

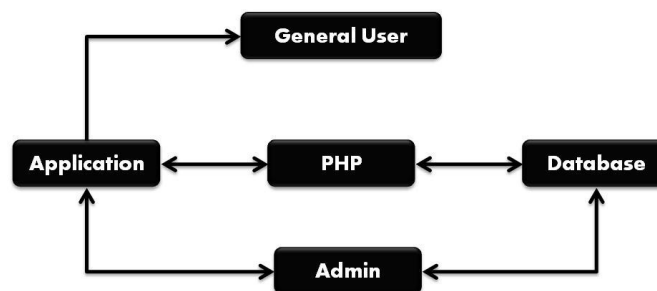
BAB III

PERANCANGAN APLIKASI

Pemodelan sistem Tugas Akhir Aplikasi Pegontrolan Data Inventaris Barang di Laboratorium Komputer ST3 Telkom Purwokerto yang memanfaatkan jaringan *Wi-Fi* dibuat berdasarkan tampilan di *handphone* berbasis android.

3.1 PEMODELAN SISTEM BERDASARKAN TAMPILAN PADA *HANDPHONE* BERBASIS ANROID

Aplikasi kontrol data inventaris barang akan mengambil data dari *database server* dengan bantuan PHP, PHP digunakan untuk menerjemahkan bahasa pemrograman *database* kedalam bahasa pemrograman JAVA, begitu juga sebaliknya ketika aplikasi mengirimkan data ke *database server* akan di jembatani oleh bahasa pemrograman PHP. Aplikasi dan *database server* merupakan *interface* bagi admin untuk mengontrol data didalamnya. Untuk lebih jelasnya perhatikan gambar



Gambar 3.1 Diagram block Prinsip kerja aplikasi

3.1.1 ANALISA KEBUTUHAN SISTEM

Dalam perancangan aplikasi pengontrolan data inventaris barang dibutuhkan sebuah *software* dan *hardware* agar mencapai tujuan yang diharapkan. *Software* dan *hardware* yang diperlukan dalam merancang aplikasi peginputan data inventaris adalah sebagai berikut:

1. *Software*

Dalam perancangan aplikasi ini dibutuhkan suatu *software* (perangkat lunak) agar aplikasi ini dapat digunakan sesuai dengan

kebutuhan dan sesuai dengan tujuan. Perangkat lunak yang dibutuhkan dalam perancangan aplikasi ini adalah sebagai berikut:

a. *Text (JDK)*

Dalam perancangan aplikasi dibutuhkan instalasi JDK karena pada dasarnya android menggunakan bahasa java. Versi JDK yang digunakan dalam perancangan aplikasi ini yaitu versi JDK 7u51.

b. *Integrated Development Environment (IDE) EclipseADT*

Instalasi IDE Eclipse ADT dibutuhkan pada perancangan aplikasi ini. IDE Eclipse ini digunakan pengembang membangun sebuah aplikasi Android karena pada IDE Eclipse disediakan Android plug-in sehingga memudahkan pengembang membuat suatu aplikasi Android dimana *source software* dapat didapatkan secara langsung dari situs resmi Google. Sedangkan untuk Eclipse sendiri dapat di download di <http://www.eclipse.org/downloads/> . Eclipse dijalankan dengan cara memilih *file* bertipe application atau berekstensi ".exe" yang berada pada folder Eclipse. Kemudian akan muncul tampilan seperti gambar 3.2.

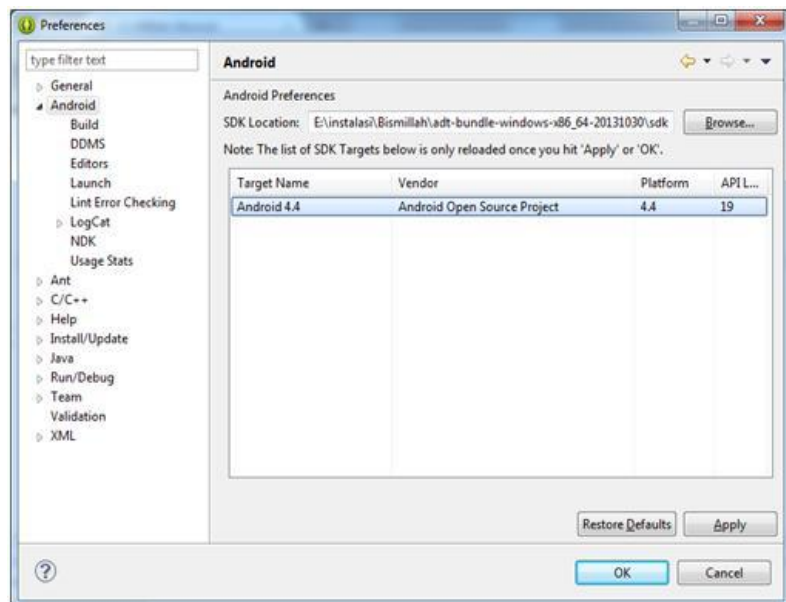


Gambar 3.2 Tampilan Awal Eclipse

c. Android SDK

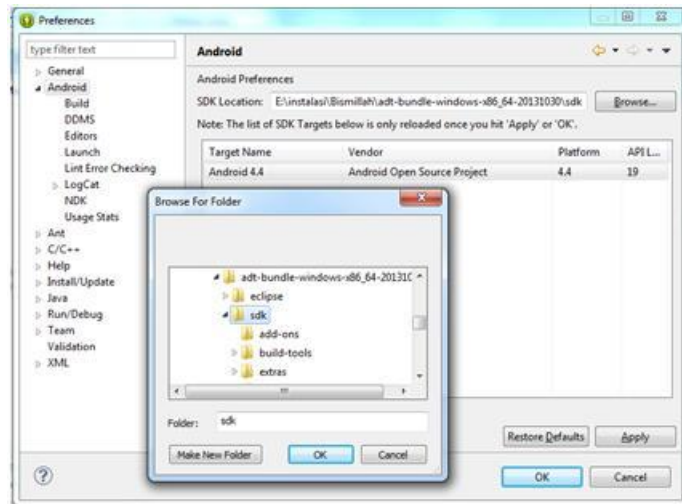
Android SDK dibutuhkan dalam perancangan aplikasi pegontrolan data inventaris karena Android SDK merupakan *tool* pelengkap API dalam mengembangkan aplikasi pada *platform* Android dengan menggunakan bahasa pemrograman Java.[12] Android SDK diinstalasi setelah penginsalan Android ADT. Pemasangan Android SDK pada Eclipse ADT yaitu:

- Pada menu bar Eclipse ADT, memilih menu windows kemudian *preference*, maka akan muncul tampilan seperti gambar 3.3



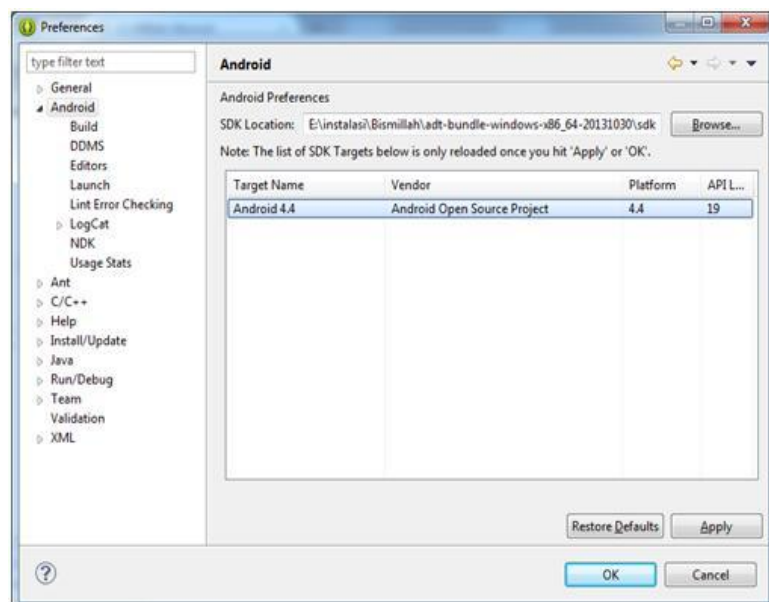
Gambar 3.3 Tampilan Lokasi SDK

- Pada *SDK Location*, setelah itu memilih *browse* kemudian mencari lokasi direktori Android seperti pada gambar 3.4.



Gambar 3.4 Mencari Lokasi Direktori Android

- Kemudian memilih tombol OK. Maka akan muncul seperti pada gambar 3.5. Pada halaman tersebut memilih Android 4.4 yang akan menjadi target aplikasi yang akan dipakai kemudian memilih OK.



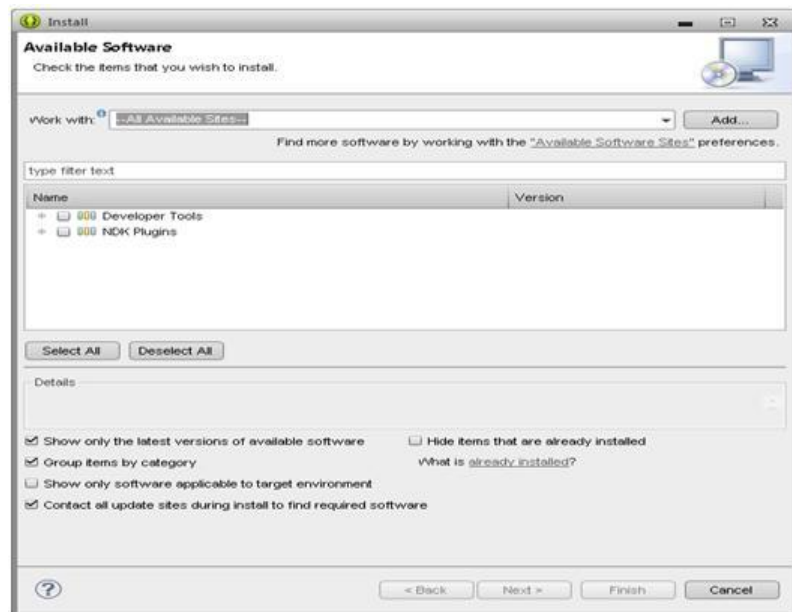
Gambar 3.5 Pemilihan Target SDK

d. ADT

ADT pada perancangan aplikasi ini dibutuhkan untuk membuat proyek berbasis Android. ADT menjadi penghubung antara IDE Eclipse dengan Android SDK[3]. Instalasi ADT

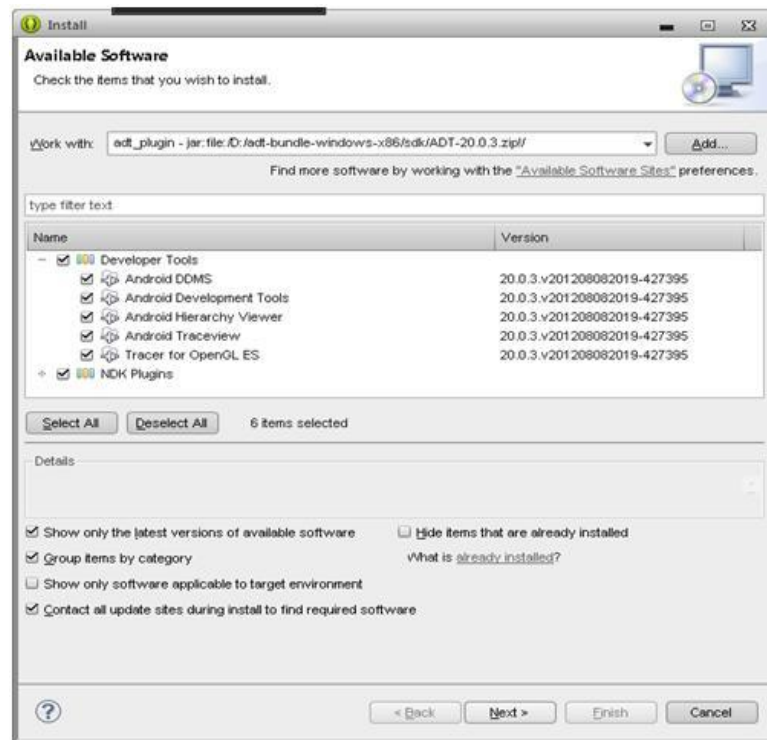
dilakukan dengan menghubungkan koneksi internet. Langkah - langkah instalasi ADT yaitu sebagai berikut:

- Pada menu bar Eclipse memilih menu *help>instal new software*. Maka akan muncul tampilan seperti pada gambar 3.6.



Gambar 3.6 Tampilan ADT Plugins

- Kemudian memilih *add site* dan lokasi diisi dengan <https://dl-ssl.google.com/android/eclipse/>. Jika sudah men-*download* sebelumnya ADT terbaru dapat di browse. Seperti pada gambar 3.7. Kemudian *checklist* pada *developer tools* setelah itu memilih tombol *next*, pada tampilan berikutnya yaitu memilih persetujuan dan kemudian memilih tombol *finish* maka proses instalasi akan berjalan.

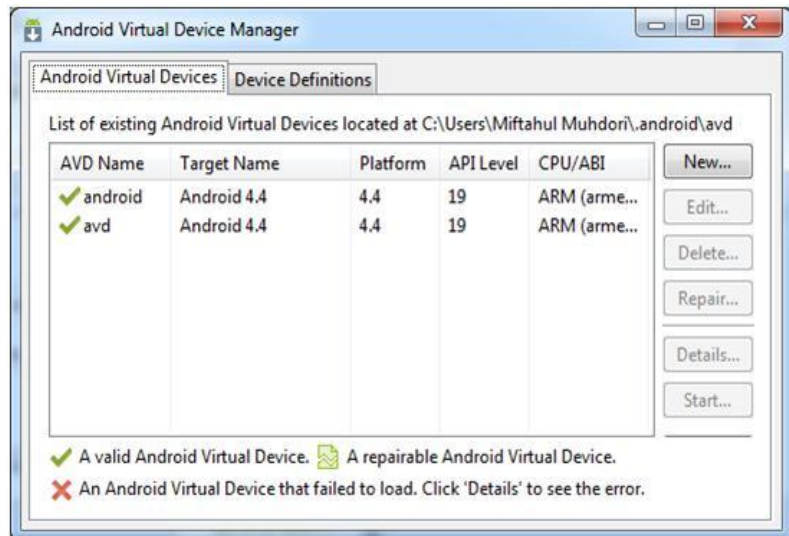


Gambar 3.7 Instalasi ADT

- Pada saat proses instalasi sedang berjalan maka akan muncul dialog perintah melakukan restart kemudian memilih *yes*, agar ADT *plugins* yang baru terinstal terpasang dengan benar pada IDE Eclipse.
 - Proses *restart* pada Eclipse selesai maka muncul kotak dialog *workspace launcher* yang menanyakan *workspace* yang akan digunakan. *Workspace launcher* merupakan letak hasil dari pembuatan proyek nantinya atau lokasi *file* kerja.[8]
- e. *Android Virtual Device (AVD)*

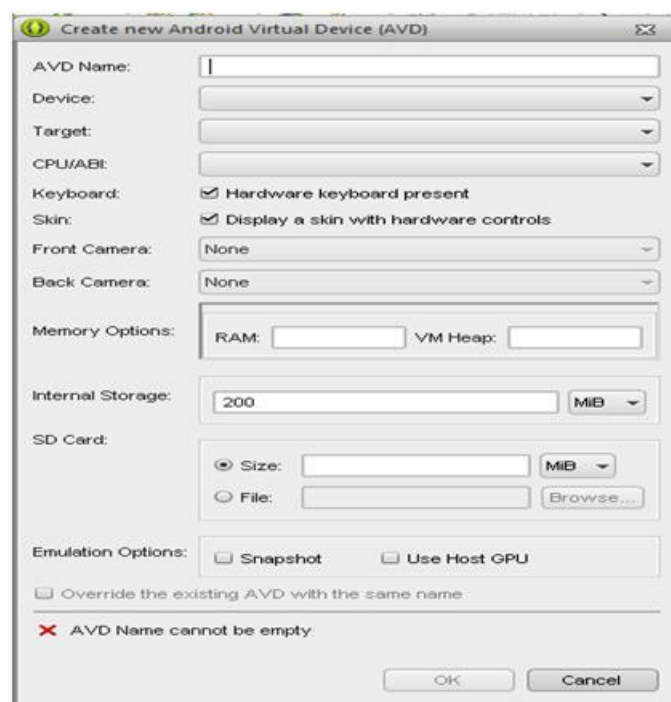
AVD merupakan *emulator* yang digunakan untuk menjalankan program aplikasi yang telah dibuat sehingga pengembang aplikasi dapat mengetahui aplikasi yang telah dibuat berjalan dengan baik atau tidak aplikasi tersebut. Langkah - langkah membuat AVD yaitu sebagai berikut:

- Memilih menu windows pada menu bar Eclipse kemudian memilih *android virtual devices managers* maka muncul tampilan seperti pada gambar 3.8.



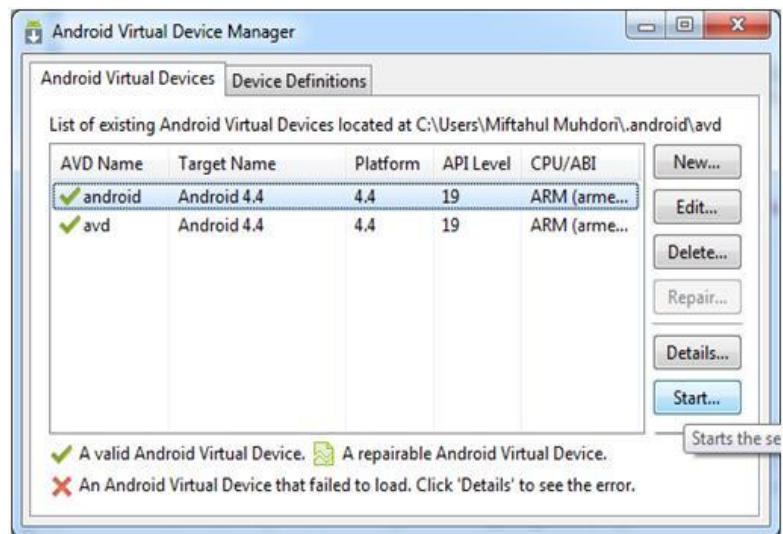
Gambar 3.8 Tampilan *Android Virtual Device Managers*

- Kemudian memilih tombol *new* maka akan tampil seperti pada gambar 3.9



Gambar 3.9 Tampilan Pembuatan AVD

- Pada gambar 3.9 AVD name sebagai label penamaan diisi sesuai dengan dengan versi Android atau nama yang mudah diingat. *Device* diisi dengan ukuran layar yang akan digunakan atau sesuai dengan kebutuhan. Target diisi dengan *platform* Android dimana disesuaikan dengan aplikasi yang dibuat, contohnya API level 19, dan *Hardware* diisi sesuai keperluan. Apabila sudah kemudian memilih Ok.



Gambar 3.10 Pemilihan AVD

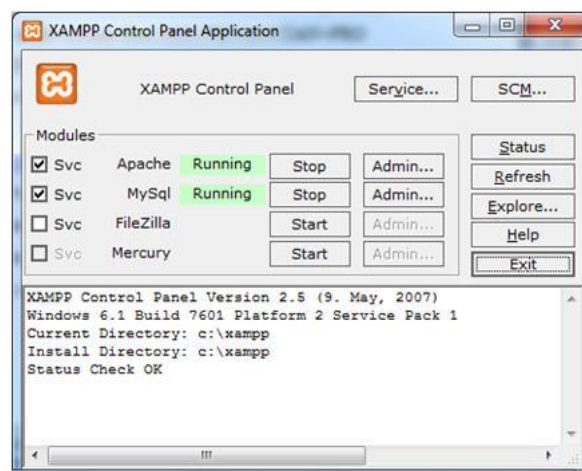
- Pada gambar 3.10 memilih android atau avd kemudian memilih tombol *start* maka akan muncul tampilan *launch option*, pada tampilan tersebut memilih *launch* maka *emulator* memulai proses *booting*, setelah proses *booting* selesai maka akan tampil seperti pada gambar 3.11, *emulator* siap digunakan.

Gambar 3.11 Tampilan Awal *Emulator*

f. XMAPP

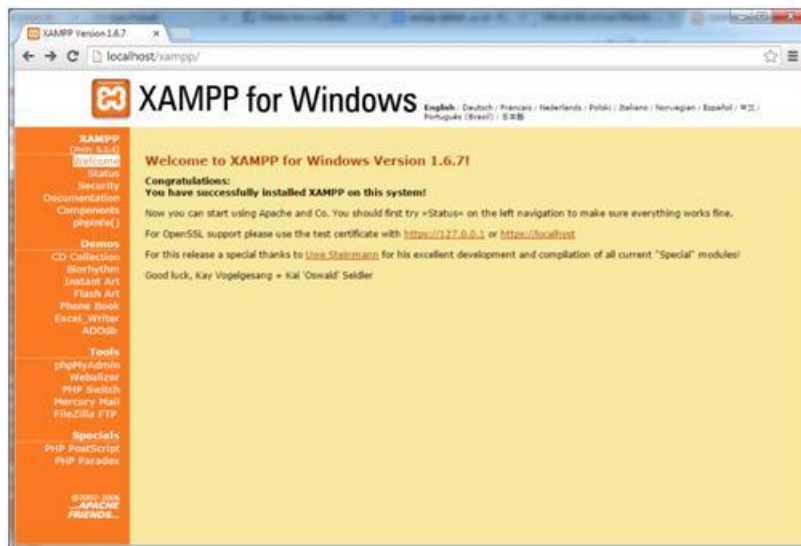
XAMPP merupakan sebuah paket instalasi untuk PHP, APACHE dan MySQL.[14] XAMPP digunakan untuk membuat *server* yang berisi data dari inventaris barang, dalam pembuatannya penulis menggunakan *source code* bahasa SQL, berikut langkah pembuatannya,

Menjalankan Control Panel XMAPP, dengan menu Apache dan MySQL active



Gambar 3.12 XAMPP Control Panel

Software xampp dapat dibuka di *browser*, dengan mengetikan alamat yaitu *localhost*,



Gambar 3.13 Tampilan awal XAMPP

Dalam pembuatan *database server* ini data yang dibuat dishare secara umum, sehingga ada kemungkinan *user* selain admin bisa mengubah *database* ini dengan mengetikan alamat IP dari *database* (192.168.0.103), untuk mengatasi hal tersebut penulis membuat *security* untuk mengamankan *database* ini, sehingga *database* ini hanya bisa di akses oleh admin, dengan cara seperti berikut,

1. mengganti *password* dari *user* root di MySQL XAMPP, dengan cara akses ke `http://localhost/phpmyadmin` dan masuk ke menu Privileges.
2. Di *file* yang bernama `config.inc.php` `(C/xampp/phpMyAdmin.)` Search tulisan `$cfg['Servers'][$i]['auth_type'] = 'config'`; selanjutnya mengganti script config menjadi http atau cookie.
3. Kemudian di bagian `$cfg['Servers'][$i]['user'] = '[isi nama user]'`; penulis mengisikan *user* dengan nama "admin".
4. Untuk bagian `$cfg['Servers'][$i]['password'] = '[isi pwd]'`penulis menggunakan *password* "admin".

Hasil akhir yaitu ketika orang mengakses nomer IP / *host* yang ada *service* xampp misal `http://192.168.0.103/phpmyadmin`. Maka akan langsung ditanya *username* dan *password*.

g. Notepad++

Software ini digunakan untuk membuat *source code* PHP, dimana berfungsi untuk menjembatani antara aplikasi dengan *database server*, penulis membuat lima *scirpt* PHP untuk masing – masing *interface* pengguna, salah satunya yaitu untuk menampilkan *list* nama barang, berikut *source codenya*,

```
<?php
$link = mysql_connect('localhost', 'root', '') or die('Cannot connect
to the DB');
mysql_select_db('inventaris', $link) or die('Cannot select the DB');

$query = "SELECT * FROM barang";
$result = mysql_query($query, $link) or die('Errorquery: '.$query);

$rows = array();
while ($r = mysql_fetch_assoc($result)) {
    $rows[] = $r;
}
$data = "{barang:".json_encode($rows)."}";
echo $data;
?>
```

Source code tersebut merupakan *source code* yang menghubungkan *MainActivity.class* di java dengan *database server* dimana bahasa SQL diterjemahkan menjadi bahasa yang dapat dibaca oleh Java dengan menggunakan *json_encode*. Untuk *source code* “`mysql_select_db('inventaris', $link) or die('Cannot select the DB');`”;`$query = "SELECT * FROM barang";`” berfungsi untuk mengambil data dari tabel yang bernama “barang” dari folder “inventaris” yang berisi *list* barang inventaris. Penulis membuat 4 *file* .php dimana keempat *file* tersebut diletakan di folder `C://xampp/htdocs/data`, sehingga *file.php* tersebut bisa membaca *database* dan menterjemahkannya ke aplikasi di android.

2. *Hardware*

Dalam perancangan aplikasi pengontrolan data inventaris barang selain membutuhkan perangkat lunak (*software*) tetapi juga membutuhkan perangkat keras (*hardware*). Perangkat keras yang dibutuhkan pada perancangan aplikasi ini adalah sebagai berikut:

a. Komputer atau laptop

Adapun minimal spesifikasi komputer atau laptop yang digunakan untuk menginstal dan menjalankan IDE Eclipse ADT adalah sebagai berikut:

1. Windows XP, Windows 7 (32 / 64 bit), Linux
2. *Space harddisk* 315 MB
3. RAM minimal 512 MB sampai dengan 1 GB agar mendapatkan kinerja terbaik

Laptop yang digunakan penulis untuk membuat, menginstal dan menjalankan simulator aplikasi ini spesifikasinya adalah sebagai berikut:

1. Laptop Asus X44H, processor 1,6 Ghz
2. Windows Seven Ultimate 64 bit
3. *Space harddisk* >20GB
4. RAM 2,0 GB

b. Kabel Data

Kabel data yang digunakan merupakan kabel data dari *handphone* Acer Liquid Z200 yang berfungsi untuk mengirimkan aplikasi (.apk) yang sudah siap diinstal dari komputer ke android *mobile*.

c. *Handphone*

Handphone pada perancangan aplikasi ini digunakan sebagai alat untuk menjalankan aplikasi. *Handphone* yang digunakan yaitu *handphone* Acer Liquid Z200 dengan sistem operasi Android versi 4.4 (KitKat).

d. *Access Point*

Access Point yang digunakan dalam aplikasi ini adalah TP-LINK dengan menggunakan jenis jaringan *Wi-Fi* 802.11bgn yang mampu mengirimkan data dengan kecepatan 300Mbps, dengan spesifikasi sebagai berikut:

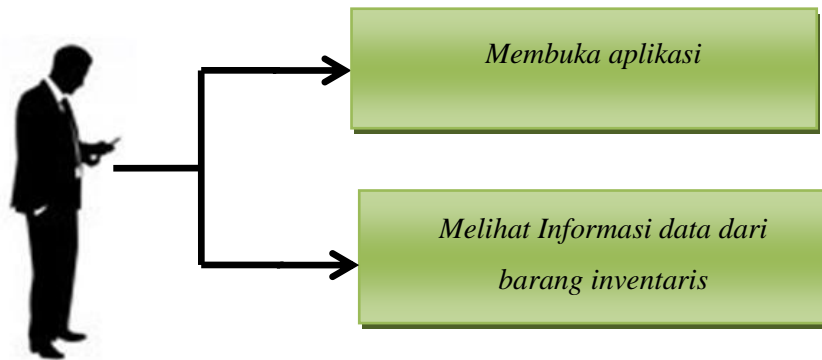
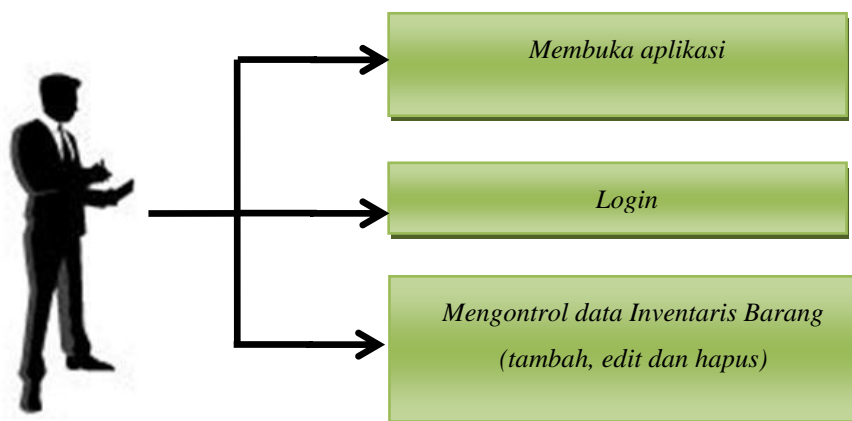
- a. *MAC Address* : C0-4A-00-93-AF-8A
- b. *IP Address* : 192.168.0.254
- c. *Subnet Mask* : 255.255.255.0
- d. *Operation Mode* : *Access Point*
- e. *Wireless Network Name* : TA
- f. *Channel* : *Auto (Current channel 6)*
- g. *Mode* : *11bgn mixed*
- h. *Channel Width* : *Automatic*

3.2 DESAIN APLIKASI

3.2.1 DESAIN UML

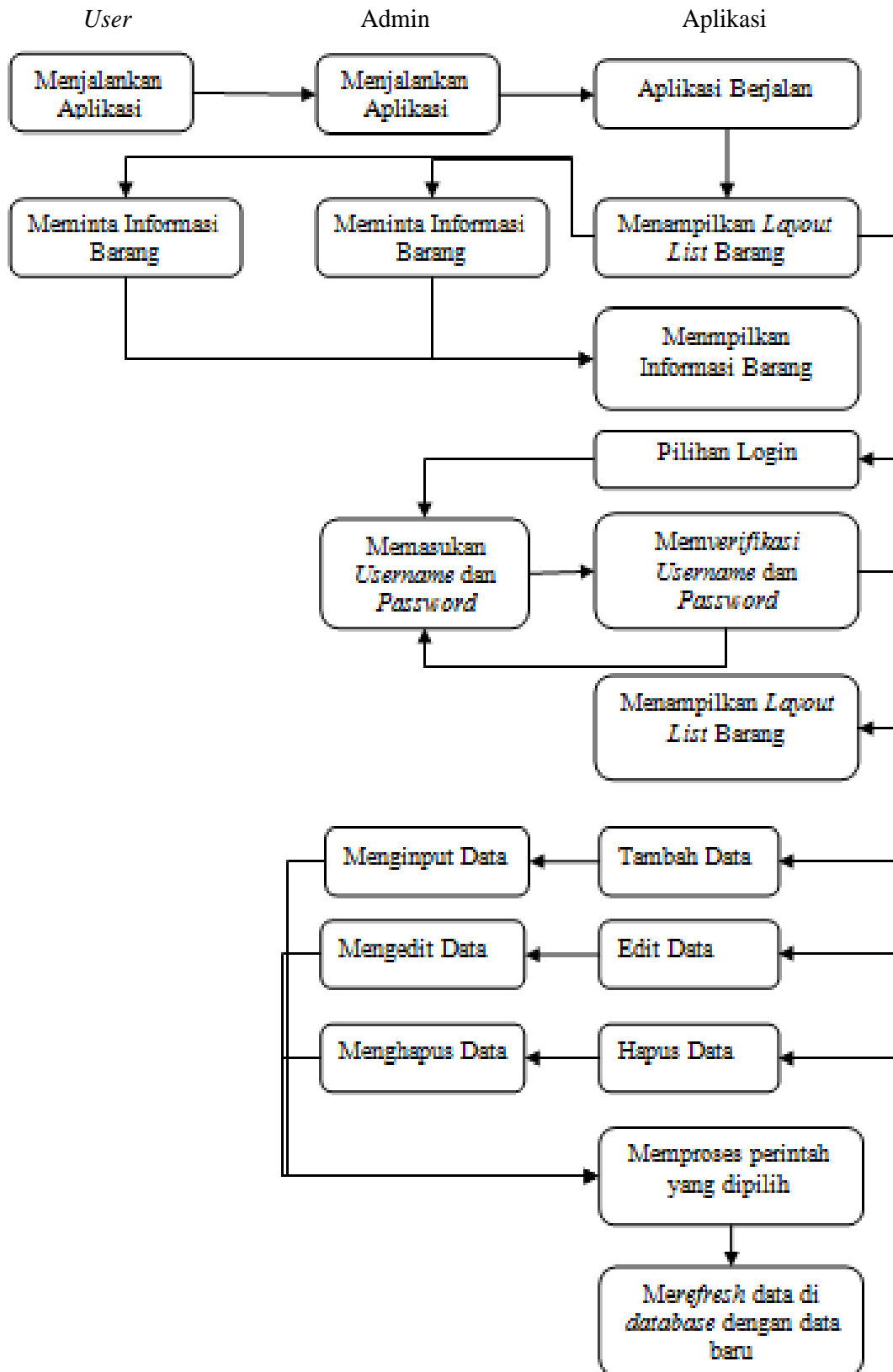
Pada perancangan aplikasi ini penulis membutuhkan desain UML karena pada dasarnya UML merupakan metodologi yang sesuai dengan pemrograman berorientasi objek. Pada perancangan aplikasi ini penulis menggunakan diagram UML antara lain *diagram use case*, *diagram sequence*, dan *diagram activity*. *Diagram use case* digunakan pada perancangan aplikasi ini karena diagram *use case* mendeskripsikan secara lengkap interaksi yang terjadi antara aktor dengan sistem/perangkat lunak yang sedang dikembangkan. *Diagram sequence* adalah sebuah tehnik modeling yang dinamis, sebagai diagram komunikasi UML. *Sequence* banyak digunakan untuk memeriksa dan menyempurnakan logika dan skenario yang berjalan dalam sistem. Penggunaan diagram *sequence* pada perancangan aplikasi ini karena diagram *sequence* menggambarkan rangkaian langkah - langkah yang dilakukan sebagai respon dari sebuah event untuk menghasilkan keluaran tertentu seperti pada gambar 3.14 dan gambar 3.15. *Activity* diagram adalah sebuah penggambaran alur kerja yang dapat digunakan dalam berbagai cara. *Activity* diagram dapat juga digunakan untuk mendeskripsikan tingkat kedetailan alur kerja didalam atau diantara *use case*.

Activity diagram pada perancangan aplikasi ini digunakan karena menggambarkan alur atau proses aktivitas dalam sistem yang sedang dirancang atau di bagian manakah sistem tersebut berjalan[3] seperti pada gambar 3.16.

Gambar 3.14 Diagram Hak Akses *User*

Gambar 3.15 Diagram Hak Akses Admin

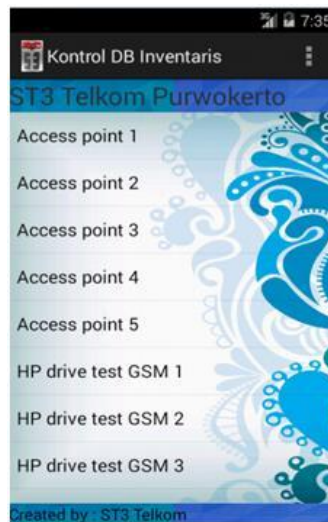
Pada gambar 3.16 terdapat dua pengguna aplikasi yaitu *user* dan admin, masing – masing pengguna mempunyai hak akses yang berbeda, dimana *user* hanya mempunyai hak akses untuk melihat informasi dari barang inventaris di sisi aplikasi, sedangkan untuk admin mempunyai hak akses untuk melihat informasi barang dari sisi aplikasi dan *server*. Dalam penggunaan aplikasi, admin mempunyai hak akses untuk menambah, mengedit dan menghapus data yang terdapat dalam *database server* dengan cara *login* ke *layout* yang berbeda yaitu *list layout* dengan memasukkan *username* dan *password* yaitu “admin”, dimana memasukkan data tersebut akan diproses dan me-*refresh* data yang ada dalam *database server*.



Gambar 3.16 Diagram Activity

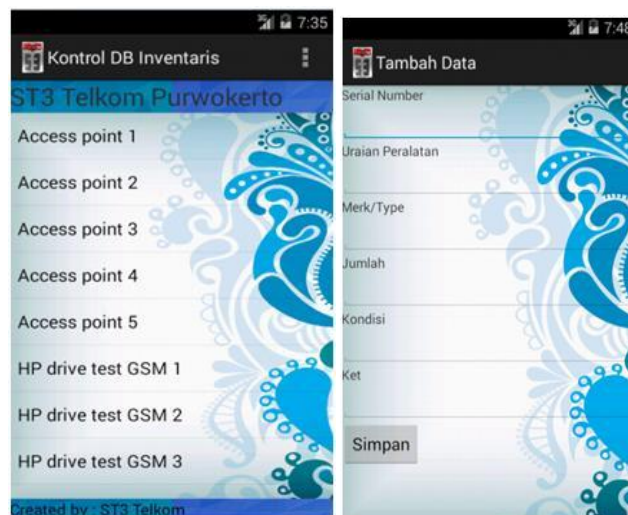
3.2.2 DESAIN USER INTERFACE

Pada desain *user interface* aplikasi ini menampilkan delapan *layout* dimana pada setiap mempunyai fungsinya masing – masing, contohnya pada *layout* pertama yaitu daftar *list* dari nama barang inventaris yang ada di *database server* seperti ditunjukkan pada gambar 3.17.



Gambar 3.17 *User Interface list barang*

Tampilan pada gambar 3.17 hampir sama dengan tampilan pada *layout* ketiga ,namun *layout* ini mempunyai fitur tambahan yaitu menu edit data dan tambah data yang ditunjukkan gambar 3.18(a) dan gambar 3.18(b) .



(a) (b)

Gambar 3.18 *User Interface tambah dan edit data*

Pada gambar 3.18(a) dan gambar 3.18(b) terdiri dari Serial Number, Nama Barang, Merk, Jumlah, Kondisi, dan Keterangan, perbedaannya pada gambar 3.18(a) sudah terdapat data yang dipilih, dimana data tersebut dapat di-*update*, sedangkan pada gambar 3.18(b) hanya terdapat kotak isian tanpa adanya data karena *layout* ini berfungsi untuk menambahkan data ke *database server*.

3.3 PEMBUATAN KODE

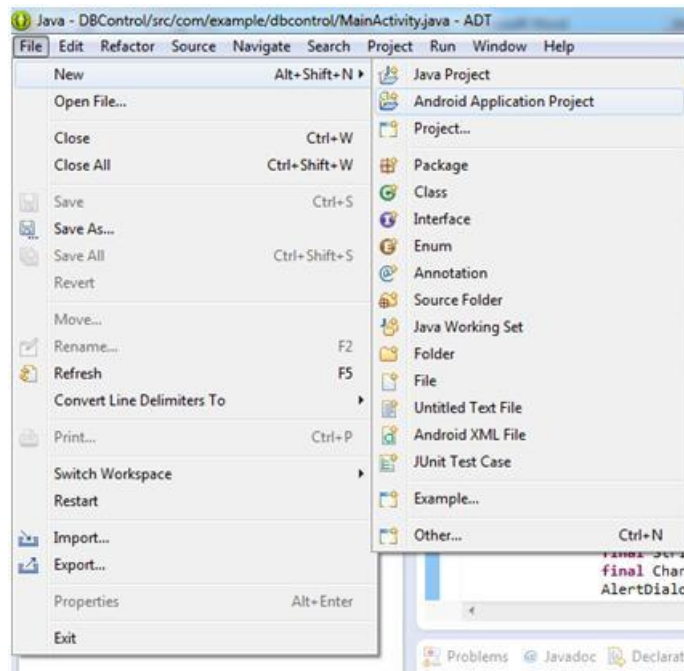
Pembuatan kode atau coding membuat suatu perintah kepada aplikasi yang dibuat dalam bahasa mesin. Pembuatan kode berfungsi sebagai penerjemah desain ke dalam bahasa pemrograman yang dapat diketahui komputer. Penulis menggunakan *software* Eclipse ADT sebagai instrumen. Langkah - langkah dalam pembuatan kode yaitu sebagai berikut:

1. Langkah awal pembuatan aplikasi dengan membuka *software* Eclipse ADT.



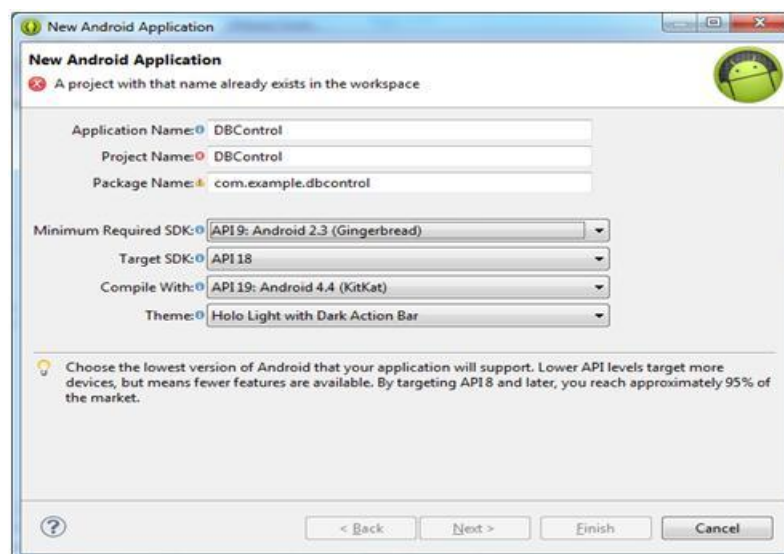
Gambar 3.19 Proses Membuka Eclipse ADT

2. Setelah tampilan awal terbuka, maka untuk pembuatan proyek dengan cara memilih *File > New > Android Application Project*.

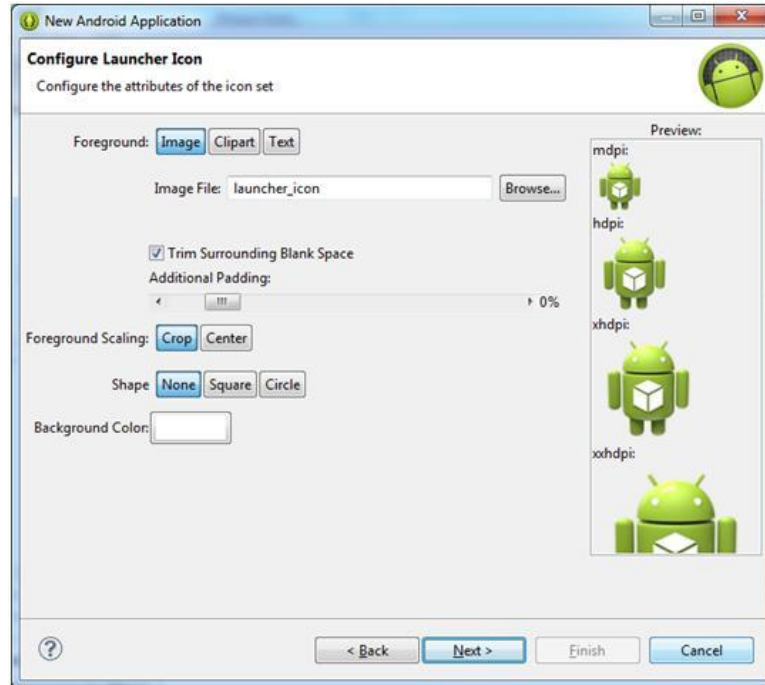


Gambar 3.20 Awal Pembuatan Proyek

3. Ketika memilih *Android Application Project*, maka akan tampil jendela *New Android Application* seperti pada gambar 3.21. Pada kolom *application name* diisi sesuai dengan keinginan masing - masing tetapi alangkah baiknya diisi sesuai dengan aplikasi yang akan dibuat, *project name* dan *package name* akan terisi otomatis pada saat mengisi di kolom *application name*, *minimum required SDK*, *target SDK*, *compile with* sesuai dengan kebutuhan pembuatan proyek.

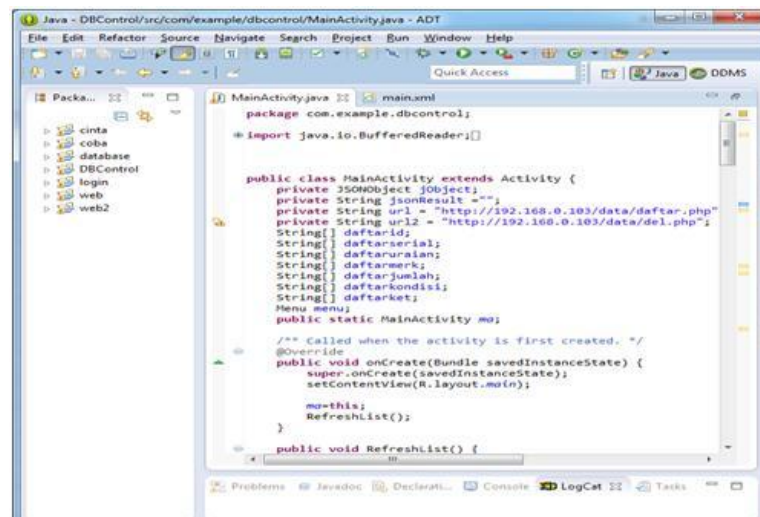
Gambar 3.21 Tampilan *New Android Application*

- Selanjutnya memilih tombol *next* maka akan tampil jendela pemilihan atau pengisian *icon* untuk aplikasi yang akan dibuat. Selanjutnya memilih *next* dan mengisi *activity* yang akan dibuat kemudian memilih *finish*.



Gambar 3.22 Tampilan Pemilihan *Icon* Untuk Aplikasi

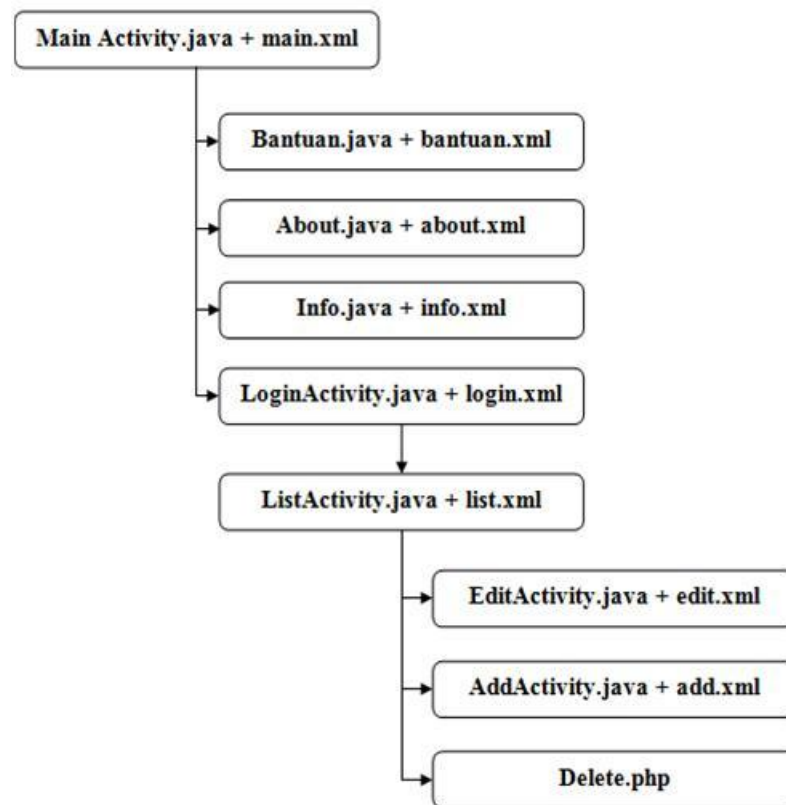
- Tampilan selanjutnya yaitu tampilan *text* editor seperti pada gambar 3.23 yang digunakan untuk penulisan program.



Gambar 3.23 Tampilan *Text* Editor

Kemudian setelah program selesai langkah selanjutnya yaitu running program. Dengan cara memilih tombol run atau meng-klik kanan proyek yang akan di running kemudian memilih *Run as>Android Application*. [3]

Dalam perancangan aplikasi ini penulis mengacu ke block diagram seperti pada gambar 3.24



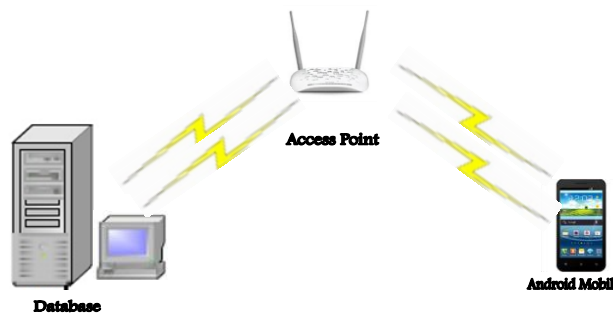
Gambar 3.24 *Flowchart* prinsip kerja masing-masing *class*

Dalam bahasa java terdapat istilah *import* yang merupakan *source code* yang dibutuhkan dalam pembuatan aplikasi untuk memasukan fungsi dari masing – masing *import*, sedangkan untuk Variable yang digunakan, untuk *private* berarti hanya bisa diakses oleh *class* yang bersangkutan, sedangkan untuk “url” dan “url2” menunjukkan bahwa *class* tersebut terhubung ke *localhost database* dan mengambil *script PHP* yang dituju sebagai *interface/ jalan masuk ke database*. Dalam *source code* ini terdapat perintah untuk masuk ke *class* selanjutnya yaitu *info.class* yang terdapat di *source code* “*Intent i = new Intent(getApplicationContext(), Info.class);*” yang artinya jika perintah dijalankan maka akan masuk ke *class info* yang merupakan *class* dari *layout info.xml*.

3.4 PEMBUATAN STRUKTUR JARINGAN *WI-FI*

Aplikasi pengontrolan data inventaris barang akan memanfaatkan jaringan *Wi-Fi* dengan menggunakan *Access Point* (AP), AP digunakan sebagai *hardware* penyedia jaringan *Wi-Fi* untuk media transmisi pengiriman data dari *PC server* ke *Android mobile* atau sebaliknya, untuk lebih jelasnya perhatikan gambar 3.25 Aplikasi ini memanfaatkan jaringan *Wi-Fi* dimana Jenis jaringan yang digunakan adalah *WLAN (Wireless LAN)* disini penulis menggunakan jaringan yang terpisah dari jaringan kampus, dikarenakan jaringan kampus menggunakan jenis *DHCP (Dynamic Host Configuration Protocol)* yang bertolak belakang dengan aplikasi, dimana aplikasi ini menggunakan IP yang sudah ditentukan atau *Static*, dan juga untuk menghindari adanya tabrakan data jika menggunakan jaringan kampus yang dikarenakan kemungkinan terjadinya alamat IP yang sama antara aplikasi yang menggunakan IP *static* dengan Jaringan kampus yang bersifat otomatis memberikan IP ke setiap *host* yang memungkinkan alamat IP sama dengan aplikasi.

Untuk jaringan *WLAN* yang dibuat penulis menggunakan satu buah *Access Point* sebagai perantara pengiriman data dari Aplikasi ke *server* atau sebaliknya seperti pada gambar 3.25 .



Gambar 3.25 Struktur jaringan aplikasi