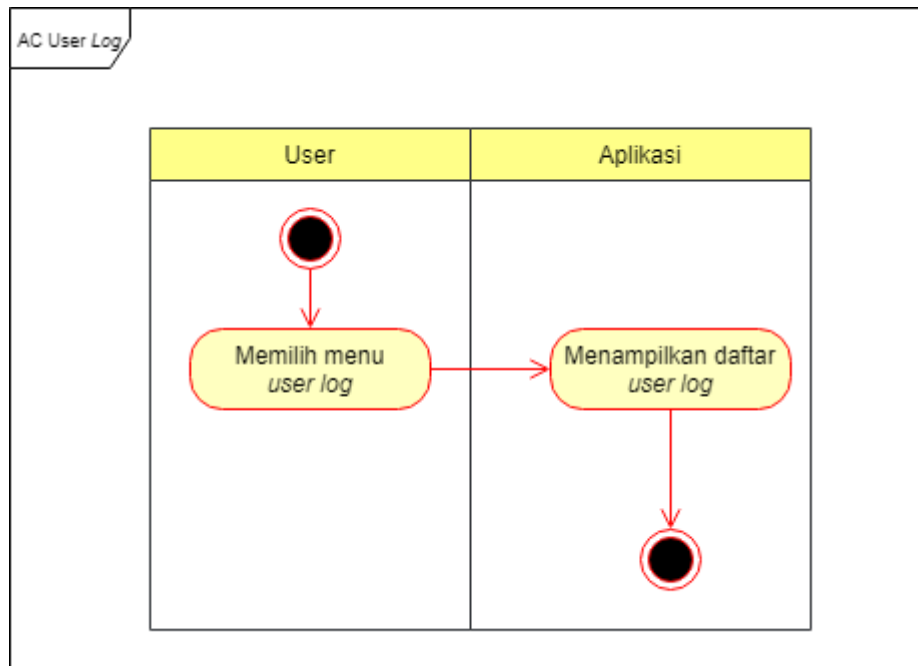


**Gambar 3.10** Activity Diagram Router Log

Pada Gambar 3.9, dapat dilihat bahwa pada halaman router log pengguna dapat melihat history dari router mikrotik.

e. Activity Diagram User Log

Berikut adalah activity diagram user log dapat dilihat pada Gambar 3.10.

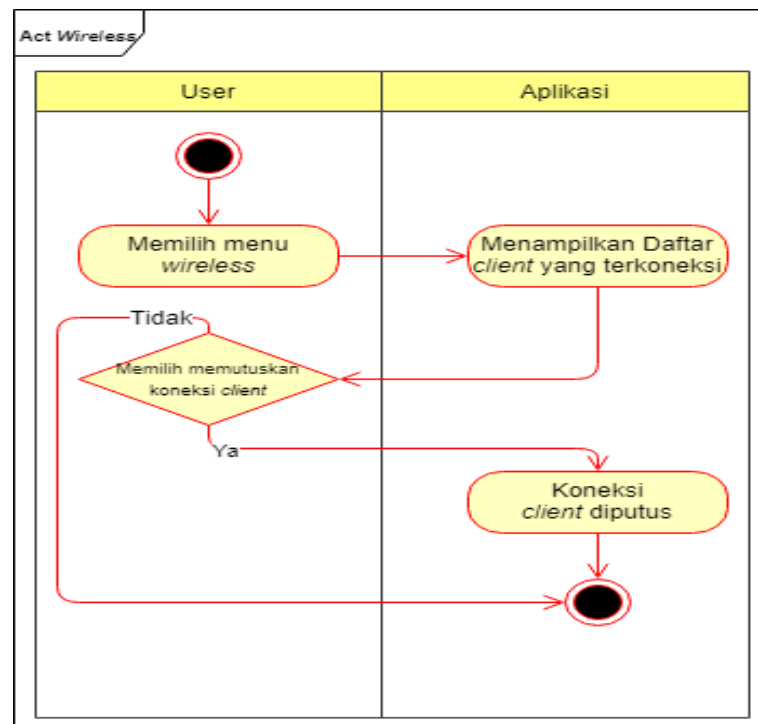


**Gambar 3.11** Activity Diagram User Log

Pada Gambar 3.10, dapat dilihat bahwa pada halaman user log pengguna dapat melihat history dari aplikasi manajemen jaringan mikrotik.

f. Activity Diagram Wireless

Berikut adalah activity diagram wireless dapat dilihat pada Gambar 3.11.

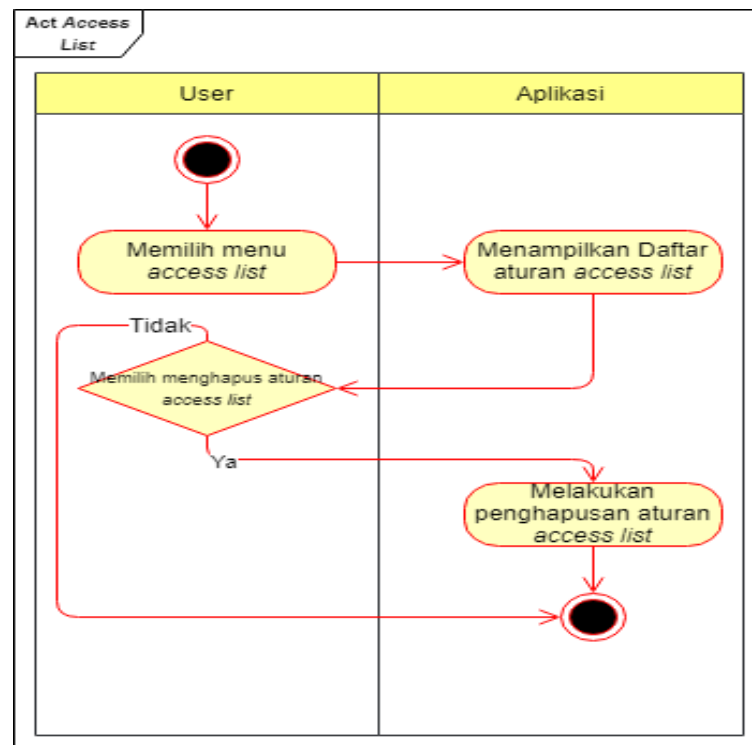


**Gambar 3.12** Activity Diagram Wireless

Pada Gambar 3.11, dapat dilihat bahwa pada halaman wireless pengguna dapat melihat daftar client yang terkoneksi ke mikrotik melalui wireless, dan pengguna juga dapat memutuskan koneksi client.

g. Activity Diagram Ban List

Berikut adalah activity diagram *ban list* dapat dilihat pada Gambar 3.12.

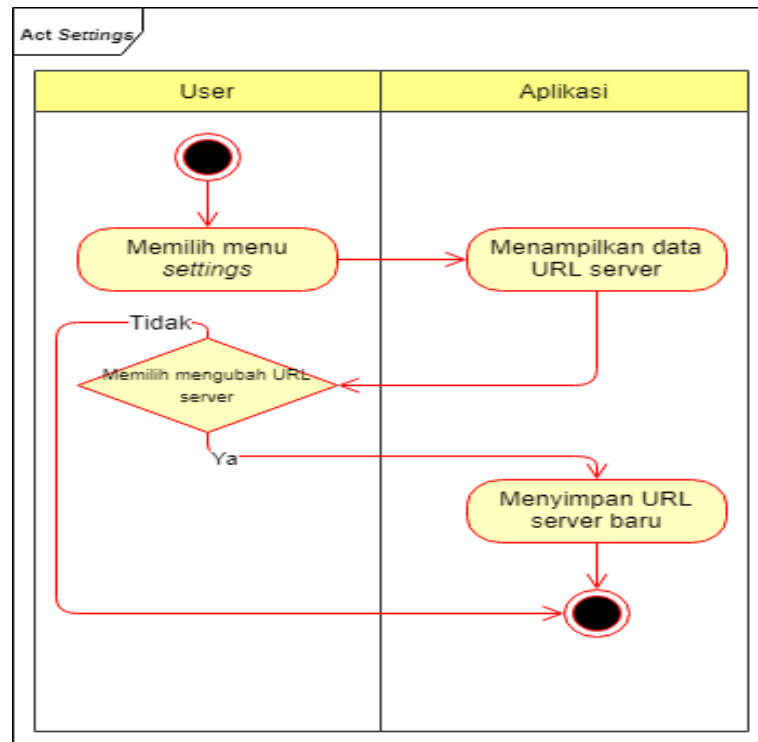


**Gambar 3.13** Activity Diagram Ban List

Pada Gambar 3.12, dapat dilihat bahwa pada halaman *ban list* pengguna dapat melihat daftar *user* atau *client* yang disable atau dibanned, dan pengguna juga dapat mengubah status atau menghapus user dari daftar *ban list*.

h. Activity Diagram Settings

Berikut adalah activity diagram settings dapat dilihat pada Gambar 3.13.

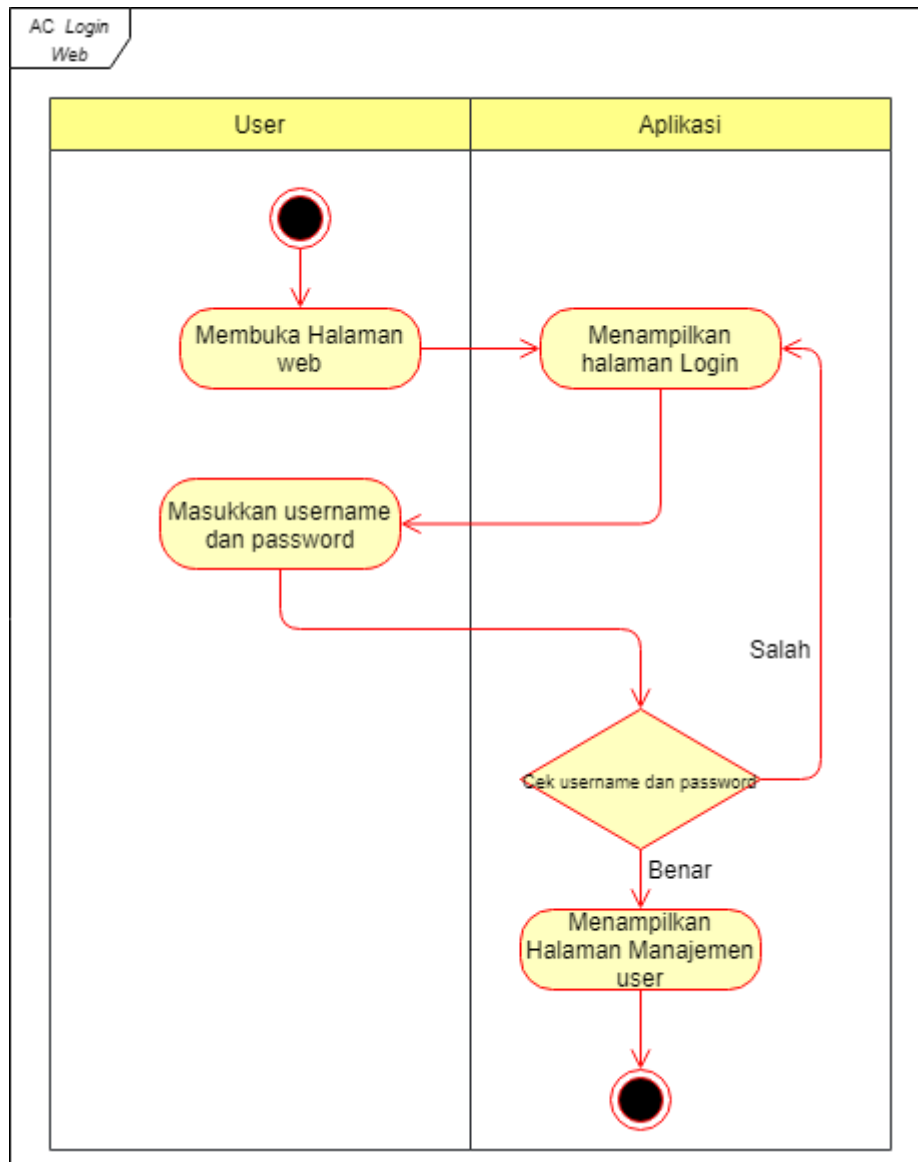


**Gambar 3.14** Activity Diagram Settings

Pada Gambar 3.13, dapat dilihat bahwa pada halaman settings pengguna dapat melihat alamat URL server dan dapat mengubah ke alamat URL server baru.

i. Activity Diagram Login Web

Berikut adalah activity diagram login web dapat dilihat pada Gambar 3.14.

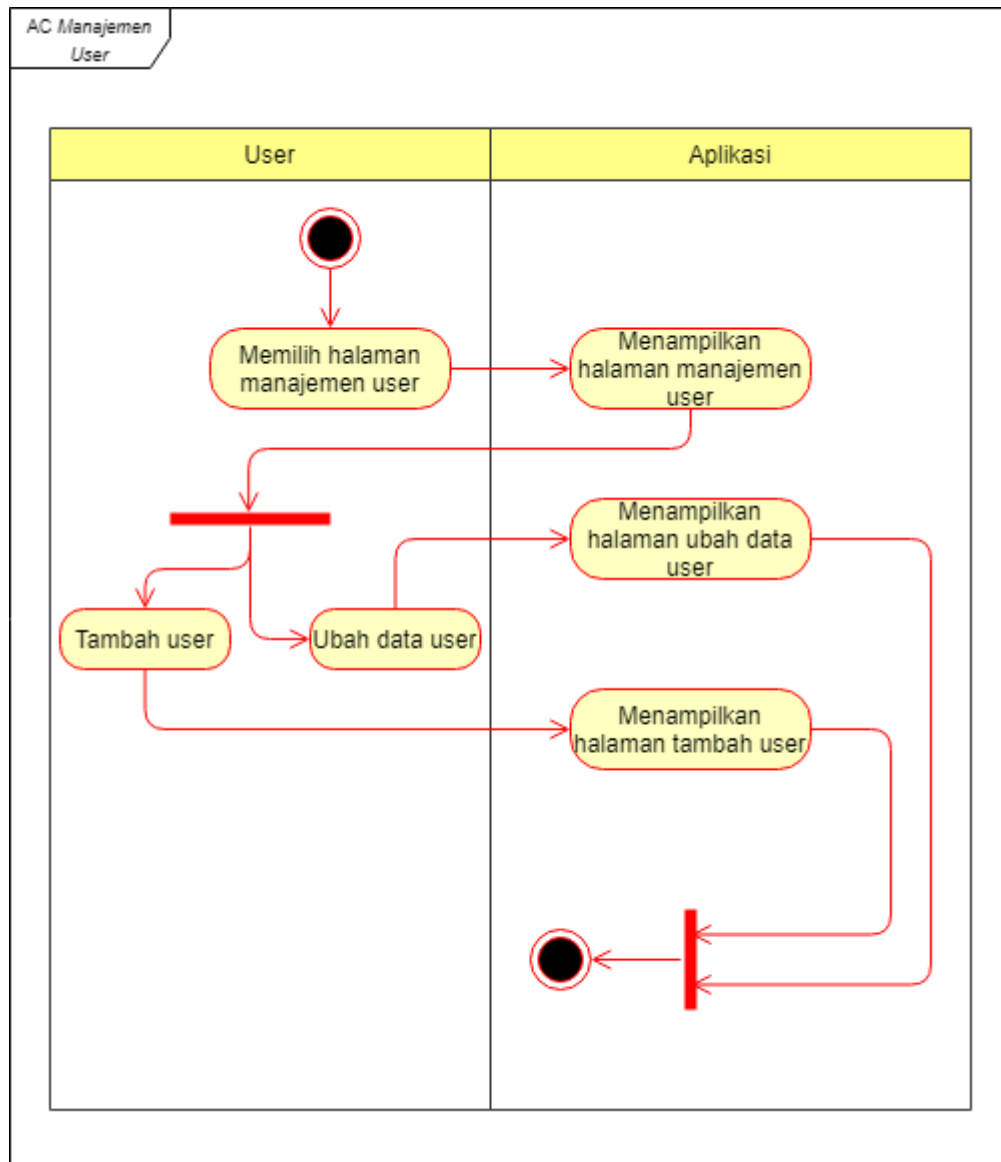


**Gambar 3.15** Activity Diagram Login Web

Pada Gambar 3.14, dapat dilihat bahwa pada saat login, super admin atau admin memasukkan username dan password, kemudian aplikasi akan mengecek jika username dan password benar maka aplikasi akan menampilkan halaman manajemen user, jika salah maka akan kembali menampilkan halaman login.

j. Activity Diagram Manajemen User

Berikut adalah activity diagram manajemen user dapat dilihat pada Gambar 3.15.

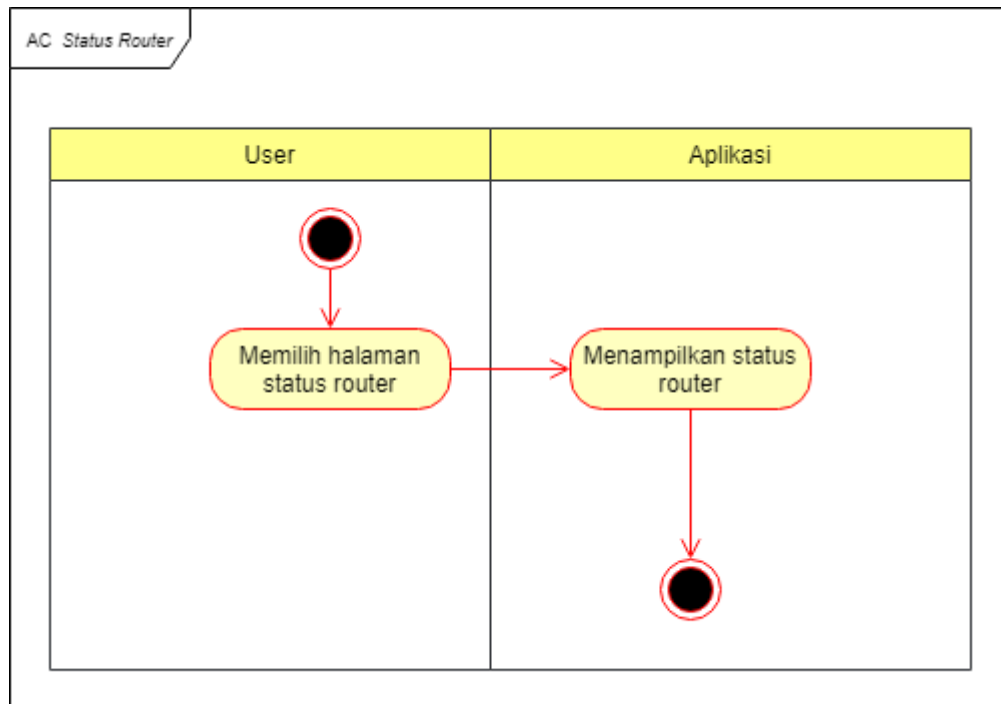


**Gambar 3.16** Activity Diagram Manajemen User

Pada Gambar 3.15, dapat dilihat bahwa pada halaman manajemen user pengguna dapat melihat daftar hak akses untuk aplikasi dan web. Pengguna juga dapat menambahkan dan merubah user yang dapat login ke aplikasi.

k. Activity Diagram Status Router

Berikut adalah activity diagram status router dapat dilihat pada Gambar 3.16.

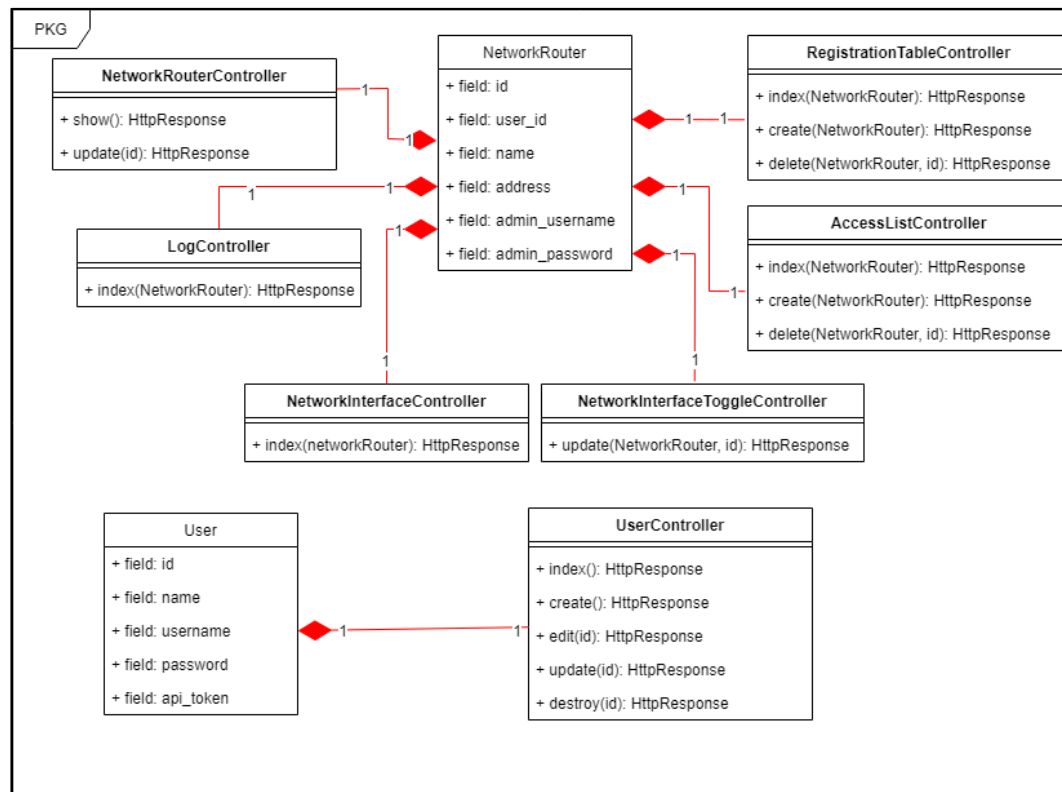


**Gambar 3.17** Activity Diagram Status Router

Pada Gambar 3.16, dapat dilihat bahwa pada halaman status router pengguna dapat melihat informasi status router.

### 3.3.2.6 Perancangan Class Diagram

Class diagram merupakan salah satu diagram utama dari UML untuk menggambarkan class atau blueprint object pada sebuah sistem. serta memiliki atribut dan operasi yang terdapat dalam sistem. Analisis pembentukan class diagram, merupakan aktivitas inti yang sangat mempengaruhi arsitektur piranti lunak yang dirancang hingga ke tahap pengkodean.



**Gambar 3.18** Class Diagram Sistem

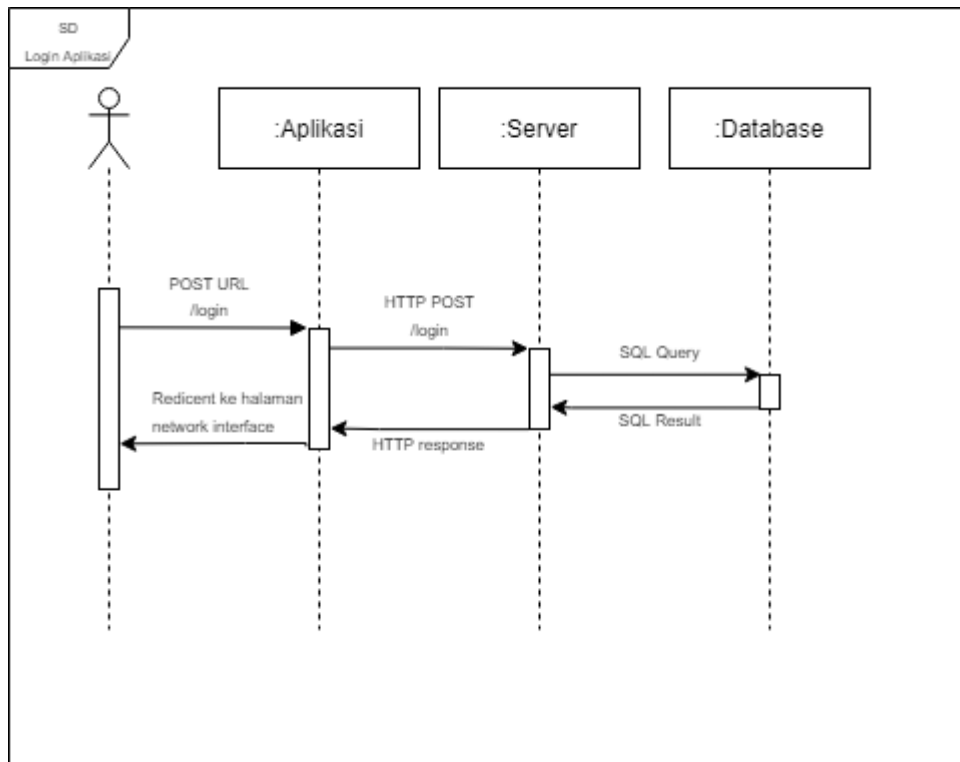
### 3.3.2.7 Perancangan Sequence Diagram

Sequence diagram adalah diagram yang menjelaskan interaksi objek berdasarkan urutan waktu. Sequence diagram menggambarkan urutan atau tahapan yang harus dilakukan pengguna. Diagram ini menunjukkan sejumlah contoh objek dan pesan yang melakukan satu tugas atau aksi tertentu. Komponen utama sequence diagram terdiri atas objek yang dituliskan dengan kotak persegi. Pesan diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan progress vertikal.

#### a. Sequence Diagram Login Aplikasi Mobile

Berikut adalah sequence diagram login aplikasi mobile dapat dilihat pada Gambar 3.18.



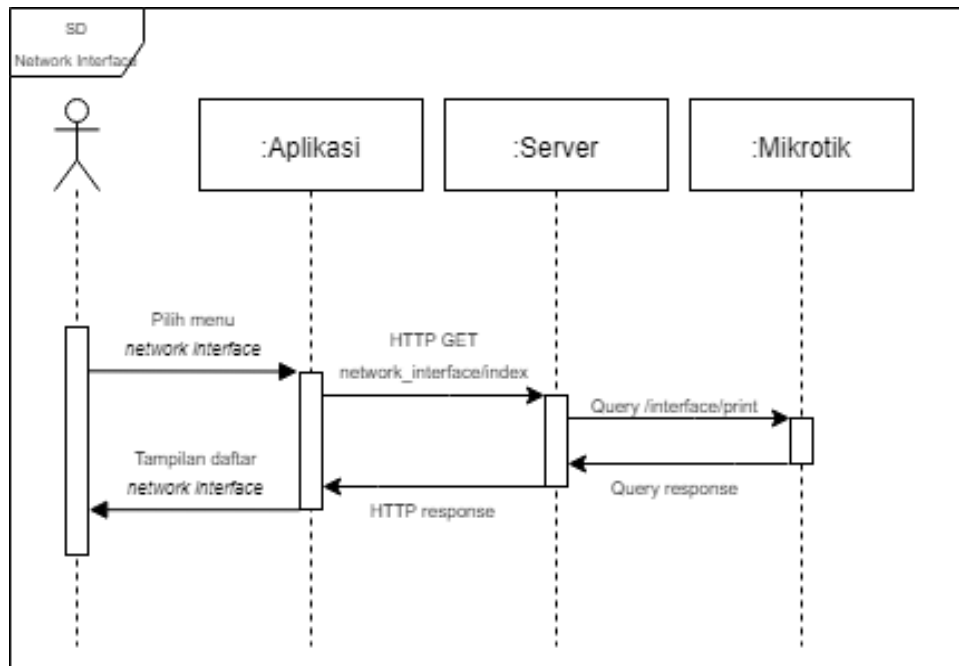


**Gambar 3.19** Sequence Diagram Login Aplikasi Mobile

Pada Gambar 3.18, dapat dilihat bahwa pengguna pada awalnya masuk aplikasi dan mengisi data untuk melakukan login kemudian menekan tombol login. kemudian sistem akan mengecek apakah username dan password benar, jika benar maka pengguna akan masuk pada aplikasi dan akan menampilkan halaman network interface sebagai halaman home.

b. Sequence Diagram Network Interface

Berikut adalah sequence diagram Network Interface dapat dilihat pada Gambar 3.19.

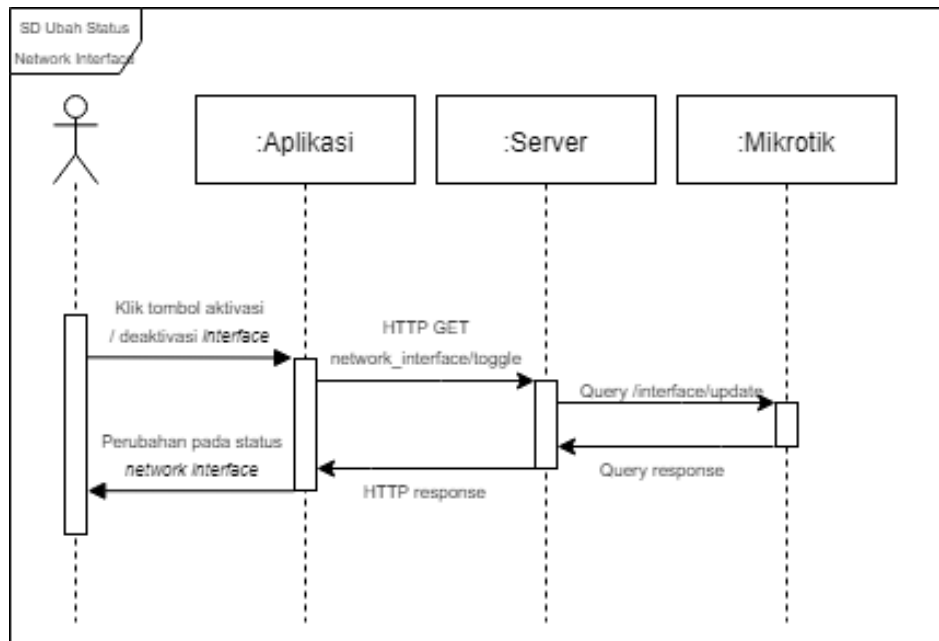


**Gambar 3.20** Sequence Diagram Network Interface

Pada Gambar 3.19, dapat dilihat bahwa pengguna pada awalnya memilih menu network interface pada aplikasi. Tindakan tersebut membuat aplikasi mengirimkan request HTTP yaitu `network_interface/index` ke server. Server lalu membaca request tersebut dan mengirimkan query yaitu `/interface/print` ke router MikroTik, yang pada akhirnya akan mengirimkan data daftar client yang terhubung ke server. Server lalu mengubah bentuk data yang diperoleh dari MikroTik menjadi format JSON. Data JSON tersebut lalu dikembalikan ke aplikasi, yang pada akhirnya aplikasi akan menampilkan data tersebut.

c. Sequence Diagram Ubah Status Network Interface

Berikut adalah sequence diagram Ubah Status Network Interface dapat dilihat pada Gambar 3.20.

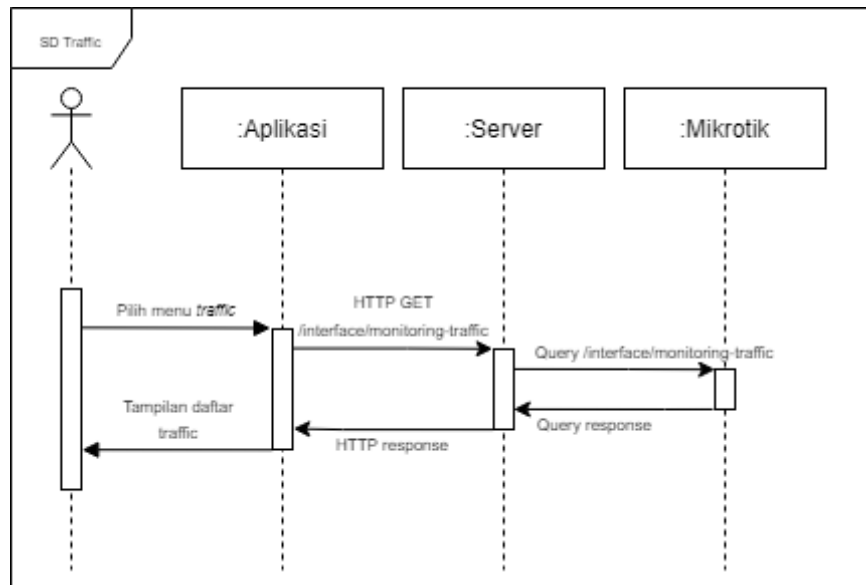


**Gambar 3.21** Sequence Diagram Ubah Status Network Interface

Pada Gambar 3.20, dapat dilihat bahwa pengguna pada awalnya memilih menu network interface pada aplikasi kemudian menekan tombol disable atau undisable. Tindakan tersebut membuat aplikasi mengirimkan request HTTP yaitu `network_interface/loggle` ke server. Server lalu membaca request tersebut dan mengirimkan query yaitu `/interface/update` ke router MikroTik dan mikrotik akan membalas dengan mengubah status menjadi disable atau undisable. Server lalu mengubah bentuk data yang diperoleh dari MikroTik menjadi format JSON. Data JSON tersebut lalu dikembalikan ke aplikasi, yang pada akhirnya akan mengubah status menjadi disable atau undisable dan ditampilkan pada aplikasi.

d. Sequence Diagram Traffic

Berikut adalah sequence diagram log dapat dilihat pada Gambar 3.21.

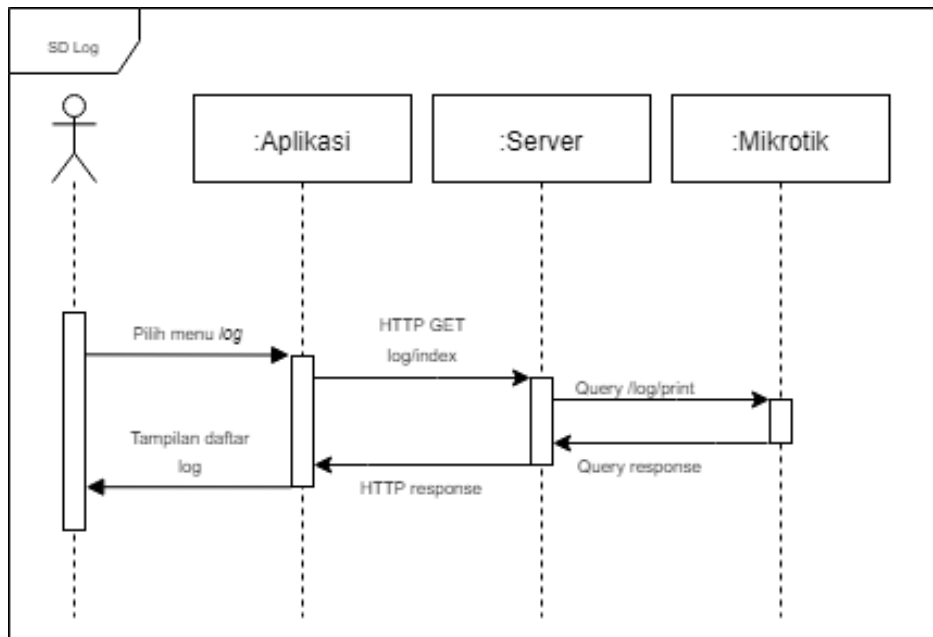


**Gambar 3.22** Sequence Diagram Traffic

Pada Gambar 3.21, dapat dilihat bahwa pengguna pada awalnya memilih menu traffic pada aplikasi. Tindakan tersebut membuat aplikasi mengirimkan request HTTP yaitu `/interface/monitor-traffic` ke server. Server lalu membaca request tersebut dan mengirimkan query yaitu `/interface/monitor-traffic/print` ke router MikroTik, yang pada akhirnya akan mengirimkan data traffic atau kecepatan upload dan download mikrotik ke server. Server lalu mengubah bentuk data yang diperoleh dari MikroTik menjadi format JSON. Data JSON tersebut lalu dikembalikan ke aplikasi, yang pada akhirnya aplikasi akan menampilkan data tersebut.

e. Sequence Diagram Router Log

Berikut adalah sequence diagram router log dapat dilihat pada Gambar 3.21.

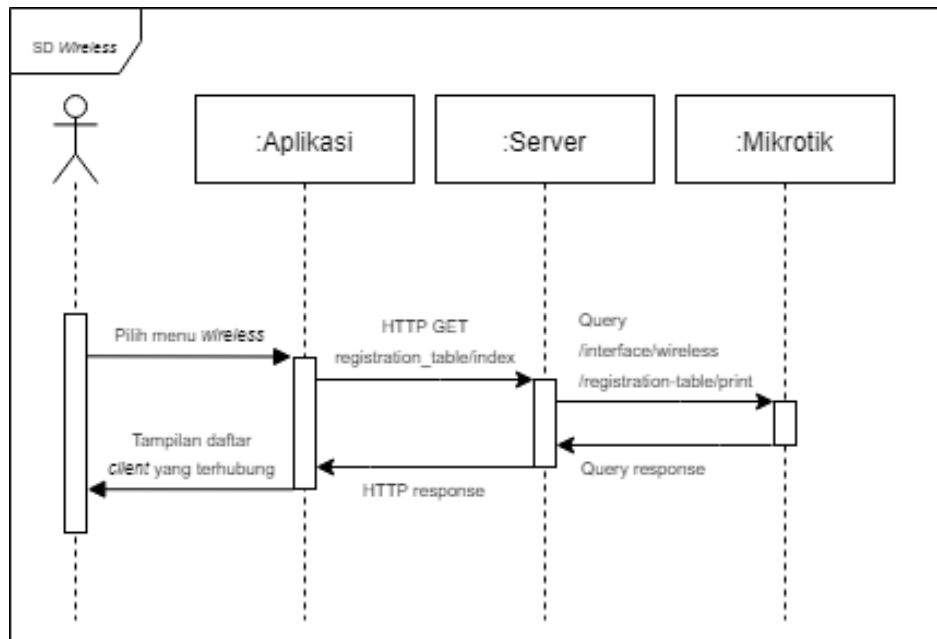


**Gambar 3.23** Sequence Diagram Router Log

Pada Gambar 3.22, dapat dilihat bahwa pengguna pada awalnya memilih menu router log pada aplikasi. Tindakan tersebut membuat aplikasi mengirimkan request HTTP yaitu log/index ke server. Server lalu membaca request tersebut dan mengirimkan query yaitu /log/print ke router MikroTik, yang pada akhirnya akan mengirimkan data log atau history ke server. Server lalu mengubah bentuk data yang diperoleh dari MikroTik menjadi format JSON. Data JSON tersebut lalu dikembalikan ke aplikasi, yang pada akhirnya aplikasi akan menampilkan data log perangkat mikrotik tersebut.

f. Sequence Diagram Wireless

Berikut adalah sequence diagram wireless dapat dilihat pada Gambar 3.23.

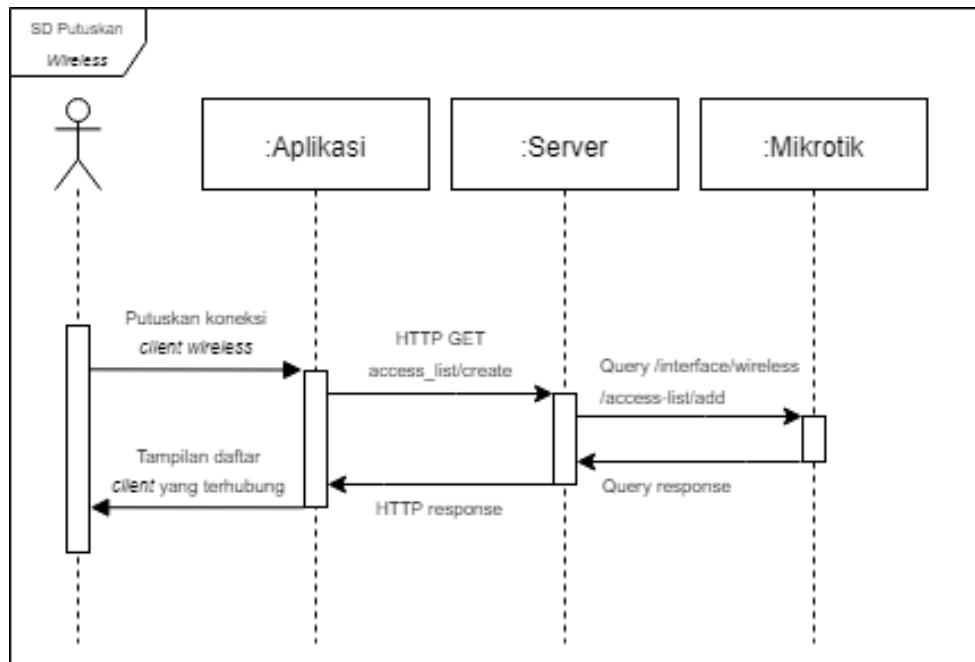


**Gambar 3.24** Sequence Diagram Wireless

Pada Gambar 3.23, dapat dilihat bahwa pengguna pada awalnya memilih menu wireless pada aplikasi. Tindakan tersebut membuat aplikasi mengirimkan request HTTP ke server yaitu `registration_table/index`. Server lalu membaca request tersebut dan mengirimkan query ke router MikroTik yaitu `/interface/wireless/registration-table/print`, yang pada akhirnya akan mengirimkan data daftar client yang terhubung ke server. Server lalu mengubah bentuk data yang diperoleh dari MikroTik menjadi format JSON. Data JSON tersebut lalu dikembalikan ke aplikasi, yang pada akhirnya akan menampilkan data tersebut.

g. Sequence Diagram Putuskan Wireless

Berikut adalah sequence diagram putuskan wireless dapat dilihat pada Gambar 3.24.

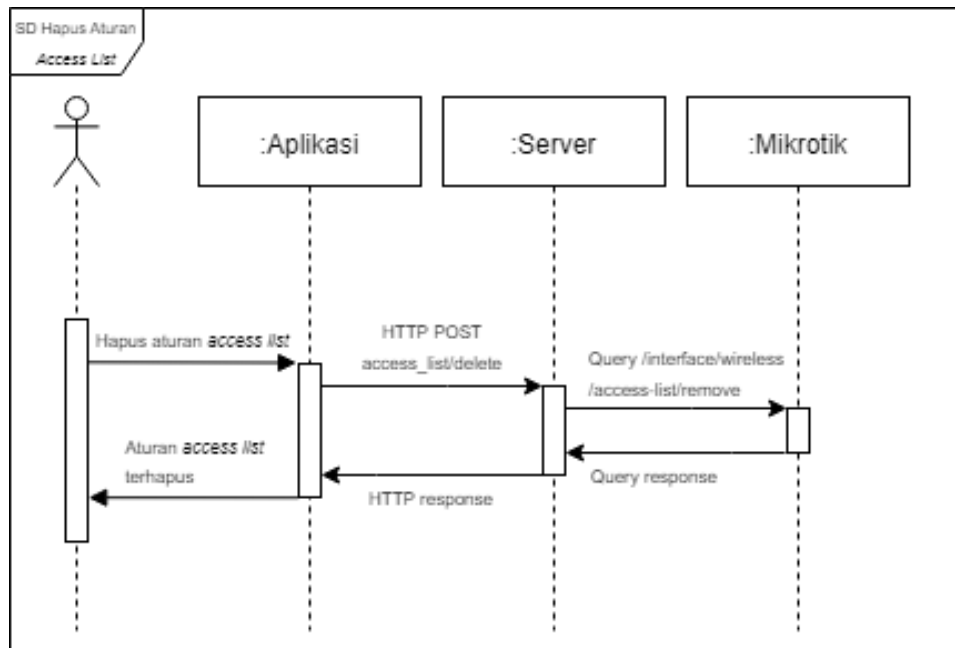


**Gambar 3.25** Sequence Diagram Putuskan Wireless

Pada Gambar 3.24, dapat dilihat bahwa pengguna pada awalnya masuk ke halaman wireless dan kemudian menekan tombol disconnect/ban. Tindakan tersebut membuat aplikasi mengirimkan request HTTP ke server yaitu `access_list/create`. Server lalu membaca request tersebut dan mengirimkan query ke router MikroTik yaitu `/interface/wireless/access-list/add` dan mikrotik akan membalas dengan memutuskan client dan mengirimkan data informasi ke server. Server lalu mengubah bentuk data yang diperoleh dari MikroTik menjadi format JSON. Data JSON tersebut lalu dikembalikan ke aplikasi, yang pada akhirnya akan ditampilkan pada aplikasi.

#### h. Sequence Diagram Hapus Aturan *Ban List*

Berikut adalah sequence diagram hapus aturan *ban list* dapat dilihat pada Gambar 3.25.



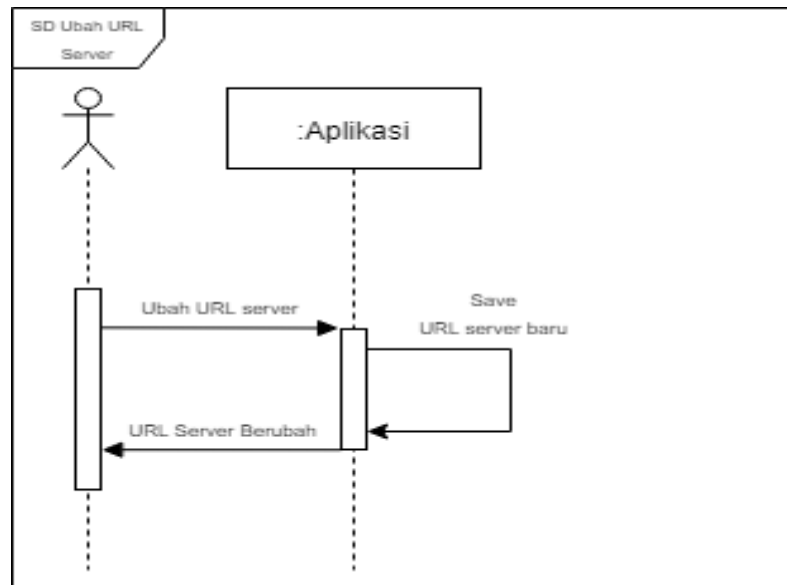
**Gambar 3.26** Sequence Diagram Hapus Aturan *Ban List*

Pada Gambar 3.25, dapat dilihat bahwa pengguna pada awalnya masuk ke halaman *ban list* dan kemudian menekan tombol *delete*. Tindakan tersebut membuat aplikasi mengirimkan request HTTP ke server yaitu `access_list/delete`. Server lalu membaca request tersebut dan mengirimkan query ke router MikroTik yaitu `/interface/wireless/access-list/remove` dan mikrotik akan membalas dengan menghapus dari *ban list* dan mengirimkan data informasi ke server. Server lalu mengubah bentuk data yang diperoleh dari MikroTik menjadi format JSON. Data JSON tersebut lalu dikembalikan ke aplikasi, yang pada akhirnya akan ditampilkan pada aplikasi.

i. Sequence Diagram Ubah URL Server

Berikut adalah sequence diagram ubah URL server dapat dilihat pada Gambar 3.26.





**Gambar 3.27** Sequence Diagram Ubah Profil

Pada Gambar 3.26, dapat dilihat bahwa pengguna mengubah alamat URL server kemudian menekan tombol save settings. Jika berhasil, maka URL server telah berubah.

### 3.3.2.8 Perancangan Basis Data

Sistem ini memiliki 8 tabel, yang dapat dilihat pada rincian berikut.

a. Spesifikasi Tabel User

Nama *database* : mikrotik\_monitoring\_api

Nama tabel : users

Keterangan : tabel yang digunakan untuk menentukan tipe *user*

**Tabel 3.3** Spesifikasi Tabel User

Nama <i>Field</i>	Tipe	Boleh Kosong	Keterangan
Id	<i>int</i> (10)	Tidak	Kunci Primer
name	<i>varchar</i> (255)	Tidak	
email	<i>varchar</i> (255)	Iya	Foreign Key
level	<i>varchar</i> (255)	Tidak	
email_verified_at	<i>timestamp</i>	Iya	
password	<i>varchar</i> (255)	Tidak	

api_token	<i>varchar(80)</i>	Iya	Foreign Key
remember_token	<i>varchar(100)</i>	Iya	
create_at	<i>timestamp</i>	Iya	
update_at	<i>timestamp</i>	Iya	
username	<i>varchar(255)</i>	Tidak	Foreign Key

b. Spesifikasi Tabel Network Router

Nama *database* : mikrotik\_monitoring\_api

Nama tabel : network\_routers

Keterangan : tabel yang digunakan untuk data mikrotik

**Tabel 3.4** Spesifikasi Tabel Network Router

<b>Nama Field</b>	<b>Type</b>	<b>Boleh Kosong</b>	<b>Keterangan</b>
id	<i>int (10)</i>	Tidak	Kunci Primer
user_id	<i>int (10)</i>	Tidak	Foreign Key
name	<i>varchar(255)</i>	Tidak	
host	<i>varchar(255)</i>	Tidak	
admin_username	<i>varchar(255)</i>	Tidak	
admin_password	<i>varchar(255)</i>	Tidak	
created_at	<i>timestamp</i>	Iya	
update_at	<i>timestamp</i>	Iya	

c. Spesifikasi Tabel User Logs

Nama *database* : mikrotik\_monitoring\_api

Nama tabel : network\_routers

Keterangan : tabel yang digunakan untuk data mikrotik

**Tabel 3.5** Spesifikasi Tabel User Logs

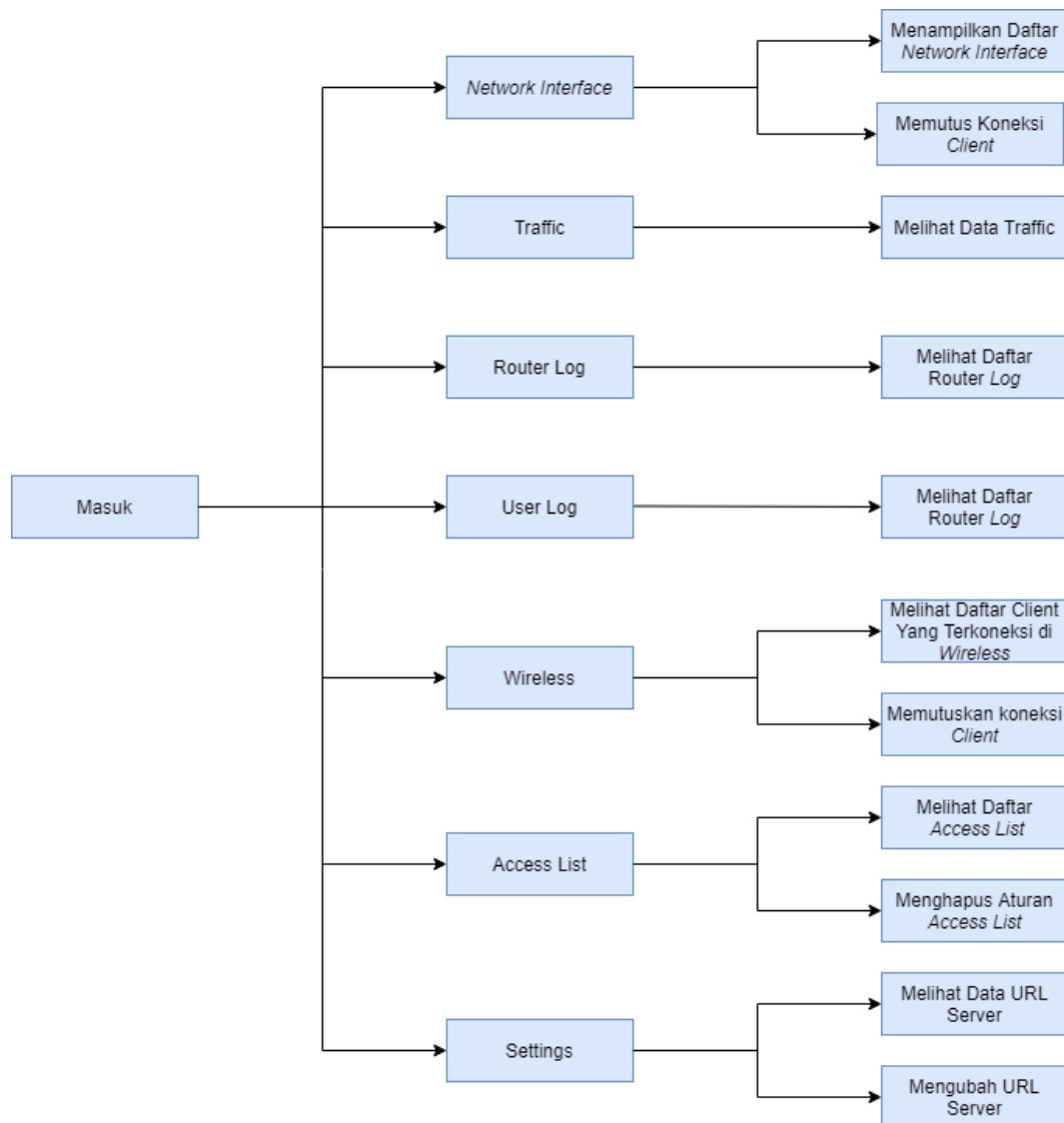
<b>Nama Field</b>	<b>Tipe</b>	<b>Boleh Kosong</b>	<b>Keterangan</b>
id	<i>int</i> (10)	Tidak	Kunci Primer
user_id	<i>int</i> (10)	Tidak	Foreign Key
text	<i>varchar</i> (255)	Tidak	
created_at	<i>timestamp</i>	Iya	
update_at	<i>timestamp</i>	Iya	

### 3.3.2.9 Perancangan Antarmuka

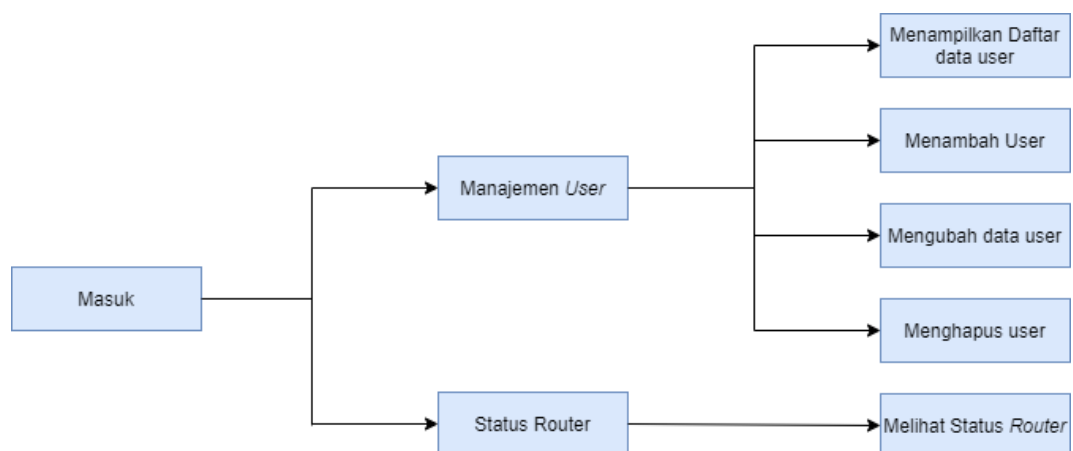
Antarmuka merupakan tampilan dari suatu program aplikasi yang berperan sebagai sarana interaksi antara program dengan user. Sistem yang dibangun diharapkan dapat menyediakan antarmuka yang mudah dimengerti dan mudah digunakan oleh user. Berikut ini adalah perancangan antarmuka pada sistem yang akan dibuat.

#### 3.3.2.9.1 Perancangan Struktur Antarmuka Sistem

Adapun struktur antarmuka aplikasi dijabarkan sebagai berikut.



**Gambar 3.28** Perancangan Struktur Antarmuka Sistem Aplikasi Android



**Gambar 3.29** Perancangan Struktur Antarmuka Sistem Website

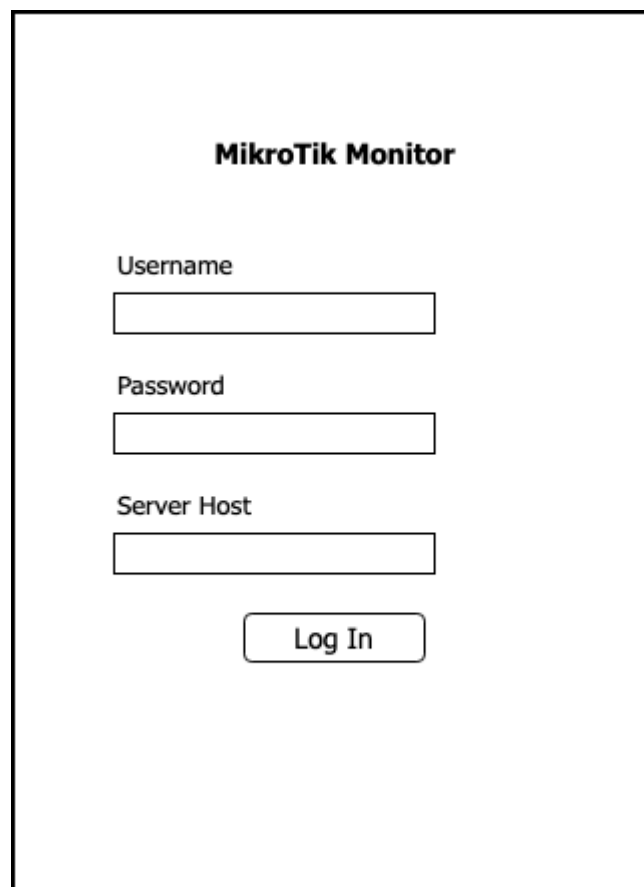
Sistem ini merupakan aplikasi berbasis android dan website sebagai web service yang nantinya setiap data yang masuk akan tersimpan di database. Antarmuka aplikasi yang dibangun memiliki beberapa tampilan yang disesuaikan dengan menu-menu aplikasi yang ada.

#### **3.3.2.9.2 Perancangan Layout**

Adapun perancangan layout dari aplikasi yang akan dibuat pada aplikasi platform android menggunakan Draw io. Adapun perancangan layout aplikasi dijabarkan sebagai berikut.

a. Perancangan Halaman Login Aplikasi Mobile

Halaman Login Aplikasi Mobile berfungsi untuk menampilkan form yang harus di isi oleh pengguna untuk masuk ke aplikasi. Halaman ini berisi username, password, dan server host apabila pengguna mengisi data username, password, dan server host dengan benar maka pengguna dapat masuk ke aplikasi manajemen jaringan. Rancangan Halaman Login Aplikasi Mobile dapat dilihat pada Gambar 3.28.



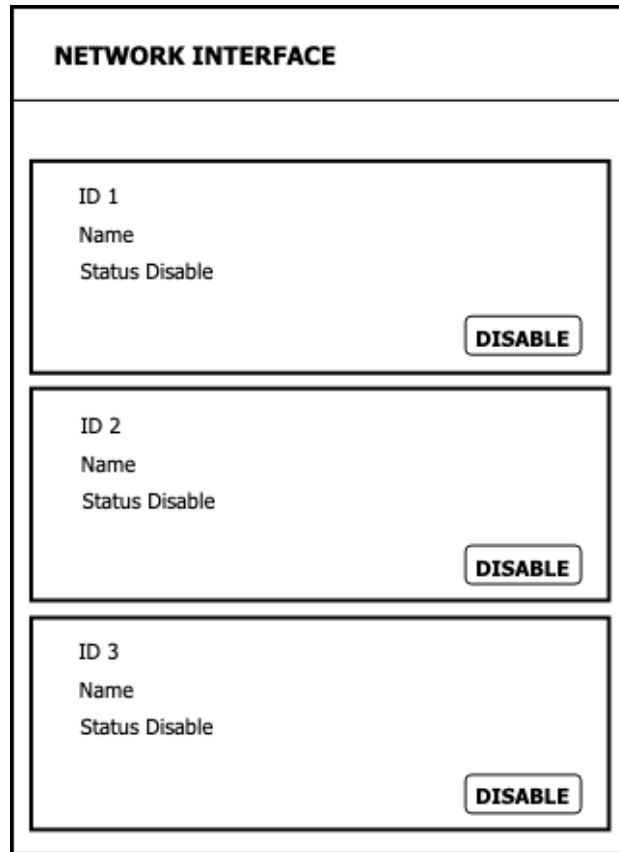
The image shows a login form titled "MikroTik Monitor". It contains three input fields: "Username", "Password", and "Server Host". Below these fields is a "Log In" button. The form is enclosed in a rectangular border.

```
graph TD; Title[MikroTik Monitor]; Username[Username]; Password[Password]; ServerHost[Server Host]; LogIn[Log In];
```

**Gambar 3.30** Rancangan Halaman Login

b. Perancangan Halaman Network Interface

Halaman Network Interface berfungsi untuk menampilkan data dari mikrotik yaitu port-port yang ada di mikrotik, WLAN, dan bridge. Rancangan Halaman Network Interface dapat dilihat pada Gambar 3.29.



The image shows a wireframe for a 'NETWORK INTERFACE' page. It features a title bar at the top, followed by three vertically stacked card-like containers. Each container displays 'ID 1', 'ID 2', and 'ID 3' respectively, along with labels for 'Name' and 'Status Disable'. A 'DISABLE' button is positioned at the bottom right of each container.

NETWORK INTERFACE	
ID 1	<div>Name Status Disable</div> <div>DISABLE</div>
ID 2	<div>Name Status Disable</div> <div>DISABLE</div>
ID 3	<div>Name Status Disable</div> <div>DISABLE</div>

**Gambar 3.31** Rancangan Halaman Network Interface

c. Perancangan Halaman Traffic

Halaman Traffic berfungsi untuk menampilkan informasi data kecepatan upload dan download dari Mikrotik. Rancangan halaman Traffic dapat dilihat pada Gambar 3.30.

TRAFFIC
Download Speed
Upload Speed

**Gambar 3.32** Rancangan Halaman Traffic

d. Perancangan Halaman Router Log

Halaman Router Log berfungsi untuk menampilkan informasi data history dari Router Mikrotik. Rancangan halaman router log dapat dilihat pada Gambar 3.31.

ROUTER LOG
ID 1 Time Topic Message
ID 2 Time Topic Message
ID 3 Time Topic Message

**Gambar 3.33** Rancangan Halaman Router Log

e. Perancangan Halaman User Log

Halaman User Log berfungsi untuk menampilkan informasi data history dari aplikasi manajemen jaringan. Rancangan halaman user log dapat dilihat pada Gambar 3.32.



USER LOG
<div><div>ID 1 Time Content User</div></div>
<div><div>ID 1 Time Content User</div></div>
<div><div>ID 1 Time Content User</div></div>

**Gambar 3.34** Rancangan Halaman User Log

f. Perancangan Halaman Wireless

Halaman Wireless berfungsi untuk menampilkan user yang terhubung dengan mikrotik melalui wireless. Rancangan Halaman Wireless dapat dilihat pada Gambar 3.33.

WIRELESS	
ID 1 Interface MAC Address Access Point Bandwith Usage Download Speed Upload Speed	DISCONNECT / BAN
ID 2 Interface MAC Address Access Point Bandwith Usage Download Speed Upload Speed	DISCONNECT / BAN

**Gambar 3.35** Rancangan Halaman Wireless

g. Perancangan Halaman Ban List

Halaman Ban List berfungsi untuk menampilkan user yang di ban. Rancangan Halaman Ban List dapat dilihat pada Gambar 3.34.

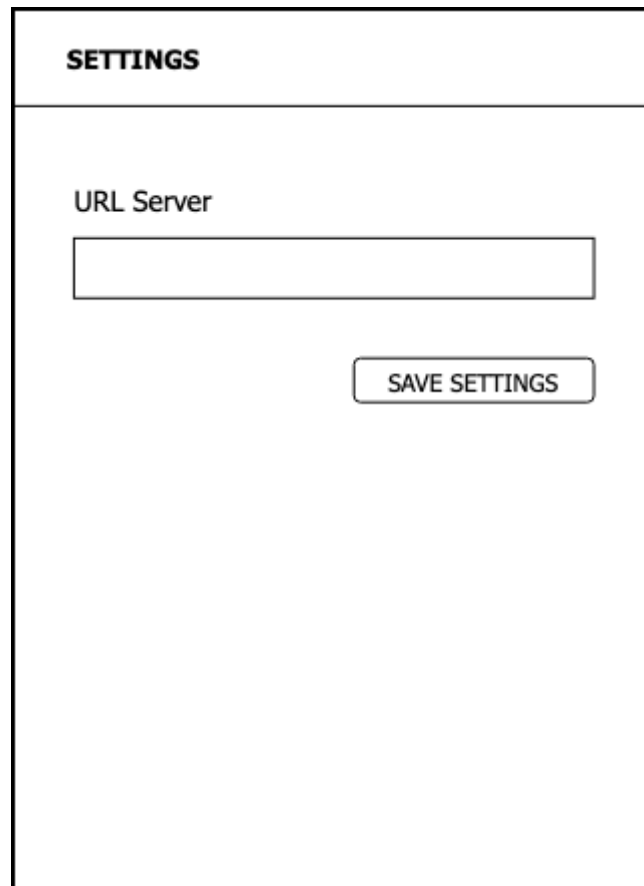
The wireframe shows a page titled "ACCESS LIST". Below the title, there are three identical entries stacked vertically. Each entry consists of a text label "MAC Address 1", "MAC Address 2", and "MAC Address 3" respectively, followed by a "DELETE" button.

ACCESS LIST	
MAC Address 1	DELETE
MAC Address 2	DELETE
MAC Address 3	DELETE

**Gambar 3.36** Rancangan Halaman Ban List

h. Perancangan Halaman Settings

Halaman Settings berfungsi untuk mengatur URL Server aplikasi. Rancangan Halaman Settings dapat dilihat pada Gambar 3.35.



**SETTINGS**

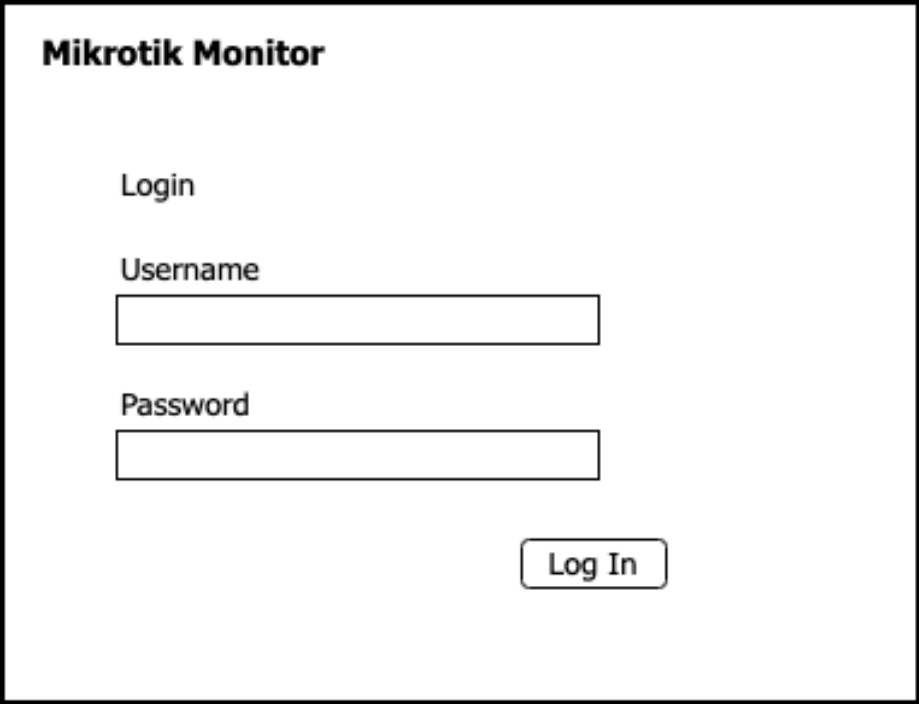
URL Server

SAVE SETTINGS

**Gambar 3.37** Rancangan Halaman Settings

i. Perancangan Halaman Login Web

Halaman Login Web berfungsi untuk menampilkan form yang harus diisi oleh pengguna untuk masuk ke halaman web. Halaman ini berisi username dan password apabila pengguna mengisi data username dan password dengan benar maka pengguna dapat masuk ke halaman web. Rancangan Halaman Login Web dapat dilihat pada Gambar 3.36.



**Mikrotik Monitor**

Login

Username

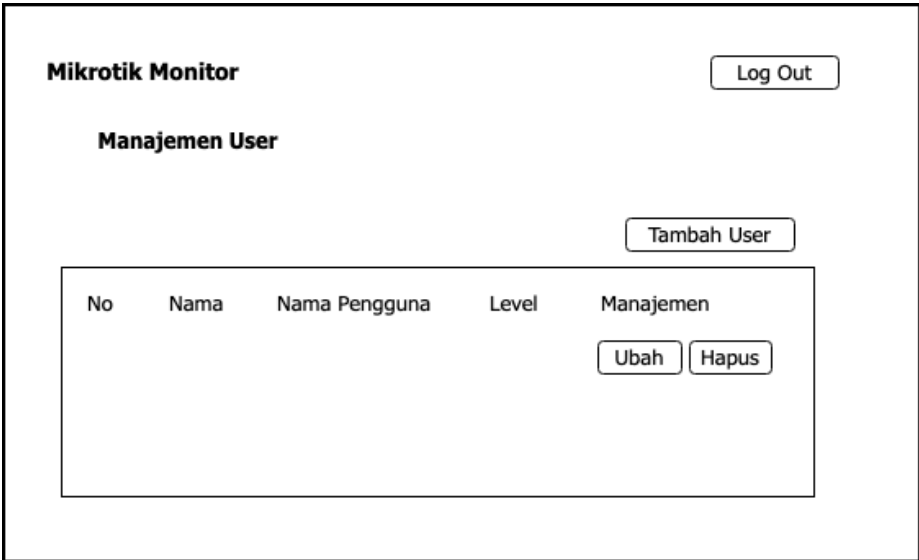
Password

**Log In**

**Gambar 3.38** Rancangan Halaman Login Web

j. Perancangan Halaman Manajemen User

Halaman Manajemen User berfungsi untuk menampilkan daftar pengguna yang bisa mengakses aplikasi manajemen jaringan mikrotik. Halaman manajemen user hanya dapat diakses oleh super admin. Halaman Manajemen User dapat dilihat pada Gambar 3.37.



**Mikrotik Monitor**

**Manajemen User**

**Log Out**

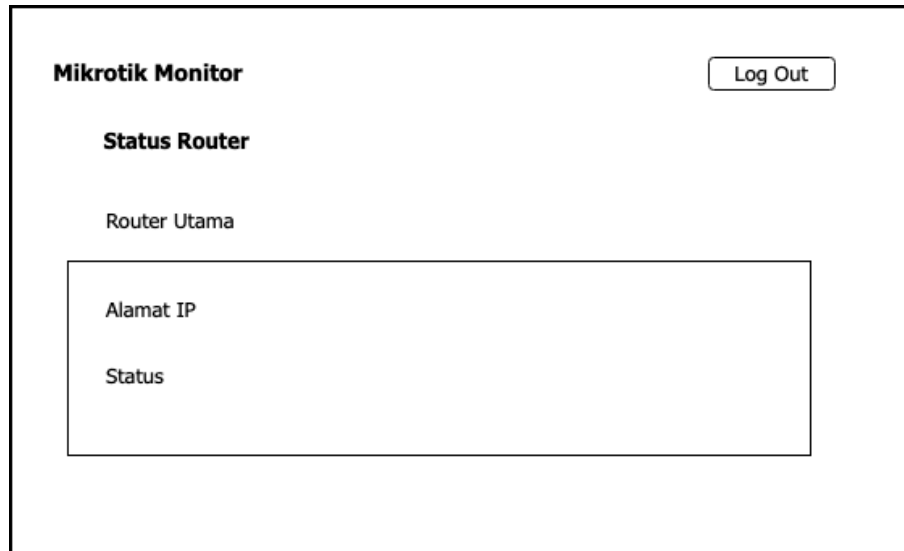
**Tambah User**

No	Nama	Nama Pengguna	Level	Manajemen
				<b>Ubah</b> <b>Hapus</b>

**Gambar 3.39** Rancangan Halaman Manajemen User

k. Perancangan Halaman Status Router

Halaman Status Router berfungsi untuk menampilkan alamat ip dan status router apakah terhubung atau tidak. Rancangan Halaman Status Router dapat dilihat pada Gambar 3.38.

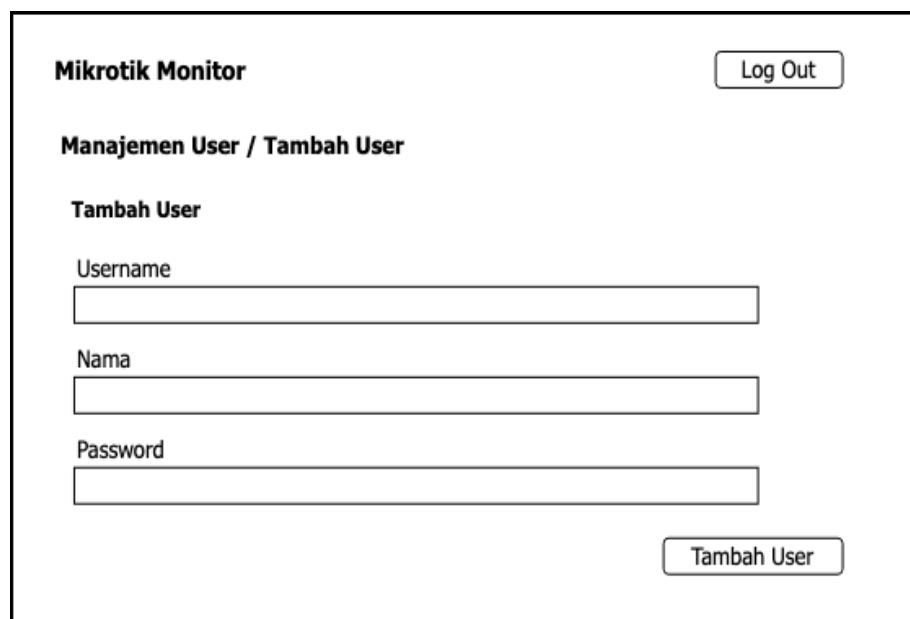


The image shows a web interface for 'Mikrotik Monitor'. At the top right is a 'Log Out' button. Below the title is the section 'Status Router'. Underneath, it says 'Router Utama'. A large rectangular box contains two labels: 'Alamat IP' and 'Status'.

**Gambar 3.40** Rancangan Halaman Status Router

l. Perancangan Halaman Tambah User

Halaman tambah User berfungsi untuk menambahkan user atau pengguna yang dapat mengakses atau masuk ke aplikasi manajemen jaringan mikrotik. Rancangan Halaman Tambah User dapat dilihat pada Gambar 3.39.



The image shows a web interface for 'Mikrotik Monitor'. At the top right is a 'Log Out' button. Below the title is the section 'Manajemen User / Tambah User'. Underneath, it says 'Tambah User'. There are three input fields labeled 'Username', 'Nama', and 'Password'. At the bottom right is a 'Tambah User' button.

**Gambar 3.41** Rancangan Halaman Tambah User

m. Perancangan Halaman Ubah User

Halaman Ubah User berfungsi untuk mengubah data dari user yang telah ada. Rancangan Halaman Ubah User dapat dilihat pada Gambar 3.40.

The image shows a web interface for Mikrotik Monitor. At the top left is the text 'Mikrotik Monitor'. At the top right is a button labeled 'Log Out'. Below this is a section titled 'Manajemen User / Ubah User'. Underneath is a sub-section titled 'Ubah User'. There are three input fields: 'Username', 'Nama', and 'Password'. At the bottom right is a button labeled 'Ubah User'.

**Gambar 3.42** Rancangan Halaman Ubah User

### 3.3.3 Perancangan Pengujian Sistem

Pengujian pada aplikasi dilakukan dengan menggunakan metode Black Box yang akan menguji fungsionalitas, performa, penggunaan memori, dan kompatibilitas, serta metode kuesioner untuk mengetahui tingkat keberhasilan pembuatan aplikasi.

#### 3.3.3.1 Pengujian *Black Box*

Pengujian fungsional yang digunakan untuk menguji sistem yang baru adalah metode pengujian alpha. Metode yang digunakan dalam pengujian ini adalah pengujian Black Box yang berfokus pada fungsional, performa, penggunaan memori, dan kompatibilitas aplikasi yang dibangun. Pada pengujian ini menggunakan sistem yaitu Bitbar Testing yang akan menggunakan empat devices yaitu Motorola Google Nexus 6 Android versi 7.1.1 dan versi 6.0.1, LG Google Nexus 5 D820, dan LG Google Nexus 5. Pengujian Black Box digunakan untuk menguji validitas dari integrasi dan konsistensi sistem.

### 3.3.3.2 Pengujian Kuesioner

Pengujian kuesioner yaitu menguji tingkat penerimaan pengguna terhadap sistem melalui kuesioner berdasarkan variabel pada user interface dan user experience. Kuesioner dibagikan kepada 10 responden. Responden akan mencoba aplikasi untuk dapat mengetahui kapabilitas dan kelancaran jalannya aplikasi itu sendiri.

Kuesioner berisi beberapa pertanyaan yang dikelompokkan menjadi tiga aspek yang digunakan dalam pengujian aplikasi tersebut, yaitu aspek rekayasa perangkat lunak, aspek fungsionalitas, dan aspek komunikasi visual. Tiga aspek ini berisikan pertanyaan-pertanyaan yang mengukur kemudahan dari suatu sistem saat digunakan seperti kemudahan mengakses fitur, kelancaran menjalankan tombol, sehingga bisa membuat user tertarik (Usability), melihat tampilan sistem yang disajikan menarik atau tidak seperti kombinasi warna, jenis dan ukuran font, antarmuka aplikasi, kemudahan melihat hasil, sehingga akan membuat user merasa senang saat menggunakan aplikasi (Pleasure), dan mengukur kesesuaian tampilan informasi yang disajikan pada aplikasi (Content dan Quality of Information).

Perhitungan dilakukan dengan menggunakan metode Likert's Summated Rating (LSR), dimana setiap pertanyaan memiliki bobot nilai. Adapun bobot jawaban yang diberikan adalah sebagai berikut:

**Tabel 3.6** Bobot Jawaban pada Perhitungan Skala Likert

Keterangan Jawaban	Bobot Nilai
Sangat Baik	5
Baik	4
Cukup Baik	3
Kurang Baik	2
Tidak Baik	1

Selanjutnya aplikasi akan di uji coba kepada pengguna. Hasil pengujian dengan kuesioner akan diukur menggunakan metode Likert's Summated Rating (LSR), pengukuran akan dilakukan kepada pengguna yang sudah mencoba aplikasi. Tabel perhitungan Skala Likert dapat dilihat pada Tabel 3.7.



**Tabel 3.7** Tabel Perhitungan Skala Likert

No.	Aspek / Pertanyaan	Nilai Pengujian					Persentase Likert
		TB	KB	CB	B	SB	
Aspek Rekayasa Perangkat Lunak							
1.	Kemudahan menjalankan aplikasi						
2.	Kompabilitas aplikasi pada perangkat						
3.	Kelancaran menjalankan aplikasi pada perangkat						
4.	Kemudahan mengakses fitur-fitur pada aplikasi						
5.	Kenyamanan dalam menggunakan aplikasi secara keseluruhan						
Aspek Fungsionalitas							
6.	Kemudahan dalam melakukan login aplikasi						
7.	Kemudahan menjalankan tombol pada aplikasi						
8.	Tingkat kesesuaian menampilkan <i>input</i> , proses, dan <i>output</i> dari pada aplikasi android						
9.	Tingkat kesesuaian aplikasi dalam menampilkan informasi berupa data pada user yang terhubung						
10.	Tingkat kemudahan dalam memonitoring besaran bandwith pada setiap user						
Aspek Komunikasi Visual							
11.	Kemudahan memahami informasi yang ditampilkan pada aplikasi						
12.	Tampilan antarmuka aplikasi						
13.	Jenis dan ukuran huruf						

No.	Aspek / Pertanyaan	Nilai Pengujian					Persentase Likert
		TB	KB	CB	B	SB	
	yang digunakan mudah dibaca						
14.	Kombinasi warna pada tampilan aplikasi						
15.	Tingkat kemudahan melihat hasil yang ditampilkan aplikasi						
<b>Total Persentase (%)</b>							

Nilai persentase skala likert dihitung dengan rumus berikut ini:

$$L = \frac{(SB \times 5) + (B \times 4) + (CB \times 3) + (KB \times 2) + (TB \times 1)}{NB} \times 100$$

Keterangan :

- L : Nilai persentase likert
- SB : Jumlah responden yang memilih pilihan sangat baik
- B : Jumlah responden yang memilih pilihan baik
- CB : Jumlah responden yang memilih pilihan cukup baik
- KB : Jumlah responden yang memilih pilihan kurang baik
- TB : Jumlah responden yang memilih pilihan tidak baik
- NB : 5 (bobot tertinggi) x jumlah responden

Untuk menyimpulkan hasil akhir dari aplikasi, harus diketahui terlebih dahulu nilai interval total persentase dengan metode pencarian nilai interval pada persamaan berikut ini.

$$I = \frac{100}{5} = 20$$

Nilai 5 didapat dari jumlah pilihan yang diberikan dalam kuesioner.

Nilai interval persentase pada kuesioner ini yaitu:

- $X \leq 19,99\%$  dikategorikan tidak baik
- $20\% \leq X \leq 39,99\%$  dikategorikan kurang baik
- $40\% \leq X \leq 59,99\%$  dikategorikan cukup baik
- $60\% \leq X \leq 79,99\%$  dikategorikan baik
- $X \geq 80\%$  dikategorikan sangat baik

#### **3.3.4 Analisis Hasil Pengujian**

Pada analisa hasil pengujian, dilakukan analisis terhadap sistem secara keseluruhan berdasarkan pengujian yang sudah dilakukan.

#### **3.3.5 Penarikan Kesimpulan**

Kesimpulan dirumuskan berdasarkan analisis hasil pengujian sistem yang dilakukan apakah sistem yang dirancang sesuai dengan tujuan yang ingin dicapai dalam penelitian.

## BAB IV

### IMPLEMENTASI DAN HASIL PENGUJIAN

#### 4.1 Implementasi

Berdasarkan hasil analisis dan pengujian terhadap aplikasi manajemen jaringan mikrotik di lab informatika untan berbasis android *mobile*, dapat disimpulkan bahwa manajemen mikrotik menggunakan aplikasi *mobile* dapat mempermudah administrator jaringan di lab informatika untan yaitu asisten lab jaringan untuk memantau *user* atau *client* yang terhubung dan dapat meremote mikrotik ntuk memutus koneksi pada *user*.

##### 4.1.1 Tampilan Antarmuka Halaman *Login* Aplikasi

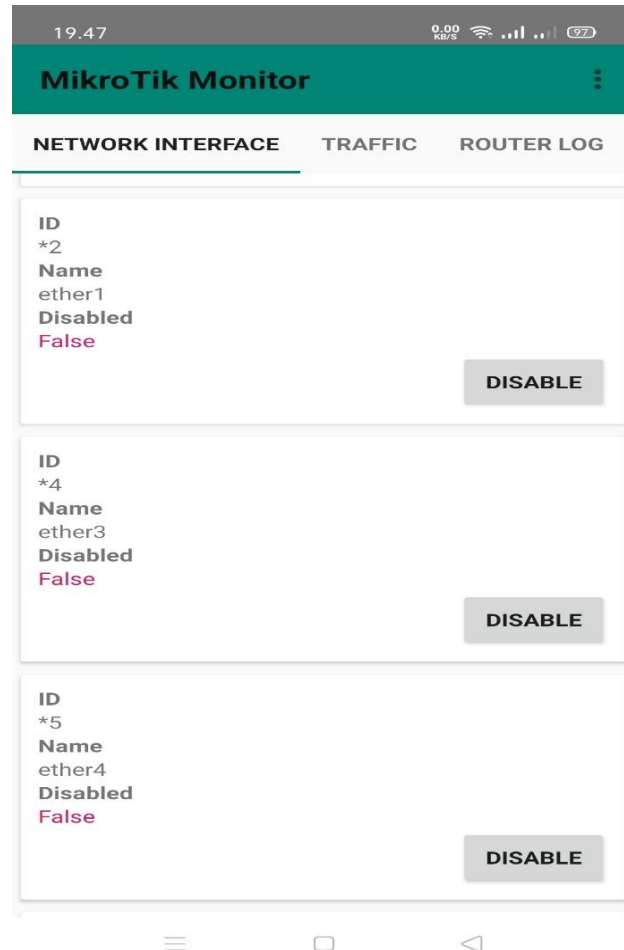
Pada tampilan halaman *Login* Aplikasi menampilkan form yang harus di isi untuk masuk ke aplikasi yaitu harus mengisi *username*, *password*, dan Server Host. Antarmuka halaman *Login* Aplikasi dapat dilihat pada Gambar 4.1.



**Gambar 4.1** Antarmuka Halaman *Login* Aplikasi

#### 4.1.2 Tampilan Antarmuka Halaman *Network Interface*

Pada tampilan halaman *Network Interface* menampilkan *client* yang terhubung atau terkoneksi pada mikrotik dan dapat mengubah status menjadi *disable*. Antarmuka halaman *network interface* dapat dilihat pada Gambar 4.2.



**Gambar 4.2** Antarmuka Halaman *Network Interface*

#### 4.1.3 Tampilan Antarmuka Halaman *Traffic*

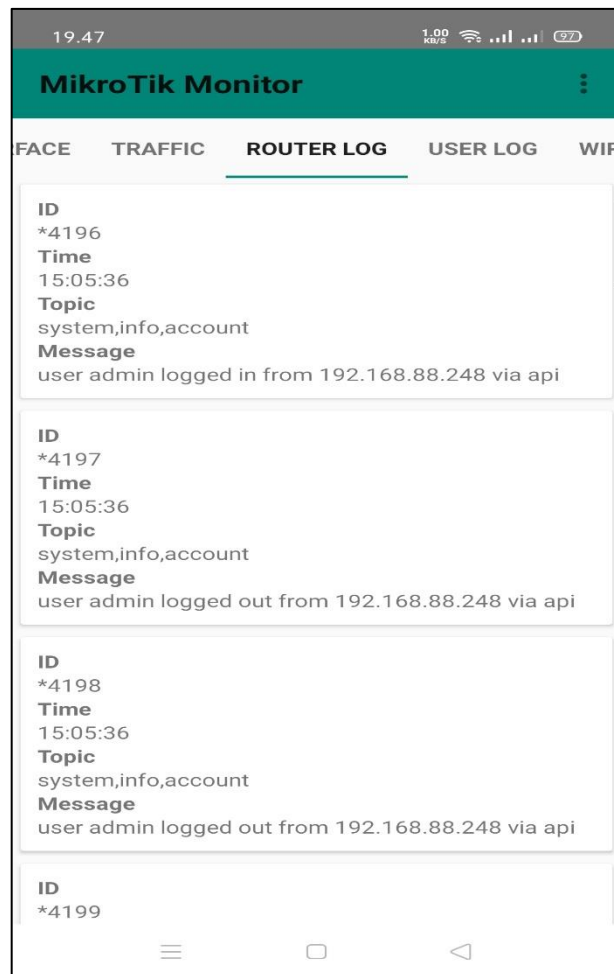
Pada tampilan halaman *traffic* menampilkan kecepatan upload dan *download* dari router mikrotik. Antarmuka halaman *traffic* dapat dilihat pada Gambar 4.3.



**Gambar 4.3** Antarmuka Halaman *Traffic*

#### **4.1.4 Tampilan Antarmuka Halaman Router Log**

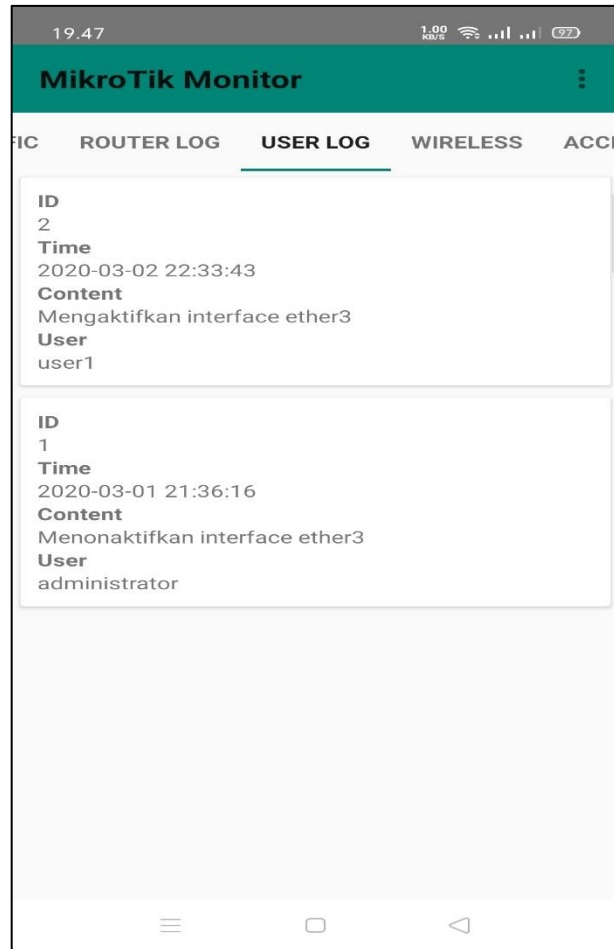
Pada tampilan halaman router *log* menampilkan *history* dari router mikrotik. Antarmuka halaman router *log* dapat dilihat pada Gambar 4.4.



**Gambar 4.4** Antarmuka Halaman Router Log

#### 4.1.5 Tampilan Antarmuka Halaman User Log

Pada tampilan halaman *user log* menampilkan *history* dari aplikasi. Antarmuka halaman *user log* dapat dilihat pada Gambar 4.5.

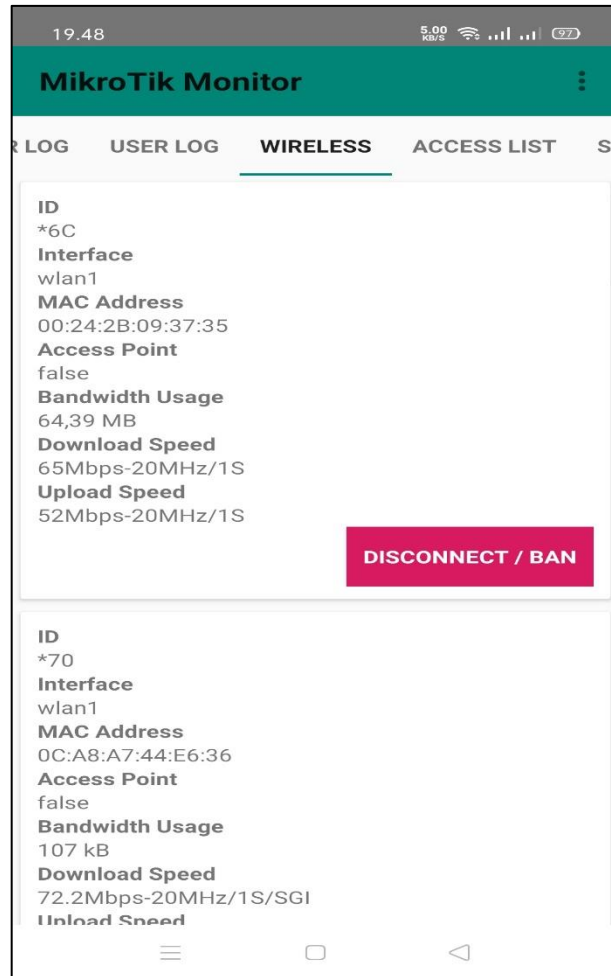


**Gambar 4.5** Antarmuka Halaman *User Log*

#### 4.1.6 Tampilan Antarmuka Halaman *Wireless*

Pada tampilan halaman *wireless* pengguna dapat melihat daftar *user* yang terkoneksi pada mikrotik melalui *wireless* dan dapat memutuskan koneksi atau membaned *user*. Antarmuka halaman beranda *wireless* dapat dilihat pada Gambar 4.6.

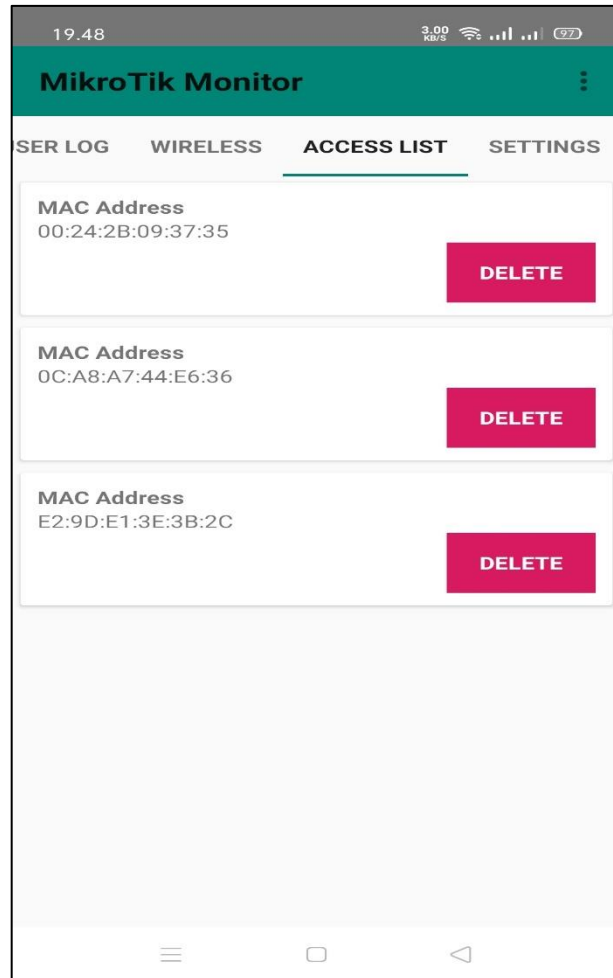




**Gambar 4.6** Antarmuka Halaman *Wireless*

#### 4.1.7 Tampilan Antarmuka *Access list*

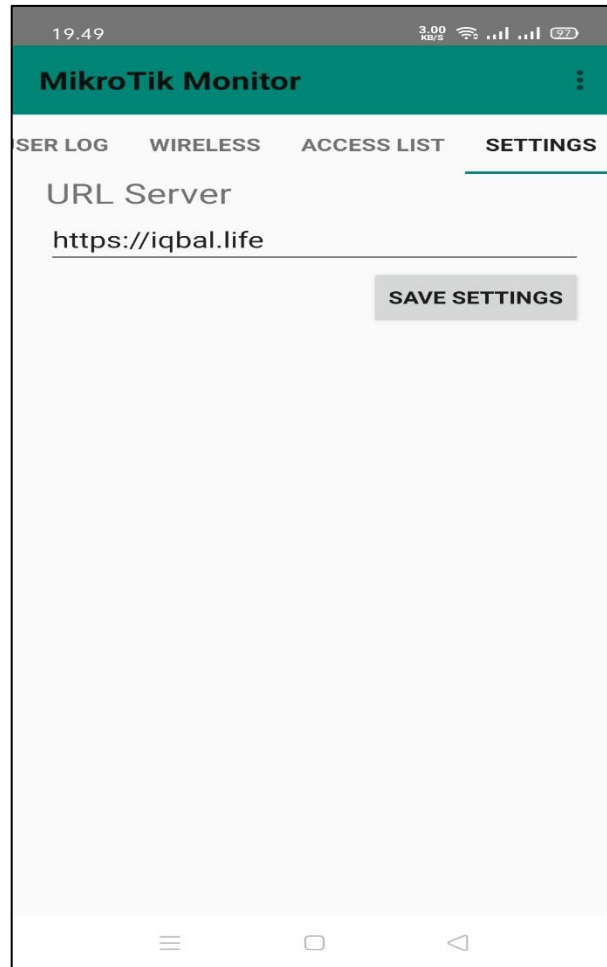
Pada halaman *access list* dapat melihat data *client* yang di baned atau disconnect dan dapat mengembalikan koneksi atau menghapus dari *access list*. Tampilan antarmuka halaman *access list* dapat dilihat pada Gambar 4.7.



**Gambar 4.7** Antarmuka Halaman *Access list*

#### 4.1.8 Tampilan Antarmuka Halaman *Settings*

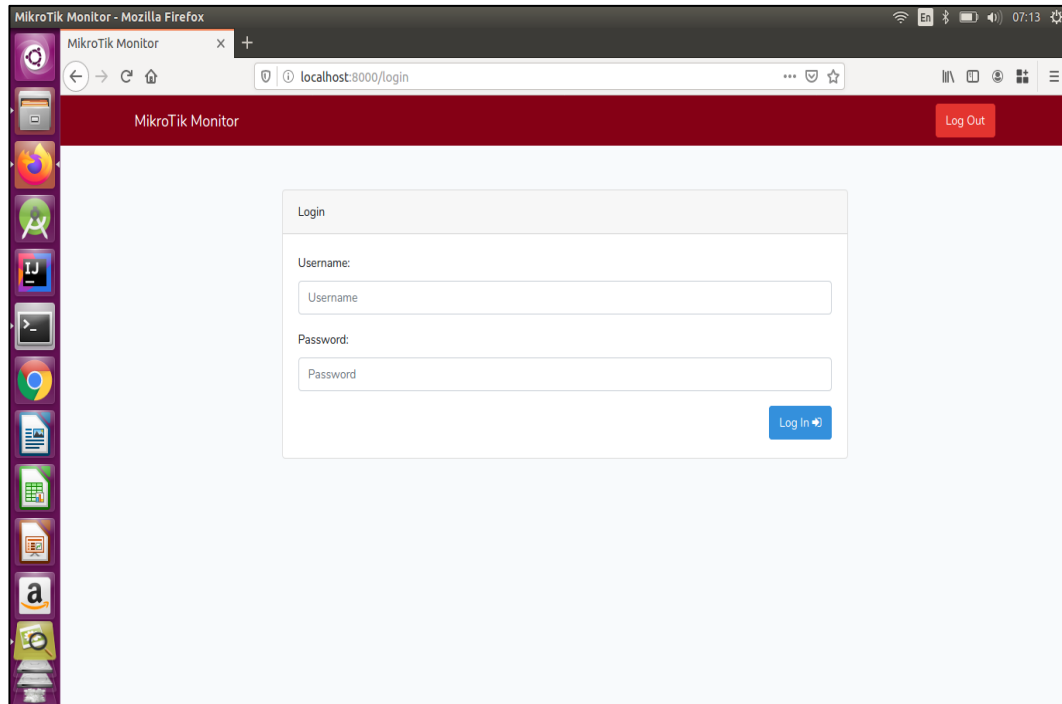
Pada tampilan halaman *settings* menampilkan informasi alamat URL server dan dapat mengganti alamat URL server yang baru. Tampilan antarmuka *settings* dapat dilihat pada Gambar 4.8.



**Gambar 4.8** Antarmuka Halaman *Settings*

#### **4.1.9 Tampilan Antarmuka Halaman *Login* Web**

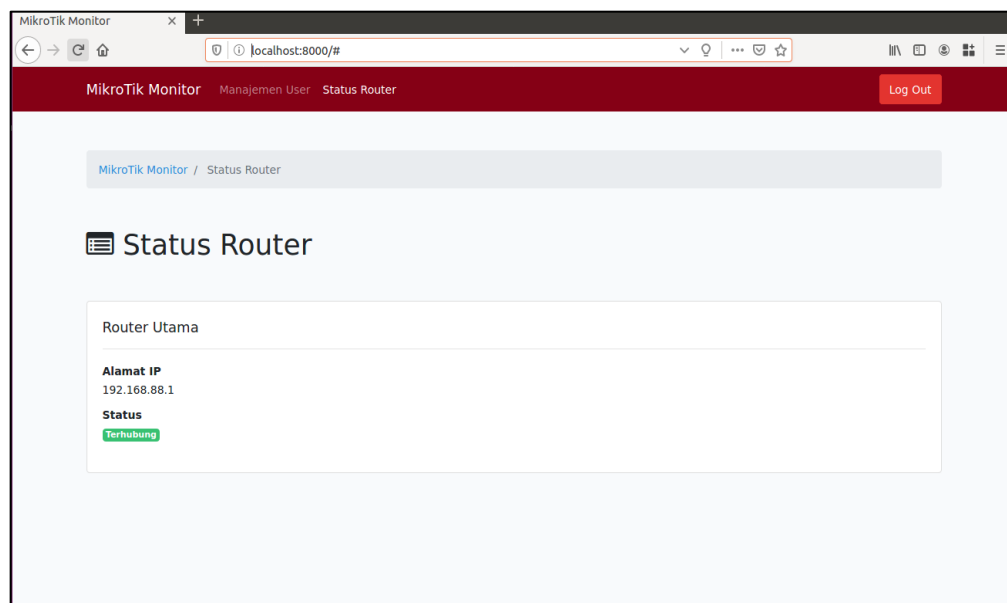
Pada tampilan halaman *login* web yang dapat melakukan *login* pada web hanya super admin dengan memasukkan *username* dan *password*. Tampilan antarmuka *login* web dapat dilihat pada Gambar 4.9.



**Gambar 4.9** Antarmuka Halaman *Login* Web

#### 4.1.10 Tampilan Antarmuka Halaman Status Router

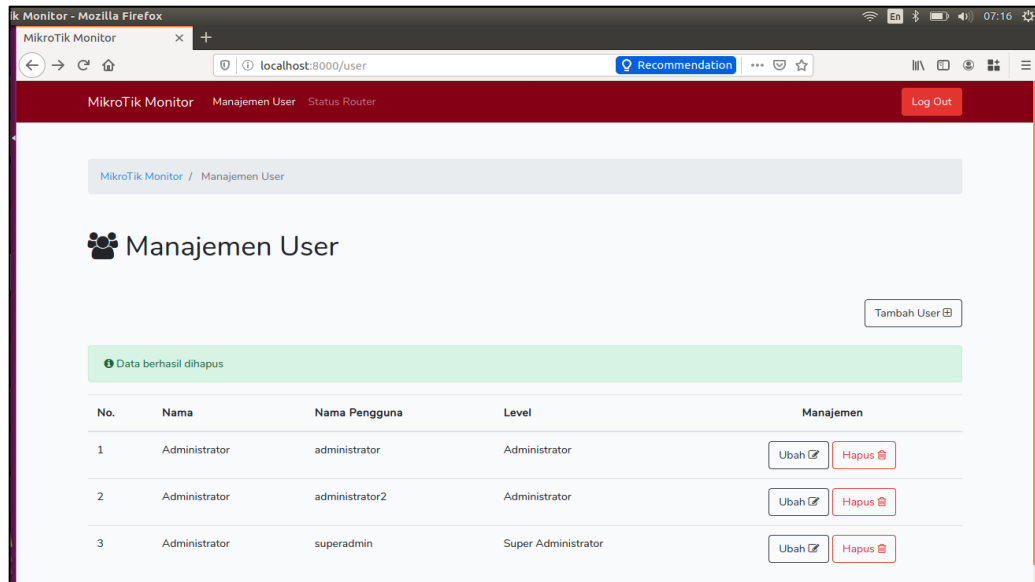
Pada tampilan halaman Status Router menampilkan informasi status router yaitu alamat ip router dan status router apakah terhubung atau tidak. Tampilan antarmuka status router dapat dilihat pada Gambar 4.10.



**Gambar 4.10** Antarmuka Status Router

#### 4.1.11 Tampilan Antarmuka Halaman Manajemen *User*

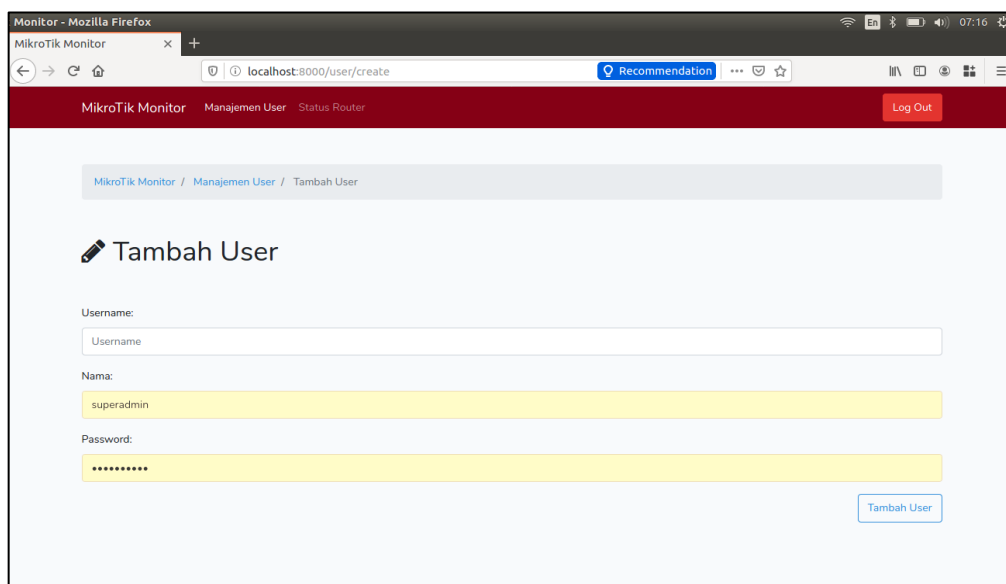
Pada tampilan halaman manajemen *user* menampilkan informasi siapa saja yang dapat *login* pada aplikasi. Tampilan antarmuka manajemen *user* dapat dilihat pada Gambar 4.11.



**Gambar 4.11** Antarmuka Halaman Manajemen *User*

#### 4.1.12 Tampilan Antarmuka Halaman Tambah *User*

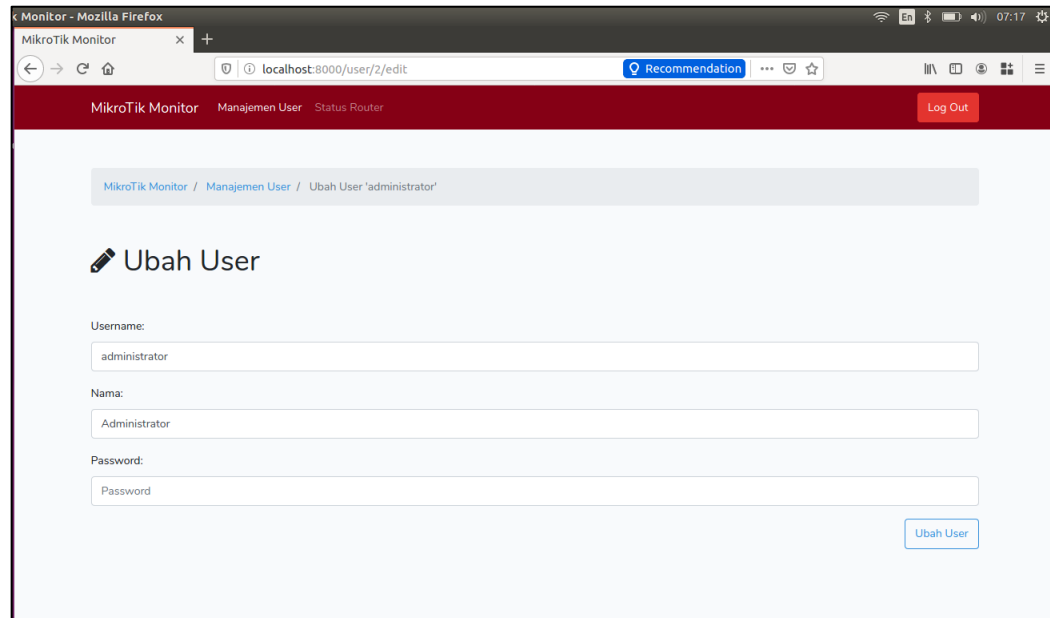
Pada tampilan halaman tambah *user* super admin dapat menambah *user* yang dapat masuk pada aplikasi. Tampilan antarmuka tambah *user* dapat dilihat pada Gambar 4.12.



**Gambar 4.12** Antarmuka Halaman Tambah *User*

#### 4.1.13 Tampilan Antarmuka Halaman Ubah *User*

Pada tampilan halaman ubah *user* super admin dapat mengubah data admin atau hak akses aplikasi. Tampilan antarmuka ubah *user* dapat dilihat pada Gambar 4.13.



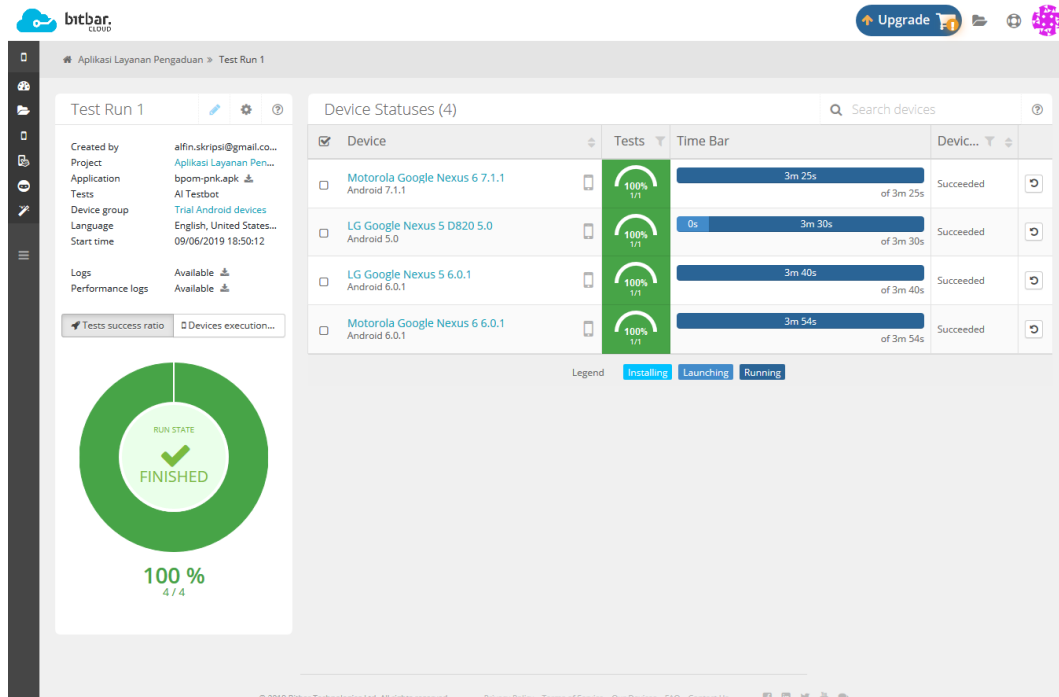
**Gambar 4.13** Antarmuka Halaman Tambah *User*

## 4.2 Hasil Pengujian

Aplikasi layanan pengaduan masyarakat menggunakan dua pengujian, yang pertama pengujian *black box* dan yang kedua pengujian kuesioner.

### 4.2.1 Hasil Pengujian *Black box*

Pengujian dilakukan untuk mengetes fungsionalitas, performa, dan memori yang terdapat dalam penelitian yang dilakukan. Teknik pengujian *black box* yang digunakan yaitu performance testing dan comparison testing. Alat yang digunakan untuk pengujian *Black box* yaitu Bitbar Testing dengan menggunakan AI Testbot yang merupakan pengujian otomatis berbasis *Artificial Intelligent*. Berikut adalah tahapan pengujian yang telah dilakukan.

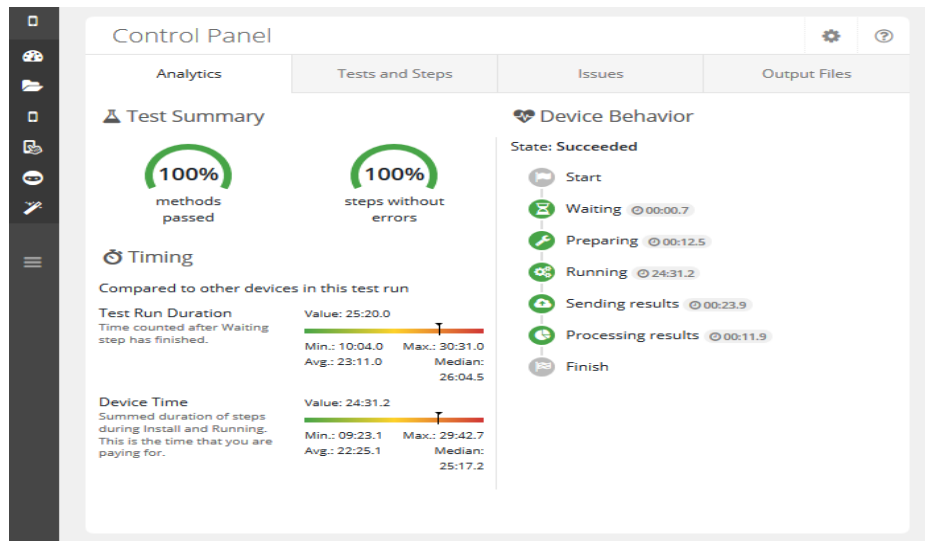


**Gambar 4.14** Pengujian *Black Box* Berhasil

Pengujian dilakukan dalam empat perangkat yaitu Motorola Google Nexus 6 Android versi 7.1.1 dan versi 6.0.1, LG Google Nexus 5 D820, dan LG Google Nexus 5. Bagian yang diujikan adalah fungsional, performa, dan memori, berikut adalah penjabaran hasil pengujiannya.

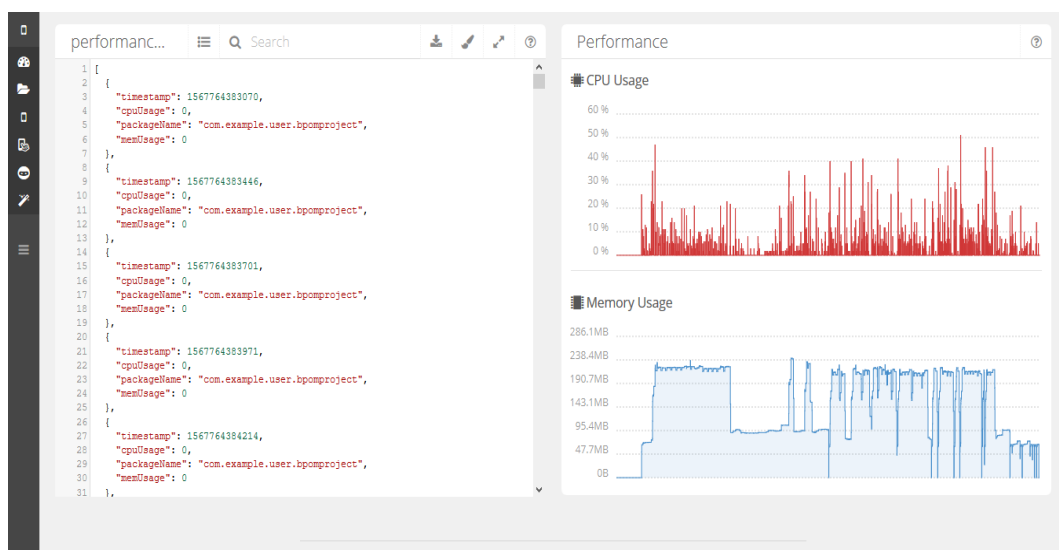
#### 4.2.1.1 Motorola Google Nexus 6 7.1.1

Pengujian pada perangkat yaitu Motorola Google Nexus 6 yang menggunakan android versi 7.1.1 telah dilakukan melalui Bitbar Testing. Pada halaman awal yaitu analisis dari pengujian metode yaitu 100%, menyatakan seluruh metode di dalam aplikasi dapat berjalan semua. Selanjutnya adalah pengujian tahapan tanpa mengalami errors yaitu 100%, menyatakan seluruh tahapan yang dilakukan aplikasi berjalan tanpa gangguan. Berikut merupakan hasil pengujian pada Gambar 4.15.



**Gambar 4.15** Hasil Pengujian Analisis Motorola Google Nexus 6 7.1.1

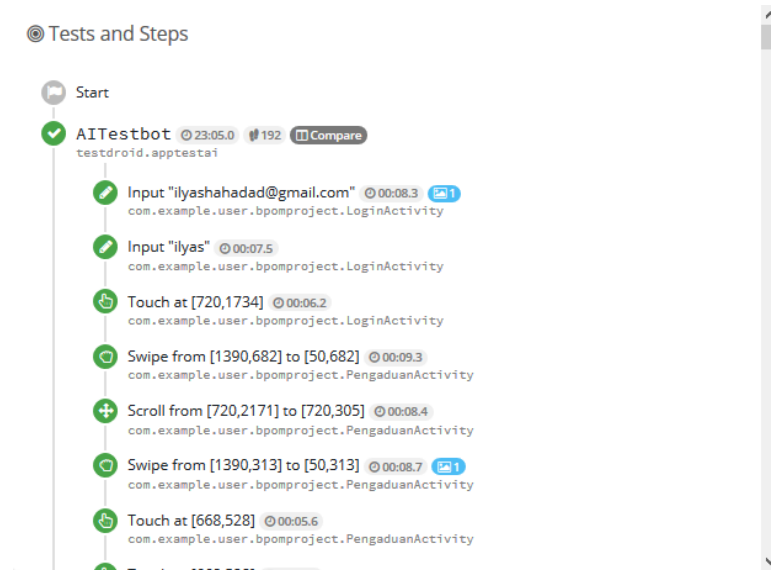
Pengujian performa atau penggunaan CPU yaitu dalam kisaran 51% dan penggunaan memori yaitu dalam kisaran 238,4MB. Berdasarkan hasil pengujian penggunaan memori cukup tinggi pada perangkat, dapat dilihat dalam Gambar 4.16.



**Gambar 4.16** Hasil Pengujian Performa Motorola Google Nexus 6 7.1.1

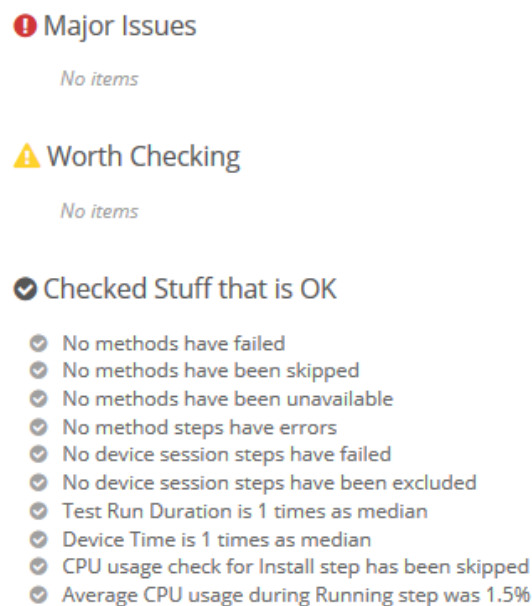
Langkah pengujian dan dokumentasi dari pengujian dapat dilihat dalam Gambar 4.17.





**Gambar 4.17** Hasil Pengujian Langkah Motorola Google Nexus 6 7.1.1

Hasil pengujian memperlihatkan tidak ada permasalahan besar dan tidak ada hal yang sebaiknya diperiksa kembali pada aplikasi. Hasil pengujian masalah pada aplikasi dapat dilihat dalam Gambar 4.18.

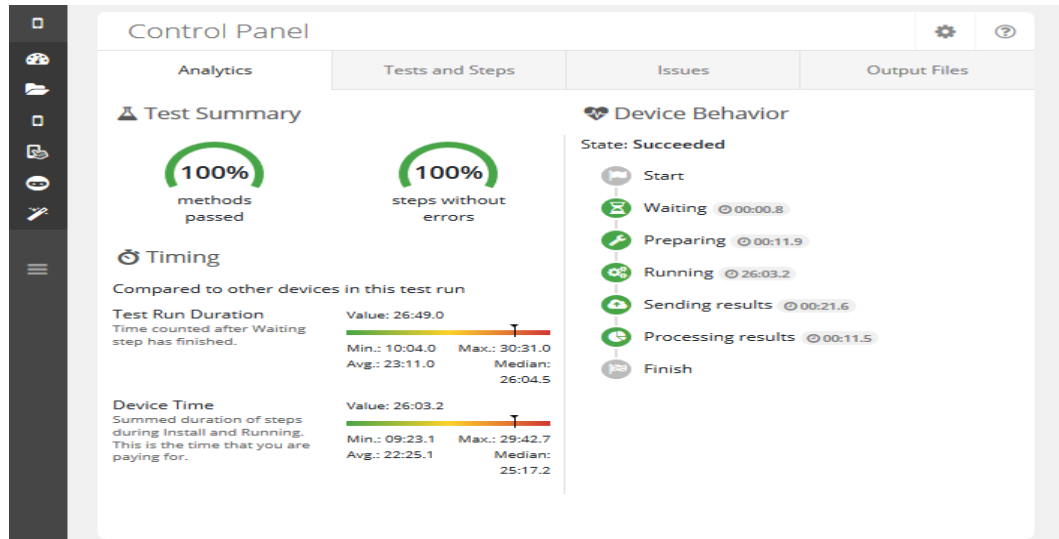


**Gambar 4.18** Hasil Pengujian Permasalahan Motorola Google Nexus 6 7.1.1

#### 4.2.1.2 Motorola Google Nexus 6 6.0.1

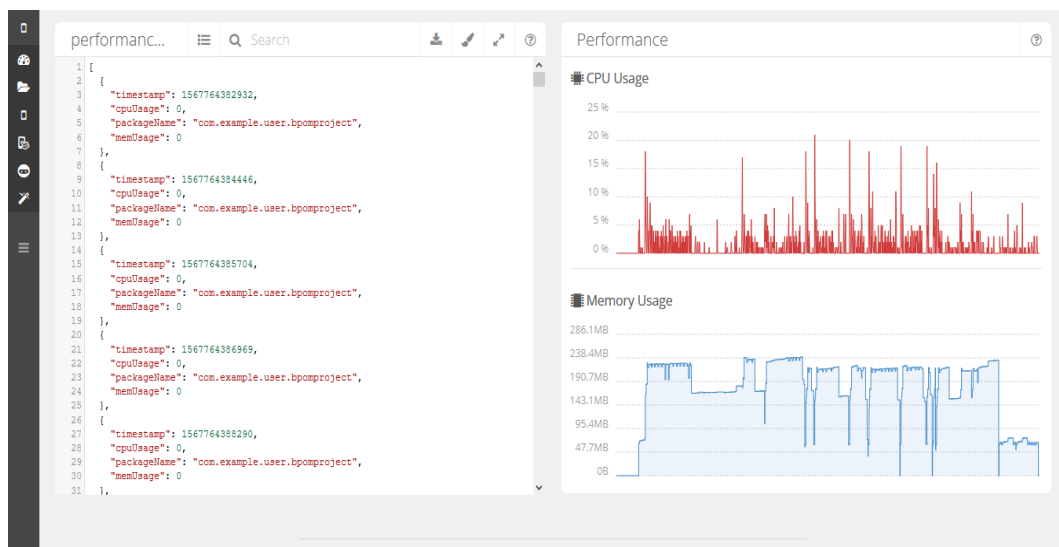
Pengujian pada perangkat yaitu Motorola Google Nexus 6 yang menggunakan android versi 6.0.1 telah dilakukan melalui Bitbar Testing. Pada halaman awal yaitu analisis dari pengujian metode yaitu 100%, menyatakan seluruh

metode di dalam aplikasi dapat berjalan semua. Selanjutnya adalah pengujian tahapan tanpa mengalami errors yaitu 100%, menyatakan seluruh tahapan yang dilakukan aplikasi berjalan tanpa gangguan. Berikut merupakan hasil pengujian pada Gambar 4.19.



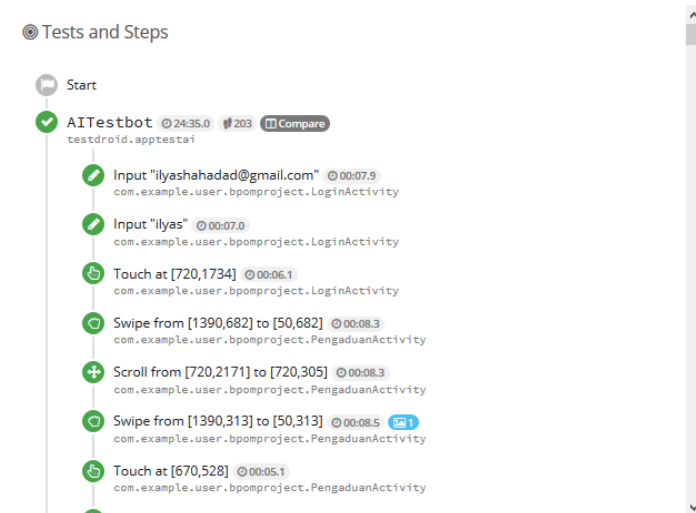
**Gambar 4.19** Hasil Pengujian Analisis Motorola Google Nexus 6 6.0.1

Pengujian performa atau penggunaan CPU yaitu dalam kisaran 21% dan penggunaan memori yaitu dalam kisaran 238,4MB. Berdasarkan hasil pengujian penggunaan memori cukup tinggi pada perangkat, dapat dilihat dalam Gambar 4.20.



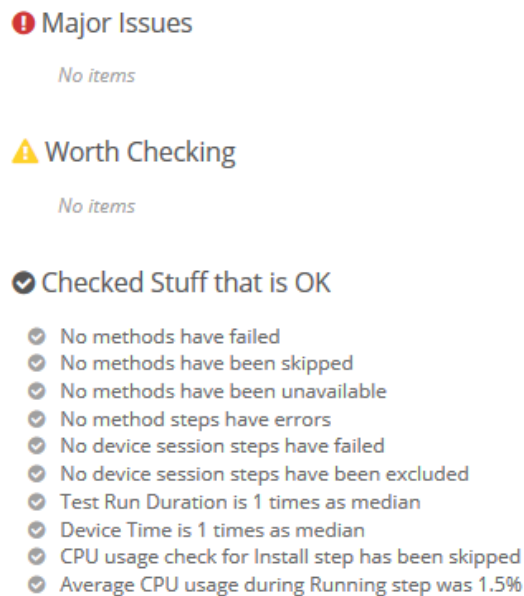
**Gambar 4.20** Hasil Pengujian Performa Motorola Google Nexus 6 6.0.1

Langkah pengujian dan dokumentasi dari pengujian dapat dilihat dalam Gambar 4.21.



**Gambar 4.21** Hasil Pengujian Langkah Motorola Google Nexus 6 6.0.1

Hasil pengujian memperlihatkan tidak ada permasalahan besar dan tidak ada hal yang sebaiknya diperiksa kembali pada aplikasi. Hasil pengujian masalah pada aplikasi dapat dilihat dalam Gambar 4.22.

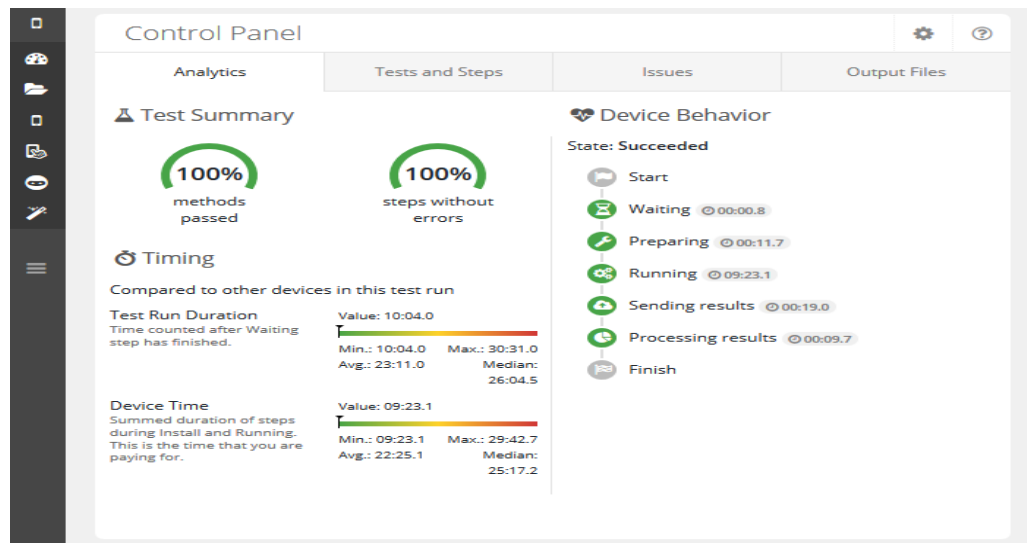


**Gambar 4.22** Hasil Pengujian Permasalahan Motorola Google Nexus 6 6.0.1

#### 4.2.1.3 LG Google Nexus 5

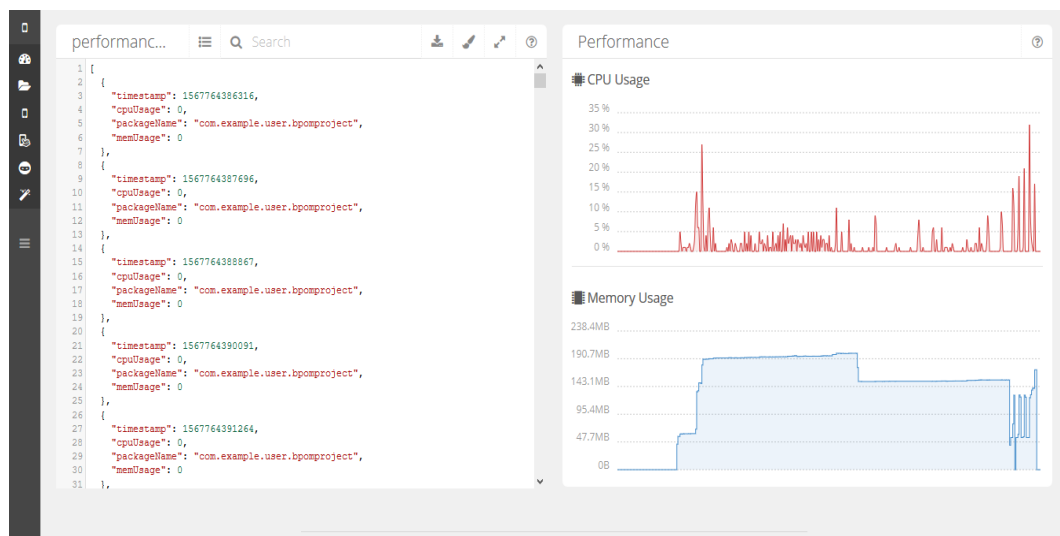
Pengujian pada perangkat yaitu LG Google Nexus 5 yang menggunakan android versi 6.0.1 telah dilakukan melalui Bitbar Testing. Pada halaman awal yaitu analisis dari pengujian metode yaitu 100%, menyatakan seluruh metode di dalam aplikasi dapat berjalan semua. Selanjutnya adalah pengujian tahapan tanpa

mengalami errors yaitu 100%, menyatakan seluruh tahapan yang dilakukan aplikasi berjalan tanpa gangguan. Berikut merupakan hasil pengujian pada Gambar 4.23.



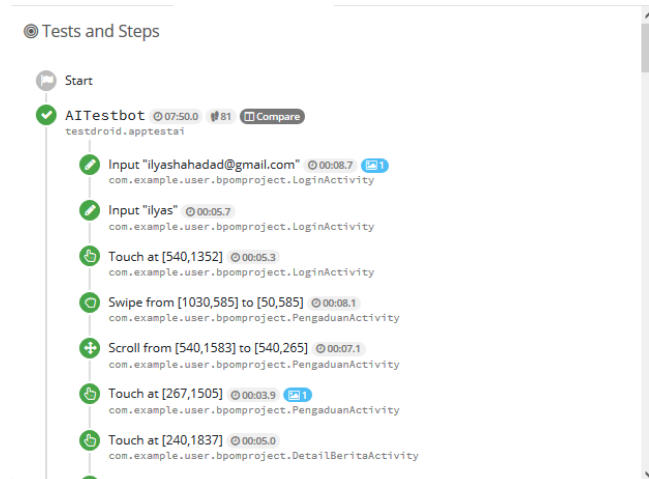
**Gambar 4.23** Hasil Pengujian Analisis LG Google Nexus 5

Pengujian performa atau penggunaan CPU yaitu dalam kisaran 32% dan penggunaan memori yaitu dalam kisaran 190,7MB. Berdasarkan hasil pengujian penggunaan memori cukup tinggi pada perangkat, dapat dilihat dalam Gambar 4.24.



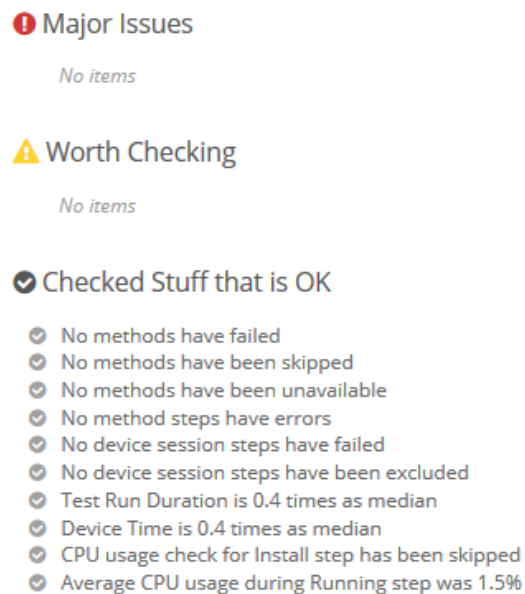
**Gambar 4.24** Hasil Pengujian Performa LG Google Nexus 5

Langkah pengujian dan dokumentasi dari pengujian dapat dilihat dalam Gambar 4.25.



**Gambar 4.25** Hasil Pengujian Langkah LG Google Nexus 5

Hasil pengujian memperlihatkan tidak ada permasalahan besar dan tidak ada hal yang sebaiknya diperiksa kembali pada aplikasi. Hasil pengujian masalah pada aplikasi dapat dilihat dalam Gambar 4.26.

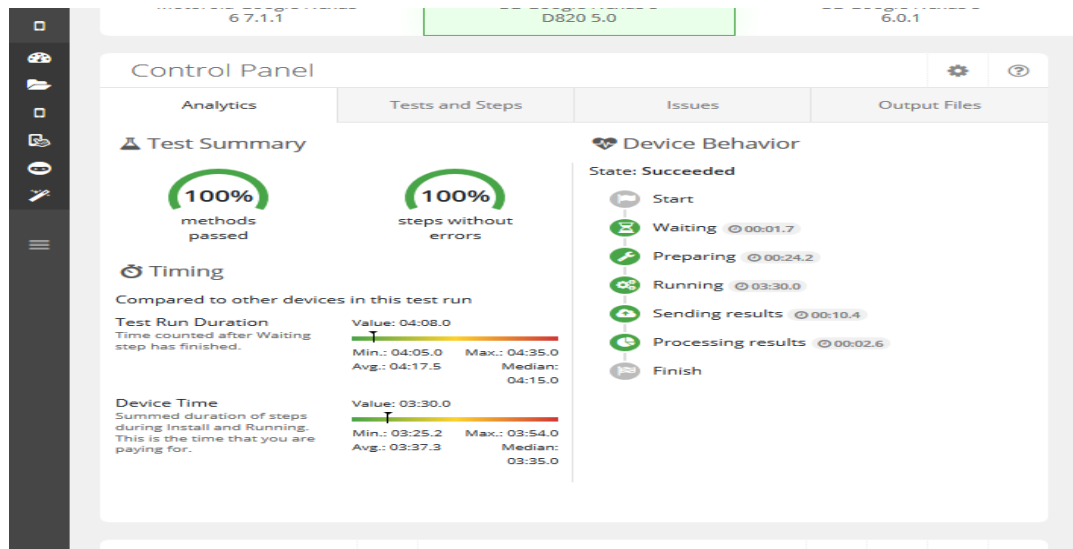


**Gambar 4.26** Hasil Pengujian Permasalahan LG Google Nexus 5

#### 4.2.1.4 LG Google Nexus 5 D820

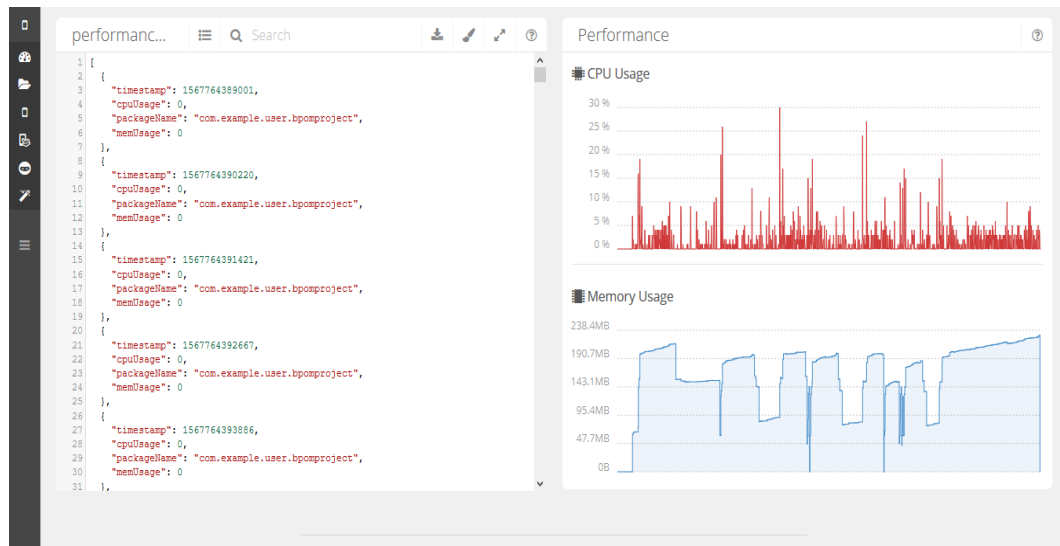
Pengujian pada perangkat yaitu LG Google Nexus 5 D820 yang menggunakan android versi 5.0 telah dilakukan melalui Bitbar Testing. Pada halaman awal yaitu analisis dari pengujian metode yaitu 100%, menyatakan seluruh metode di dalam aplikasi dapat berjalan semua. Selanjutnya adalah pengujian tahapan tanpa mengalami errors yaitu 100%, menyatakan seluruh tahapan yang

dilakukan aplikasi berjalan tanpa gangguan. Berikut merupakan hasil pengujian pada Gambar 4.27.



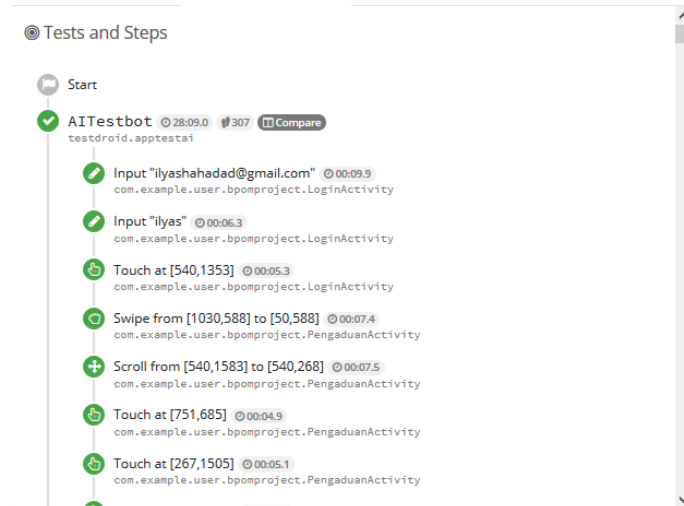
**Gambar 4.27** Hasil Pengujian Analisis LG Nexus 5 D820

Pengujian performa atau penggunaan CPU yaitu dalam kisaran 30% dan penggunaan memori yaitu dalam kisaran 238MB. Berdasarkan hasil pengujian penggunaan memori cukup tinggi pada perangkat, dapat dilihat dalam Gambar 4.28.



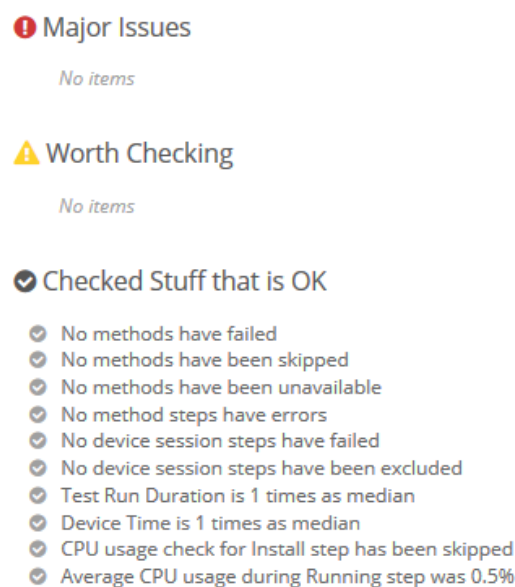
**Gambar 4.28** Hasil Pengujian Performa LG Nexus 5 D820

Langkah pengujian dan dokumentasi dari pengujian dapat dilihat dalam Gambar 4.29.



**Gambar 4.29** Hasil Pengujian Langkah LG Nexus 5 D820

Hasil pengujian memperlihatkan tidak ada permasalahan besar dan tidak ada hal yang sebaiknya diperiksa kembali pada aplikasi. Hasil pengujian masalah pada aplikasi dapat dilihat dalam Gambar 4.30.



**Gambar 4.30** Hasil Pengujian Permasalahan LG Nexus 5 D820

#### 4.2.2 Hasil Pengujian Kuesioner

Kuesioner yang dibagikan terdiri dari beberapa pertanyaan yang dikelompokkan menjadi tiga aspek, yaitu aspek rekayasa perangkat lunak, aspek fungsionalitas, dan aspek komunikasi visual. Untuk penilaian kuesioner diberikan

nilai 1 untuk tidak baik, 2 untuk kurang baik, 3 untuk cukup baik, 4 untuk baik, dan 5 untuk sangat baik.

#### 4.2.2.1 Latar Belakang Peserta Kuesioner

Dalam pengujian beta, dilakukan pengujian dengan menggunakan 10, yaitu orang yang pernah menjadi administrator jaringan yang ada di Lab Informatika Untan dan Gedung Untannet.

#### 4.2.2.2 Perhitungan Skala Likert

Hasil kuesioner perhitungan skala likert sesuai dengan aspek/pertanyaan masing-masing responden ditunjukkan pada Tabel 4.1.

**Tabel 4.1** Hasil Kuesioner Perhitungan Skala Likert

No.	Aspek / Pertanyaan	Nilai Pengujian					Persentase Likert
		TB	KB	CB	B	SB	
Aspek Rekayasa Perangkat Lunak							
1.	Kemudahan menjalankan aplikasi	0	0	0	7	3	86 %
2.	Kompabilitas aplikasi pada perangkat	0	0	0	10	0	80 %
3.	Kelancaran menjalankan aplikasi pada perangkat	0	0	0	4	6	92 %
4.	Kemudahan mengakses fitur-fitur pada aplikasi	0	0	0	7	3	86 %
5.	Kenyamanan dalam menggunakan aplikasi secara keseluruhan	0	0	0	9	1	82 %
Aspek Fungsionalitas							
6.	Kemudahan dalam melakukan <i>login</i> aplikasi	0	0	0	1	9	98 %
7.	Kemudahan menjalankan tombol pada aplikasi	0	0	1	7	2	82 %
8.	Tingkat kesesuaian menampilkan <i>input</i> , proses, dan <i>output</i> dari pada aplikasi android	0	0	0	10	0	80 %
9.	Tingkat kesesuaian aplikasi dalam	0	0	1	7	2	82 %



No.	Aspek / Pertanyaan	Nilai Pengujian					Persentase Likert
		TB	KB	CB	B	SB	
	menampilkan informasi berupa data pada <i>user</i> yang terhubung						
10.	Tingkat kemudahan dalam memonitoring besaran bandwidth pada setiap <i>user</i>	0	0	1	6	3	84 %
<b>Aspek Komunikasi Visual</b>							
11.	Kemudahan memahami informasi yang ditampilkan pada aplikasi	0	0	3	6	1	76 %
12.	Tampilan antarmuka aplikasi	0	0	0	10	0	80 %
13.	Jenis dan ukuran huruf yang digunakan mudah dibaca	0	0	0	9	1	82 %
14.	Kombinasi warna pada tampilan aplikasi	0	0	1	9	0	78 %
15.	Tingkat kemudahan melihat hasil yang ditampilkan aplikasi	0	0	0	9	1	82 %
<b>Total Persentase (%)</b>							<b>83,33 %</b>

Nilai persentase skala likert dihitung dengan rumus berikut ini:

$$L = \frac{(SB \times 5) + (B \times 4) + (CB \times 3) + (KB \times 2) + (TB \times 1)}{NB} \times 100$$

Hasil persentase menggunakan skala likert sebagai berikut:

Pertanyaan 1 :

$$L = \frac{(3 \times 5) + (7 \times 4) + (0 \times 3) + (0 \times 2) + (0 \times 1)}{5 \times 10} \times 100 = 86 \%$$

Pertanyaan 2 :

$$L = \frac{(0 \times 5) + (10 \times 4) + (0 \times 3) + (0 \times 2) + (0 \times 1)}{5 \times 10} \times 100 = 80 \%$$

Pertanyaan 3 :

$$L = \frac{(6 \times 5) + (4 \times 4) + (0 \times 3) + (0 \times 2) + (0 \times 1)}{5 \times 10} \times 100 = 92 \%$$

Pertanyaan 4 :

$$L = \frac{(3 \times 5) + (7 \times 4) + (0 \times 3) + (0 \times 2) + (0 \times 1)}{5 \times 10} \times 100 = 86 \%$$

Pertanyaan 5 :

$$L = \frac{(1 \times 5) + (9 \times 4) + (0 \times 3) + (0 \times 2) + (0 \times 1)}{5 \times 10} \times 100 = 82 \%$$

Pertanyaan 6 :

$$L = \frac{(9 \times 5) + (1 \times 4) + (0 \times 3) + (0 \times 2) + (0 \times 1)}{5 \times 10} \times 100 = 98 \%$$

Pertanyaan 7 :

$$L = \frac{(2 \times 5) + (7 \times 4) + (1 \times 3) + (0 \times 2) + (0 \times 1)}{5 \times 10} \times 100 = 82 \%$$

Pertanyaan 8 :

$$L = \frac{(0 \times 5) + (10 \times 4) + (0 \times 3) + (0 \times 2) + (0 \times 1)}{5 \times 10} \times 100 = 80 \%$$

Pertanyaan 9 :

$$L = \frac{(2 \times 5) + (7 \times 4) + (1 \times 3) + (0 \times 2) + (0 \times 1)}{5 \times 10} \times 100 = 82 \%$$

Pertanyaan 10 :

$$L = \frac{(3 \times 5) + (6 \times 4) + (1 \times 3) + (0 \times 2) + (0 \times 1)}{5 \times 10} \times 100 = 84 \%$$

Pertanyaan 11 :

$$L = \frac{(1 \times 5) + (6 \times 4) + (3 \times 3) + (0 \times 2) + (0 \times 1)}{5 \times 10} \times 100 = 76 \%$$

Pertanyaan 12 :

$$L = \frac{(0 \times 5) + (10 \times 4) + (0 \times 3) + (0 \times 2) + (0 \times 1)}{5 \times 10} \times 100 = 80 \%$$

Pertanyaan 13 :

$$L = \frac{(1 \times 5) + (9 \times 4) + (0 \times 3) + (0 \times 2) + (0 \times 1)}{5 \times 10} \times 100 = 82 \%$$

Pertanyaan 14 :

$$L = \frac{(0 \times 5) + (9 \times 4) + (1 \times 3) + (0 \times 2) + (0 \times 1)}{5 \times 10} \times 100 = 78 \%$$

Pertanyaan 15 :

$$L = \frac{(1 \times 5) + (9 \times 4) + (0 \times 3) + (0 \times 2) + (0 \times 1)}{5 \times 10} \times 100 = 82 \%$$

$$\begin{aligned}
 \text{Total Persentase} &= (86 + 80 + 92 + 86 + 82 + 98 + 82 + 80 + 82 + 84 + 76 + 80 + \\
 &\quad 82 + 78 + 82) / 15 \\
 &= 1250 / 15 \\
 &= 83,33 \%
 \end{aligned}$$

Dari hasil kuesioner yang telah dilaksanakan terhadap 10 responden didapatkan nilai dari perhitungan skala likert dari total 15 pertanyaan yang diberikan kepada responden adalah 83,33 %.

Nilai interval persentase pada kuesioner ini yaitu:

- $X \leq 19,99\%$  dikategorikan tidak baik
- $20\% \geq X \leq 39,99\%$  dikategorikan kurang baik
- $40\% \geq X \leq 59,99\%$  dikategorikan cukup baik
- $60\% \geq X \leq 79,99\%$  dikategorikan baik
- $X \geq 80\%$  dikategorikan sangat baik

Kesimpulan hasil akhir dari kuesioner pada aplikasi manajemen jaringan mikrotik berbasis android, dari nilai persentase interval mendapatkan hasil aplikasi tersebut dikategorikan sangat baik

### 4.3 Analisis Hasil Perancangan dan Pengujian

Berikut ini adalah hasil analisis hasil perancangan dan pengujian perangkat lunak aplikasi permainan dalam penelitian:

1. Hasil pengujian fungsional yang dilakukan oleh sistem oleh perusahaan BitBar menunjukkan semua fungsi yang ada dalam aplikasi dapat berjalan semua dengan lancar.
2. Hasil pengujian memori dan CPU *usage* yang dilakukan oleh sistem dari perusahaan BitBar menunjukkan aplikasi manajemen jaringan mikrotik menggunakan banyak memori untuk menjalankan aplikasi.
3. Berdasarkan hasil kuesioner, secara umum responden menerima baik aplikasi manajemen jaringan mikrotik dalam penelitian yang dilakukan.
4. Berdasarkan hasil kuesioner, secara umum responden dapat menjalankan aplikasi layanan pengaduan dengan lancar.

5. Berdasarkan hasil kuesioner, mendapatkan kesimpulan bahwa aplikasi manajemen jaringan mikrotik dapat membantu administrator jaringan dalam memanajemen mikrotik melalui aplikasi dan dinilai berhasil.
6. Hasil pengujian kuesioner yang diukur dengan perhitungan skala likert menunjukkan responden menilai aplikasi yang dibangun dengan “sangat baik” dengan skor presentase 83,33%.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan hasil analisis dan pengujian terhadap aplikasi manajemen jaringan mikrotik berbasis android yang dibuat untuk Lab Informatika Untan, maka dapat ditarik kesimpulan sebagai berikut:

1. Aplikasi manajemen jaringan mikrotik berbasis android *mobile* ini telah dirancang dan dibuat dengan baik serta berjalan dengan lancar dalam perangkat android dengan menggunakan Bitbar Testing. Melalui kuesioner aplikasi juga mendapat hasil yang baik pada kuesioner dihitung dengan menggunakan Skala Likert.
2. Penerapan aplikasi manajemen jaringan mikrotik dinilai berhasil yang ditunjukkan berdasarkan pada perhitungan skala likert yang berasal dari kuesioner yang diberikan kepada 10 responden untuk menilai aplikasi. Total presentase yang didapat dari 15 pertanyaan adalah 83,33% yang artinya mempunyai nilai “sangat baik”.
3. Aplikasi manajemen jaringan mikrotik yang dibangun sebagai media untuk membantu administrator jaringan untuk memanajemen mikrotik di Lab Informatika Untan berjalan dengan baik berdasarkan pengujian oleh sistem Bitbar Testing yaitu 100% metode berjalan dengan baik tanpa *error* dalam perangkat android dengan versi 5.0 (Lollipop) hingga versi 7.1.1 (Nougat) dengan berbagai ukuran resolusi layar.

#### **5.2 Saran**

Adapun beberapa hal yang perlu ditambahkan dalam pengembangan aplikasi ini, adalah sebagai berikut:

1. Membuat aplikasi manajemen jaringan mikrotik berbasis IOS.
2. Menambahkan *notification* pada aplikasi manajemen jaringan mikrotik berbasis android.