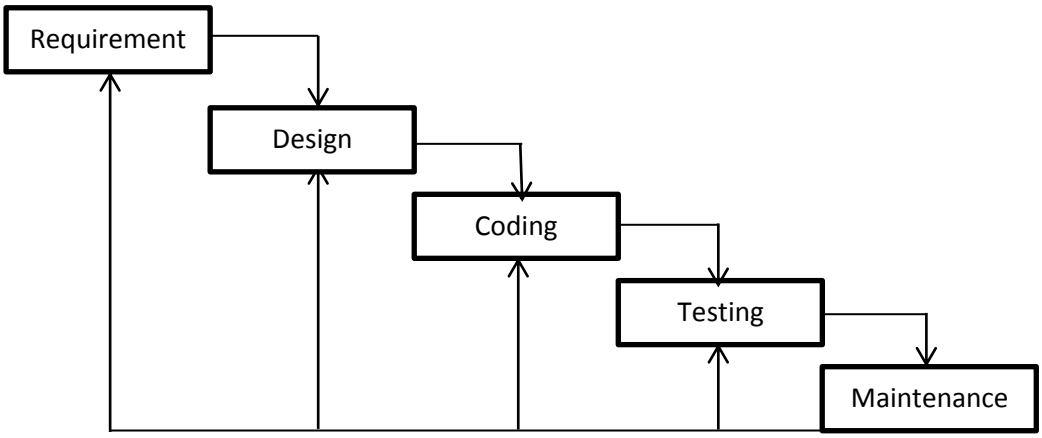
BAB III
METODE PENELITIAN

3.1 Metode Pengembangan Aplikasi

Metode *waterfall* merupakan metode yang sering digunakan oleh penganalisa sistem pada umumnya. Inti dari metode *waterfall* adalah pengerjaan dari suatu sistem dilakukan secara berurutan atau secara linear. Jadi jika langkah ke-1 belum dikerjakan, maka langkah 2 tidak dapat dikerjakan. Jika langkah ke-2 belum dikerjakan maka langkah ke-3 juga tidak dapat dikerjakan, begitu seterusnya. Secara otomatis langkah ke-3 akan bisa dilakukan jika langkah ke-1 dan ke-2 sudah dilakukan. Secara garis besar metode *waterfall* mempunyai langkah-langkah sebagai berikut : Analisa, Desain, Penulisan, Pengujian dan Penerapan serta Pemeliharaan. (Kadir, 2003). Metode air terjun atau yang dapat disebut sebagai metode *waterfall*, dimana hal ini digambarkan dengan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan, permodelan, konstruksi, serta penyerahan sistem ke para konsumen, yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan (Pressman, 2012). Berikut metode *waterfall* ditunjukkan pada gambar 3.1.



Gambar 3.1 Metode *Waterfall*

3.2 Requirement

3.2.1 Studi Literatur

Mempelajari teori dan referensi yang berhubungan dengan manajemen berbasis *android* dan *web* antara lain prinsip dan prosedur DBMS (system managemen basis data), pemodelan data yang meliputi *Flowchart*, pemrograman dengan *PHP* dan database *MySQL*.

3.2.2 Observasi

Pada metode pengamatan (observasi), dilakukan peninjauan dan penelitian langsung di lapangan untuk memperoleh dan mengumpulkan data yang dibutuhkan.

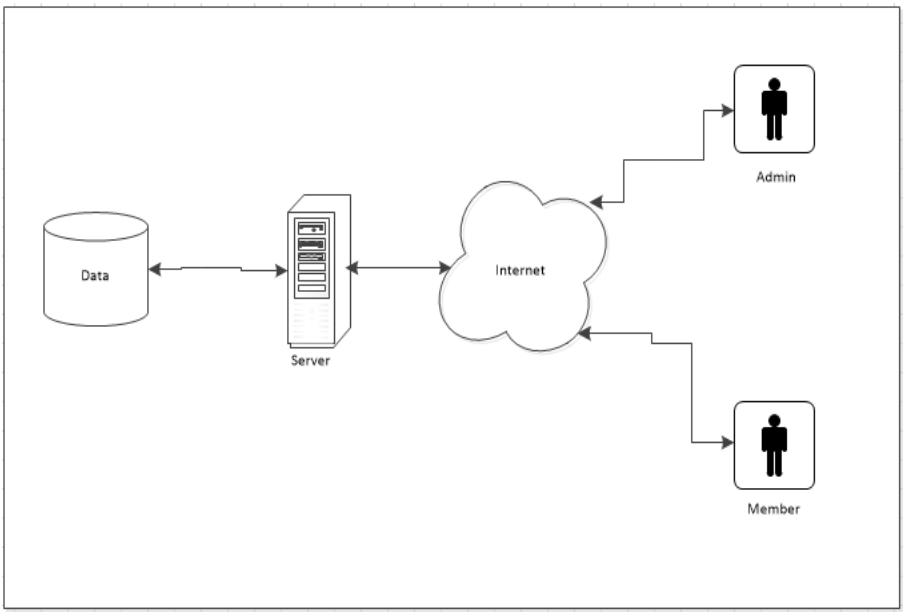
3.2.3 Analisis

Pada metode ini dari hasil perolehan dan pengumpulan data yang dilakukan kemudian akan dianalisa agar dapat di rancang sistem.

3.3 Design

4.3.6 Perancangan Sistem

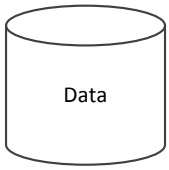
Pada tahap ini dilakukan penentuan perancangan sistem yang akan digunakan seperti *use case* diagram, *activity* diagram, *class* diagram, arsitektur sistem, ERD, dan rancangan antar muka. Berikut ini merupakan gambar perancangan sistem yang diusulkan pada gambar 3.2:



Gambar 3.2 Arsitektur Sistem yang Diusulkan

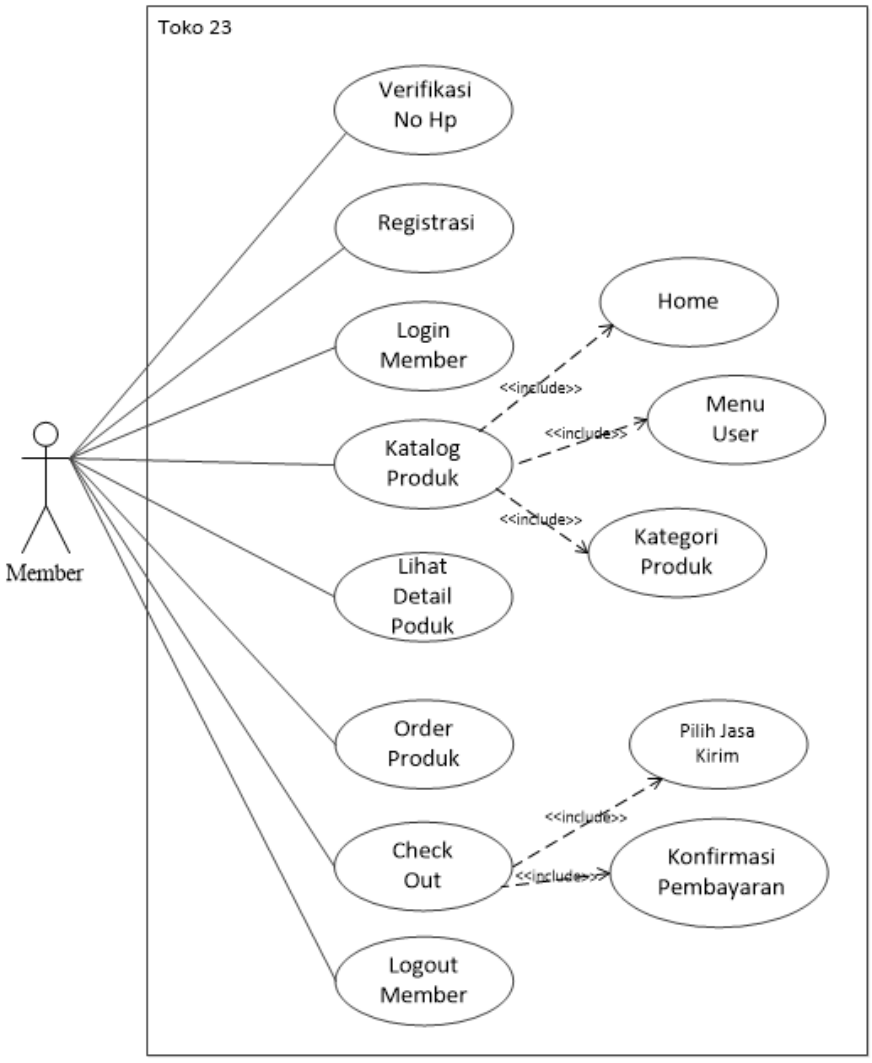
Database server yang digunakan menggunakan pada aplikasi *MySql* dan menggunakan *xampp* sebagai *web server*. Komunikasi antara pengguna dan *web server* mengunakan *internet* dan *web browser* pada perangkat pengguna. Saat pengguna mengakses aplikasi, *web server* memuat antar muka dan melakukan pengambilan data yang dilakukan dari *database server*. Melalui antar muka yang dimuat *web server* sebagai pengguna bisa menyimpan *database server* dapat lihat pada tabel 3.5.

Tabel 3.1 Keterangan Komponen Arsitektur

	Admin *Member* *Non Member*
	Berisi data-data sebagai berikut: 1. Data barang. 2. Kategori barang. 3. *List* barang . 4. Data pesanan . *5. Slide benner.* 6. Data *member*. 7. Laporan perbarang. 8. Laporan pertransaksi.
	*Server* digunakan untuk penyimpanan data-data tersebut.

4.3.7 Use Case Diagram

Use Case Diagram menurut (Widodo, 2011:10) *diagram use case* bersifat *statis* yang memperlihatkan himpunan *Use Case* dan aktor-aktor (suatu jenis khusu dari kelas) dan menggambarkan apa saja aktifitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. *Use Case* Diagram digunakan untuk memodelkan proses bisnis berdasarkan pandangan pengguna sistem. *Use case* diagram lebih menekankan pada "siapa" melakukan "apa" dalam lingkungan sistem perangkat lunak yang akan dibangun dapat dilihat pada gambar 3.3.



Gambar 3.3 *Use Case* Diagram Aplikasi Penjualan Online Perkakas Rumah Tangga

Berikut ini adalah penjelasan *use case* yang dibuat:

Aktor: Member

***Case*:**

a. *Verifikasi No Hp:* memungkinkan pengguna menginput no hp dan *memverifikasikan* no hp tersebut agar dapat terdaftar menjadi *member.*

b. *Registrasi:* memungkinkan pengguna untuk registrasi terlebih dahulu.

c. *Login:* memungkinkan pengguna untuk masuk ke aplikasi dengan *login* setelah selesai *registrasi*.

d. *Katalog Produk:* memungkinkan pengguna untuk melihat aja saja yang berada di *katalog produk*.

e. *Home:* memungkinkan pengguna untuk mengetahui produk apa saja yang dijualkan pada toko tersebut.

f. *Menu User:* memungkinkan pengguna mengetahui fitur apa saja yg ada dalam aplikasi tersebut.

g. *Kategori Produk:* memungkinkan pengguna mengetahui fitur produk apa saja yang dijual.

h. *Lihat Detail Produk:* memungkinkan pengguna untuk melihat *detail* produk yang ditawarkan.

i. *Order*: memungkinkan pengguna untuk memesan produk yang ditawarkan.

j. *Check Out*: memungkinkan pengguna untuk melihat produk apa saja yang sudah di order member dan dapat mengkonfirmasikan pembayaran serta dapat meng*input* alamat agar barang dapat sampai ditujuan serta.

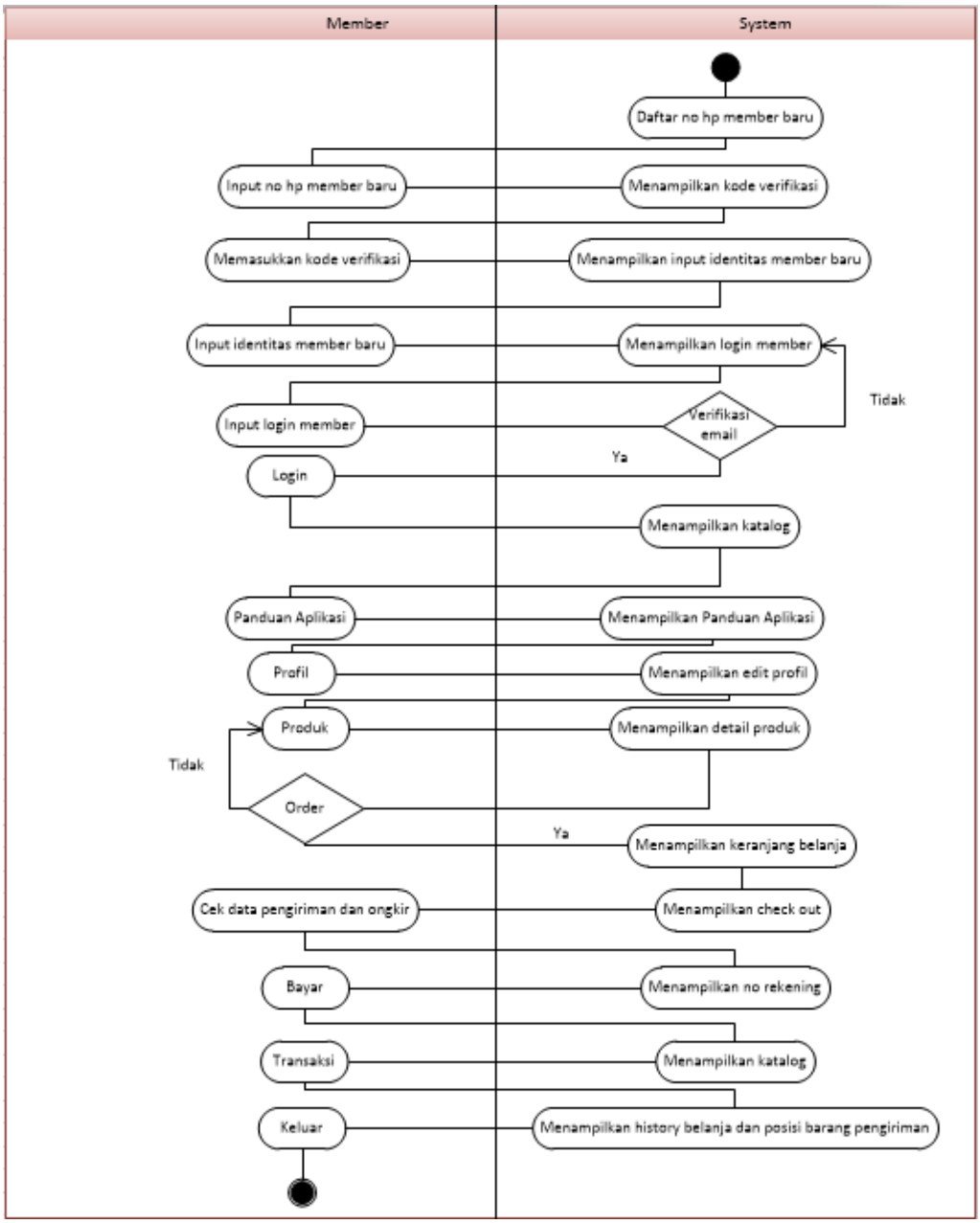
k. *Pilih Jasa:* memungkinkan pengguna untuk memilih jasa kurir COD atau JNE sesuai kebutuhan.

l. *Konfirmasi Pembayaran:* memungkinkan pengguna untuk mengkonfirmasikan pembayaran produk yang akan dikirim.

m. *Logout:* memungkinkan pengguna untuk keluar dari aplikasi.

4.3.8 Activity Diagram

Activity Diagram adalah berupa gambaran alur dari bagaimana suatu sistem mengawali, melakukan, dan mengakhiri proses tersebut bekerja. Berikut ini akan dijelaskan beberapa *activity* diagram yang akan diterapkan pada aplikasi Penjualan *Online* Perkakas Rumah Tangga Berbasis Android dapat dilihat pada gambar 3,4.

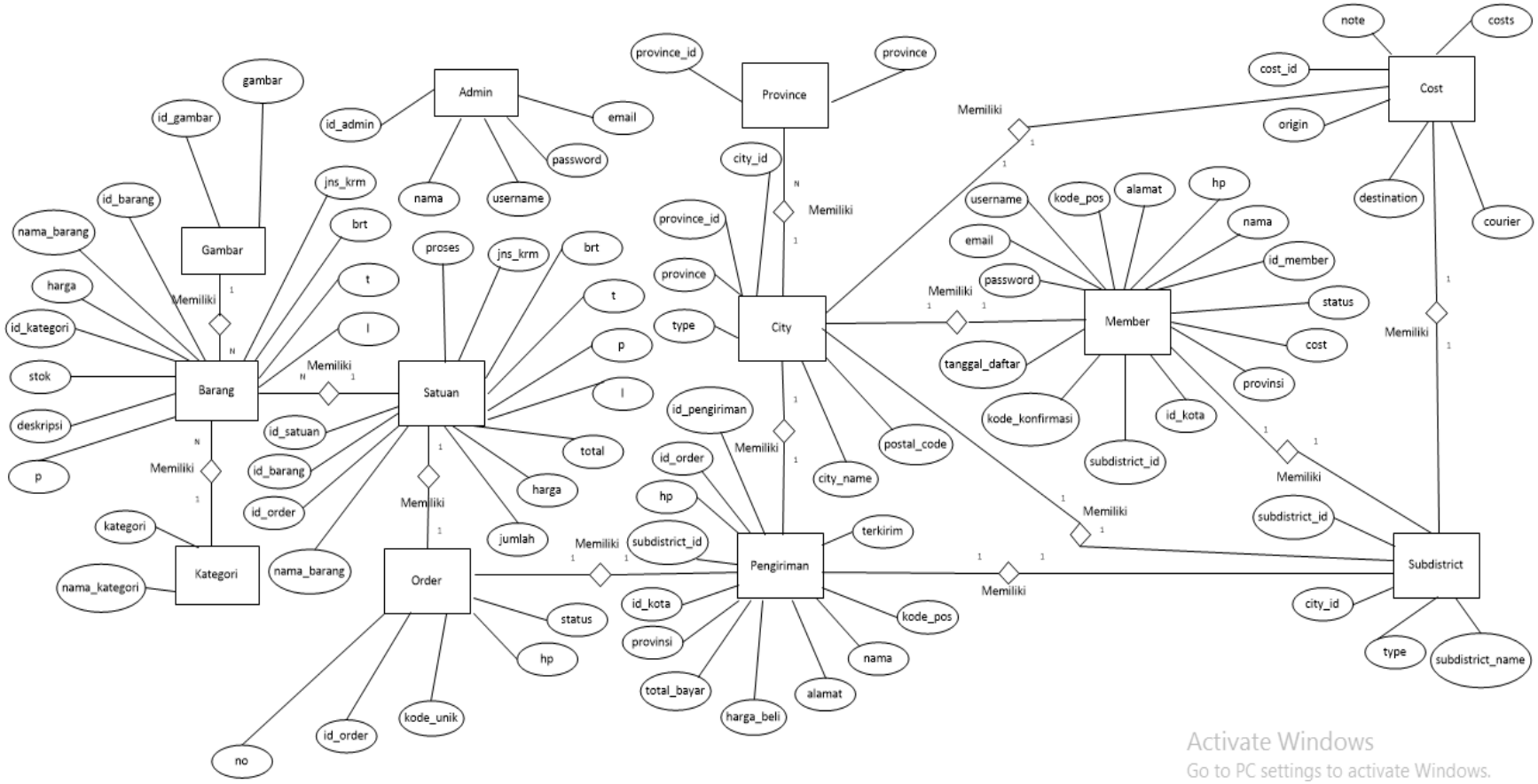


Gambar 3.4 *Activity* Diagram *Form* Penjualan Online Berbasis Android

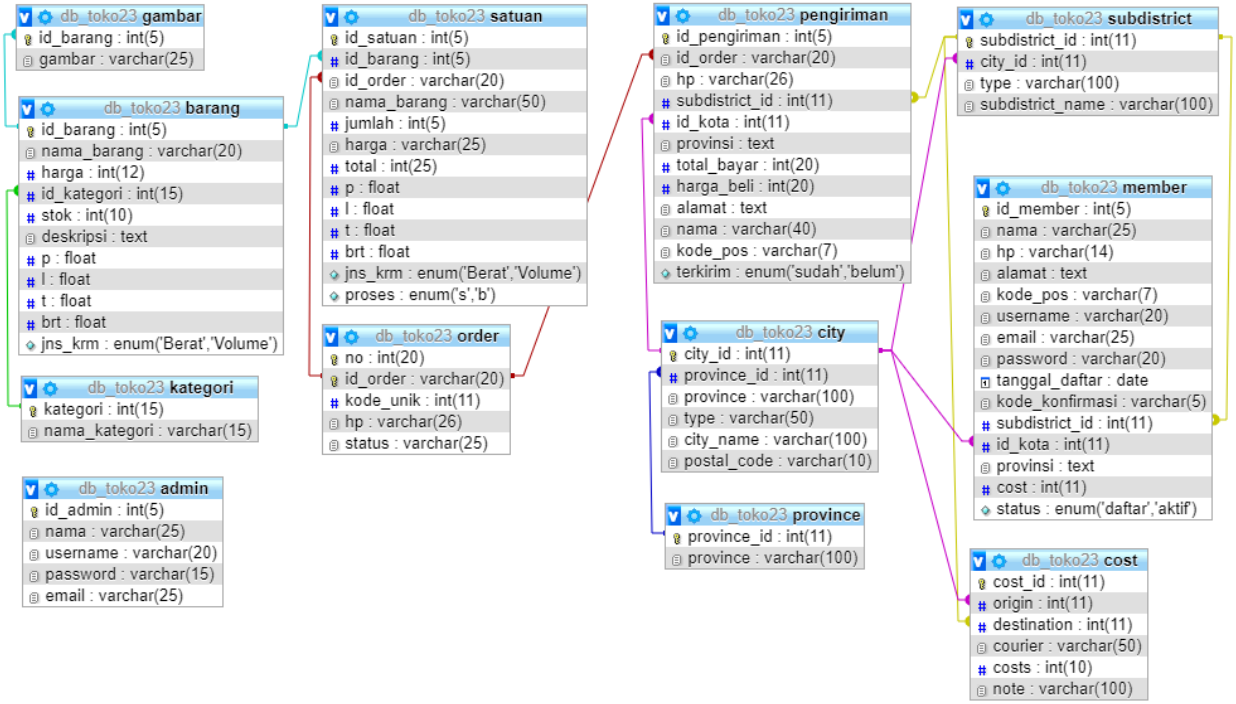
Pada Gambar 3.4 Activity Diagram diatas menjelaskan tentang bagaimana cara kerja dari sistem aplikasi secara detail, ini mulai dari membuka aplikasi terlebih dahulu, kemudian menampilkan logo dalam bentuk *splash screen* yang diikuti dengan halaman utama yang meminta menginput no telphone untuk kode verifikasi sebelum registrasi *member*, lalu *member* mendapatkan konfirmasi yang kemudian *member* melakukan registrasi agar dapat *login*. Setelah dapat *login* kemudian dapat langsung melihat penawaran atau katalog produk yang disediakan, jika ada produk yang diminati maka pemilihan produk tersebut dapat menampilkan detail produk yang akan di order. Aplikasi akan menampilkan detail order yang sudah selesai dipilih lalu selanjutnya pengguna akan check - out untuk menginput form alamat yang akan dikirim, kemudian pengguna juga diminta untuk menginput form konfirmasi pemesanan dan tagihan pembayaran.

4.3.9 Entity Relationship Diagram (ERD)

Gambar berikut ini menjelaskan hubungan relasi antar data dalam basis data yang terdapat di dalam sistem PENJUALAN ONLINE PERKAKAS RUMAH TANGGA:



Gambar 3.5 Entity Relationship Diagram (ERD) bentuk pertama



Gambar 3.6 Entity Relationship Diagram (ERD) bentuk ke dua

Berikut penjelasan tentang gambar 3.5 & 3.6:

1. Pada entitas gambar memiliki relasi dengan entitas barang yaitu *one to many* dengan atribut id_barang, artinya satu gambar dapat memiliki banyak barang yang tersedia di aplikasi android.
2. Pada entitas barang memiliki relasi dengan entitas kategori yaitu *many to one* dengan atribut id_kategori, artinya banyaknya barang memiliki satu kategori yang terdapat dalam aplikasi.
3. Pada entitas barang memiliki relasi dengan entitas satuan yaitu *many to one* dengan atribut id_barang, artinya banyaknya barang memiliki satu satuan.
4. Pada entitas satuan memiliki relasi dengan entitas *order* yaitu *one to one* dengan atribut id_order, artinya satu satuan memiliki satu pesanan barang.
5. Pada entitas *order* memiliki relasi dengan entitas pengiriman yaitu *one to one* dengan atribut id_order, artinya pemesanan yang dipesan cukup satu pengiriman.
6. Pada entitas pengiriman memiliki relasi dengan entitas *city* yaitu *one to one* dengan atribut city_id, artinya dalam satu pengiriman memiliki satu kota yang dituju.
7. Pada entitas pengiriman memiliki relasi dengan entitas *subdistrict* yaitu *one to one* dengan atribut subdistrict_id, artinya dalam satu pengiriman dimiliki oleh satu kecamatan yang dituju.
8. Pada entitas *city* memiliki relasi dengan entitas *province* yaitu *one to one* dengan atribut province_id, artinya dalam satu kota dimiliki oleh satu provinsi.
9. Pada entitas *city* memiliki relasi dengan entitas *member* yaitu *one to one* dengan atribut city_id, artinya dalam satu kota dimiliki oleh satu *member*..
10. Pada entitas *city* memiliki relasi dengan entitas *subdistrict* yaitu *one to one* dengan atribut city_id, artinya dalam satu kota dimiliki oleh satu kecamatan.
11. Pada entitas *subdistrict* memiliki relasi dengan entitas *member* yaitu *one to one* dengan atribut subdistrict_id, artinya dalam satu kecamatan dimiliki oleh satu *member*.

12. Pada entitas *city* memiliki relasi dengan entitas *cost* yaitu *one to one* dengan atribut city_id & atribut origin, artinya dalam satu kota yang dituju memiliki satu biaya.
13. Pada entitas *subdistrict* memiliki relasi dengan entitas *cost* yaitu *one to one* dengan atribut subdistrict_id & atribut destination, artinya dalam satu kecamatan yang dituju memiliki satu biaya.

Tabel 3.2 Struktur Tabel *Admin*

No	***Name***	***Type***	***Null***	***Extra***
1.	id_admin	Int(5)	NN	*Auto_Increment*
2.	Nama	Varchar(25)	NN	
3.	*Username*	Varchar(20)	NN	
4.	*Password*	Varchar(15)	NN	
5.	*Email*	Varchar(25)	NN	

Tabel 3.3 Struktur Tabel Barang

No	***Name***	***Type***	***Null***	***Extra***
1.	id_barang	Int(5)	NN	Auto_Increment
2.	nama_barang	Varchar(20)	NN	
3.	Harga	Varchar(10)	NN	
4.	id_kategori	Int(15)	NN	
5.	Stock	Int(10)	NN	
6.	Diskripsi	Text	NN	
7.	P	Float	NN	
8.	L	Float	NN	
9.	T	Float	NN	
10.	Brt	Float	NN	
11.	jns_krm	Enum(‘berat, ‘volume’)	NN	

Tabel 3.4 Struktur Tabel City

No	Name	Type	Null	Extra
1.	city_id	Int(11)	NN	Auto_increment
2.	province_id	int(11)	NN	
3.	Province	Varchar(100)	NN	
4.	Type	Varchar(50)	NN	
5.	city_name	Varchar(100)	NN	
6.	postal_code	Varchar(10)	NN	

Tabel 3.5 Struktur Tabel Cost

No	Name	Type	Null	Extra
1.	cost_id	Int(11)	NN	Auto_increment
2.	Origin	int(11)	NN	
3.	Destination	int(11)	NN	
4.	Courier	Varchar(50)	NN	
5.	Costs	int(10)	NN	
6.	Note	Varchar(100)	NN	

Tabel 3.6 Struktur Tabel Gambar

No	Name	Type	Null	Extra
1.	id_gambar	Int(5)	NN	Auto_increment
2.	Gambar	Varchar(25)	NN	

Tabel 3.7 Struktur Tabel Kategori

No	*Name*	*Type*	*Null*	*Extra*
1.	Kategori	Int(5)	NN	Auto_Increment
2.	nama_kategori	Varchar(15)	NN	

Tabel 3.8 Struktur Tabel Konfirmasi

No	***Name***	***Type***	***Null***	***Extra***
1.	id__konfirmasi	Int(5)	NN	Auto_Increment
2.	tgl__konfirmasi	Date	NN	
3.	id_order	Varchar(10)	NN	

Tabel 3.9 Struktur Tabel *Member*

No	*Name*	*Type*	*Null*	*Extra*
1.	id_member	Int(5)	NN	Auto_Increment
2.	Nama	Varchar(25)	NN	
3.	Hp	Varchar(14)	NN	
4.	Alamat	*Text*	NN	
5.	kode_pos	Varchar(7)	NN	
6.	*Username*	Varchar(20)	NN	
7.	*Email*	Varchar(25)	NN	
8.	*Password*	Varchar(20)	NN	
9.	tanggal_daftar	Date	NN	
10.	kode_konfirmasi	Varchar(5)	NN	
11.	subdistrict_id	Int(11)	NN	
12.	id_kota	Int(5)	NN	
13.	Provinsi	Text	NN	
14.	Cost	Int(10)	NN	
15.	Status	enum(‘daftar,’aktif’)	NN	

Tabel 3.10 Struktur Tabel *Order*

No	***Name***	***Type***	***Null***	***Extra***
1.	No	Int(5)	NN	Auto_Increment
2.	id_order	Varchar(20)	NN	
3.	kode_unik	Int(11)	NN	
4.	Hp	Varchar(26)	NN	
5.	Status	Int(25)	NN	

Tabel 3.11 Struktur Tabel Pengiriman

No	**Name**	**Type**	**Null**	**Extra**
1.	id_pengiriman	Varchar(5)	NN	Auto_Increment
2.	id_order	Varchar(20)	NN	
3.	Hp	Varchar(26)	NN	
4.	subdistrict_id	Int(11)	NN	
5.	id_kota	Int(5)	NN	
6.	Provinsi	Text	NN	
7.	total_bayar	Int(20)	NN	
8.	harga_beli	Int(20)	NN	
9.	Alamat	Text	NN	
10.	Nama	Varchar(40)	NN	
11.	kode_pos	Varchar(7)	NN	
12.	Terkirim	Enum(‘sudah,’belum’)	NN	

Tabel 3.12 Struktur Tabel Province

No	**Name**	**Type**	**Null**	**Extra**
1.	province_id	Int(11)	NN	Auto_increment
2.	Province	varchar(100)	NN	

Tabel 3.13 Struktur Tabel Satuan

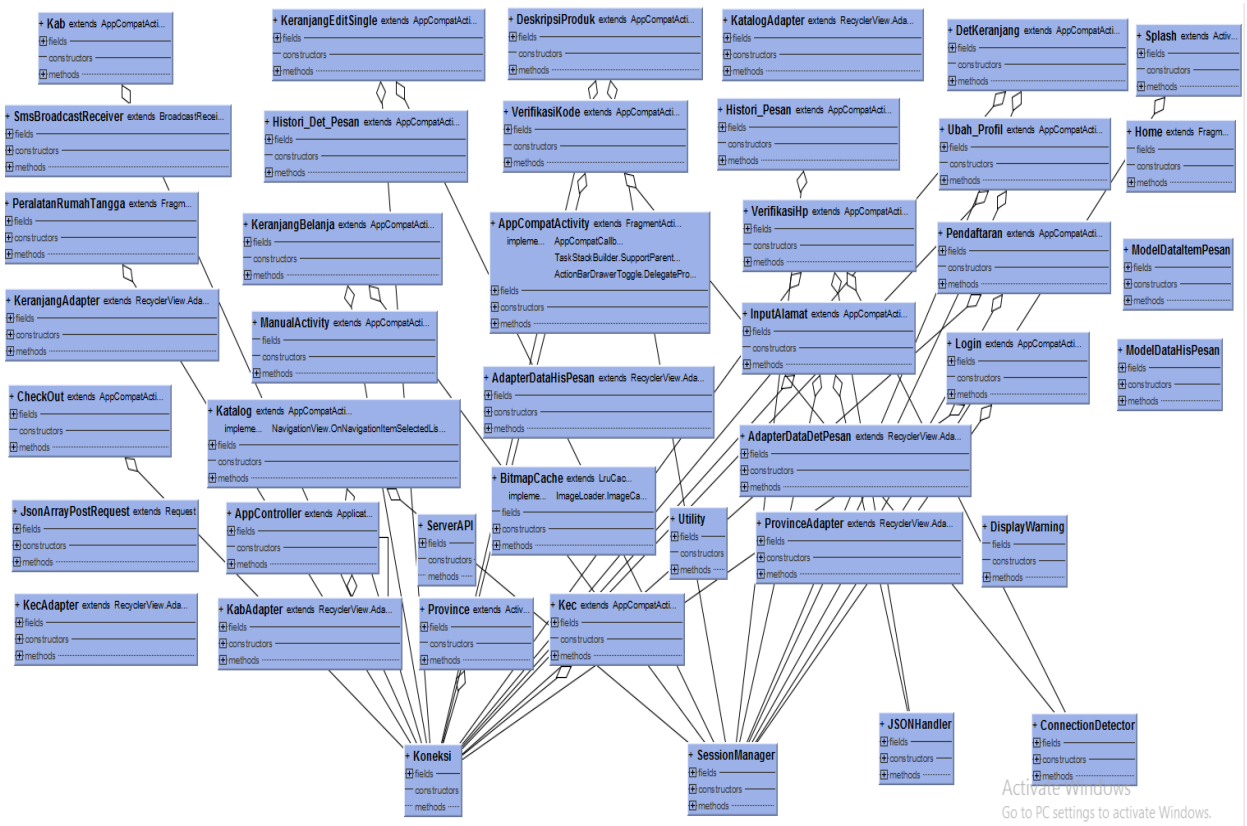
No	*Name*	*Type*	*Null*	*Extra*
1.	id_satuan	Int(5)	NN	Auto_Increment
2.	id_barang	Int(5)	NN	
3.	id_order	Varchar(20)	NN	
4.	nama_barang	Varchar(50)	NN	
5.	Jumlah	Int(5)	NN	
6.	Harga	Varchar(25)	NN	
7.	Total	Int(25)	NN	
8.	P	Float	NN	
9.	L	Float	NN	
10.	T	Float	NN	
11.	Brt	Float	NN	
12.	jns_krm	Enum('berat','volume')	NN	
13.	Proses	Enum('s','b')	NN	

Tabel 3.14 Struktur Tabel Subdistrict

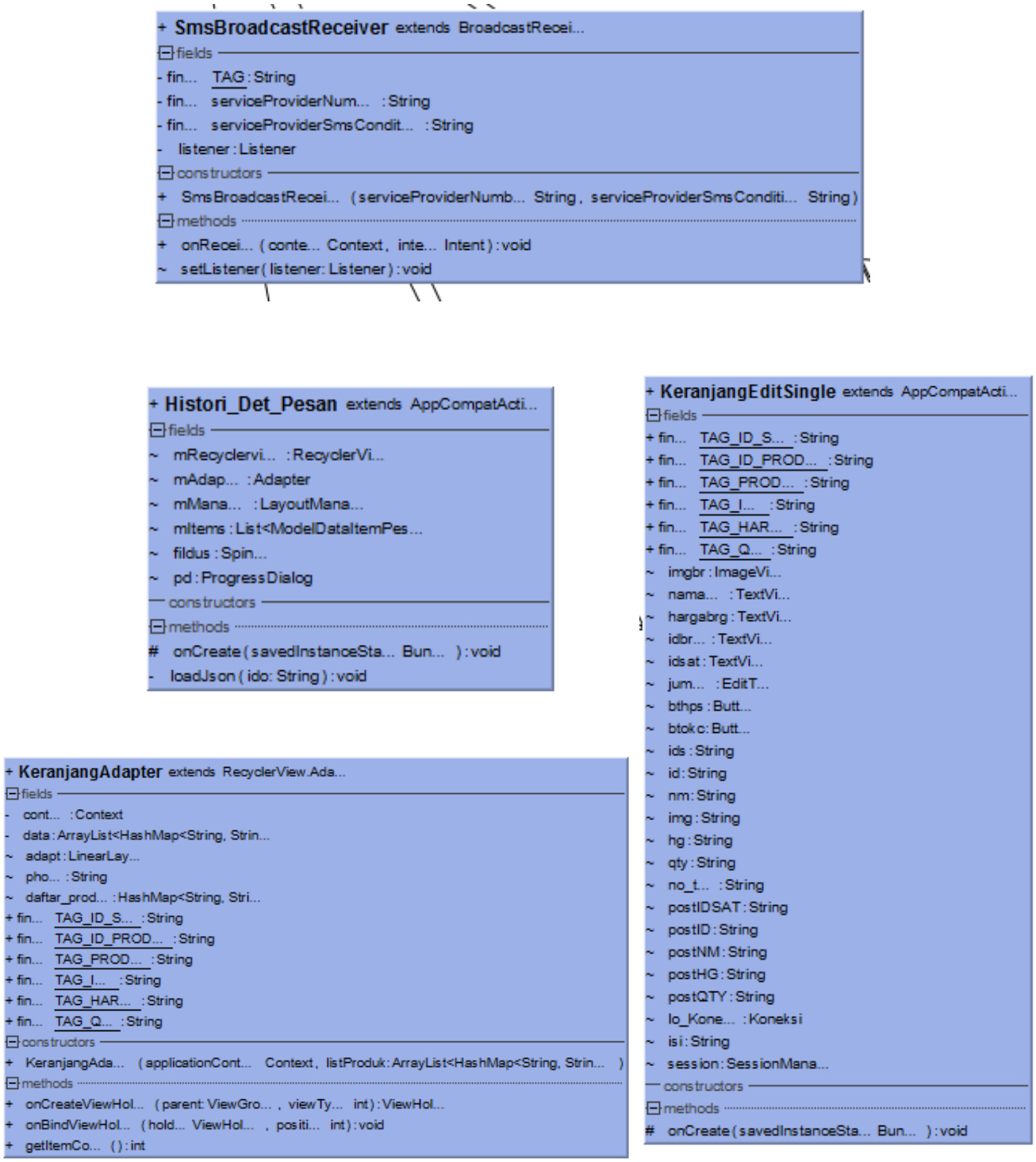
No	Name	Type	Null	Extra
1.	subdistrict_id	Int(11)	NN	Auto_increment
2.	city_id	Int(11)	NN	
3.	Type	Varchar(100)	NN	
4.	subdistrict_name	Varchar(100)	NN	

4.3.10 Class Diagram (CD)

Gambar berikut ini menampilkan class diagram dengan menggunakan simpleUML Diagram :

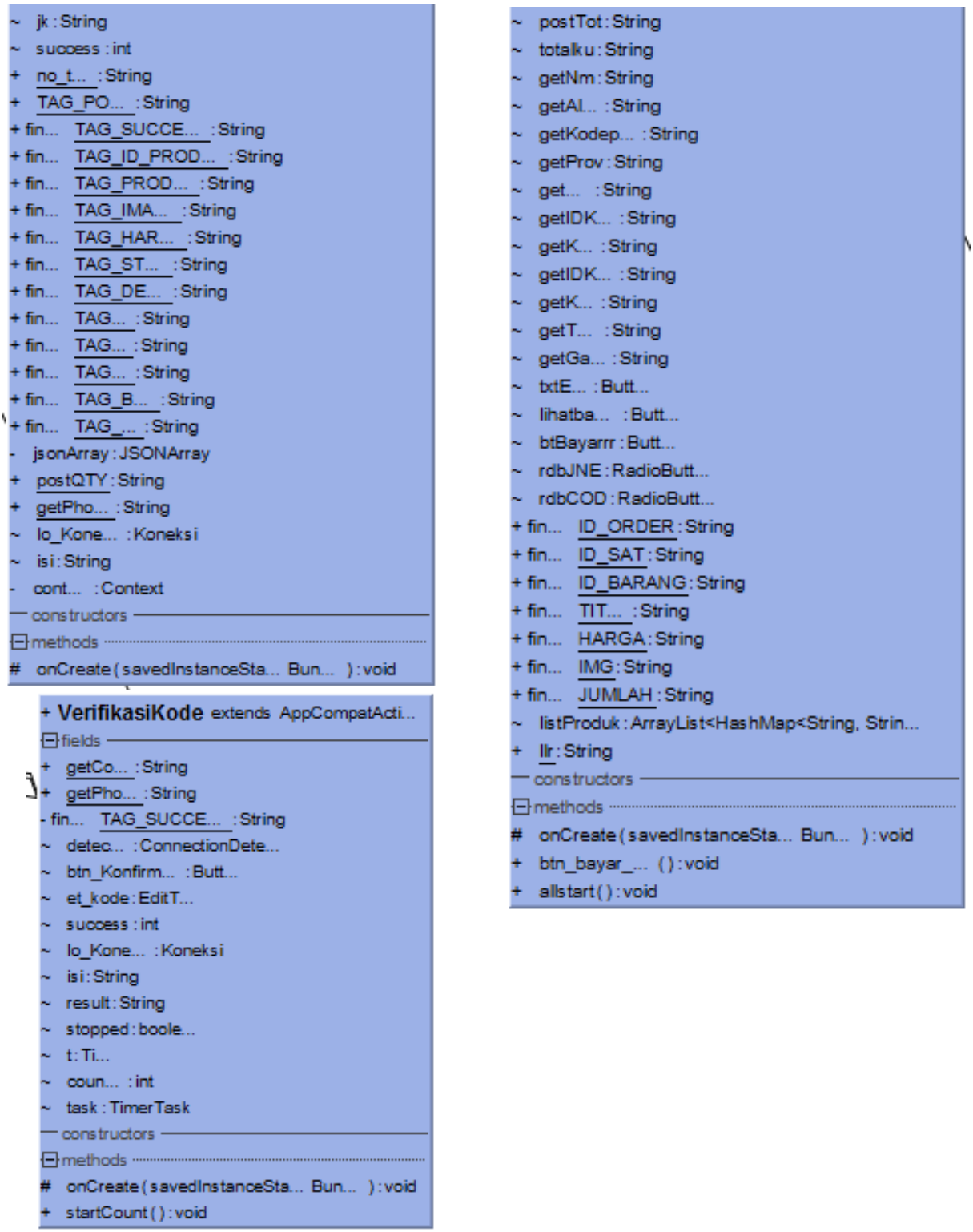


Gambar 3.7 Class Diagram (CD)



Gambar 3.8 Detail Class Diagram (CD)

Gambar 3.9 Detail Class Diagram (CD) Lanjutan



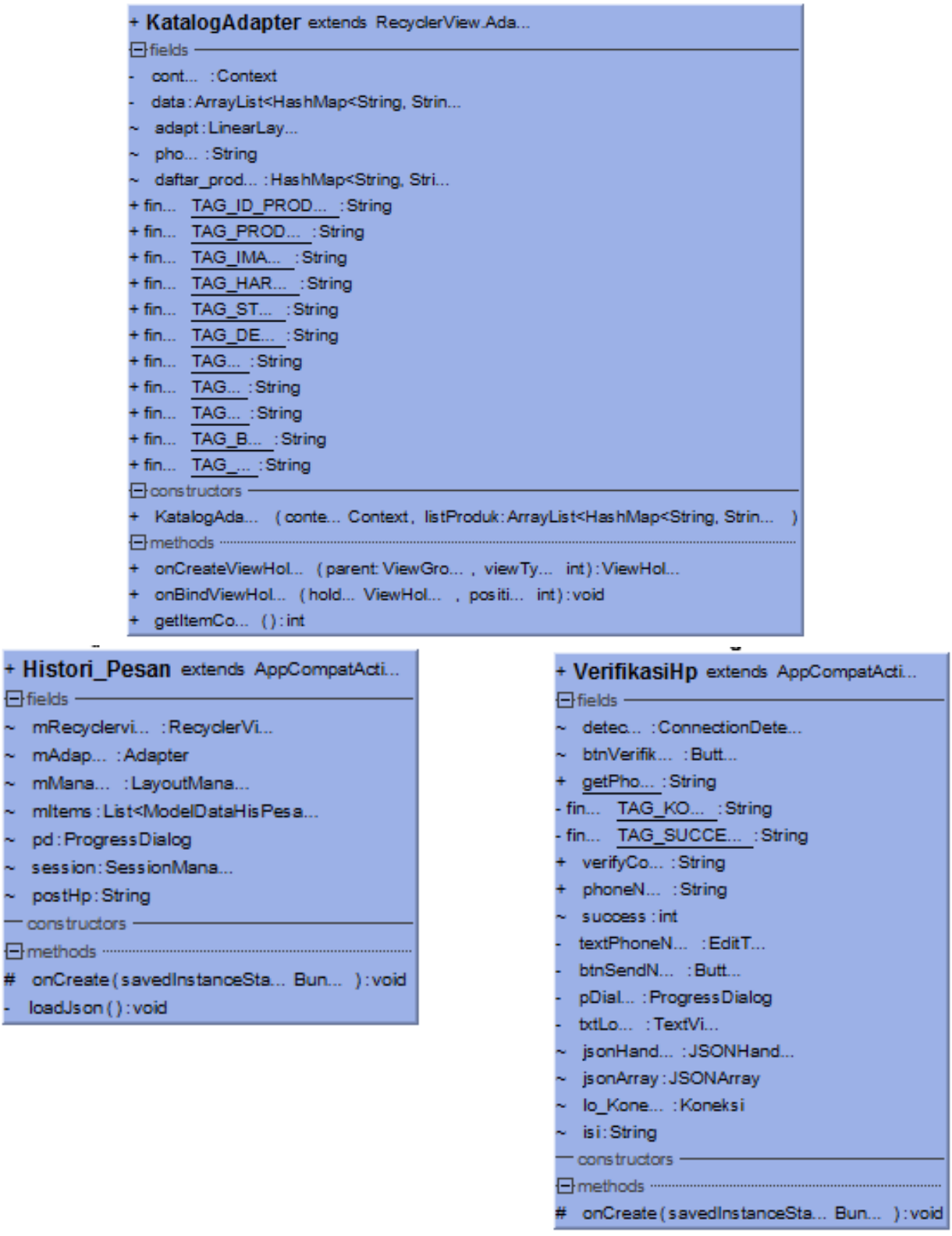
Gambar 3.10 Detail Class Diagram (CD) Lanjutan

Gambar 3.11 Detail Class Diagram (CD) Lanjutan

Gambar 3.12 Detail Class Diagram (CD) Lanjutan

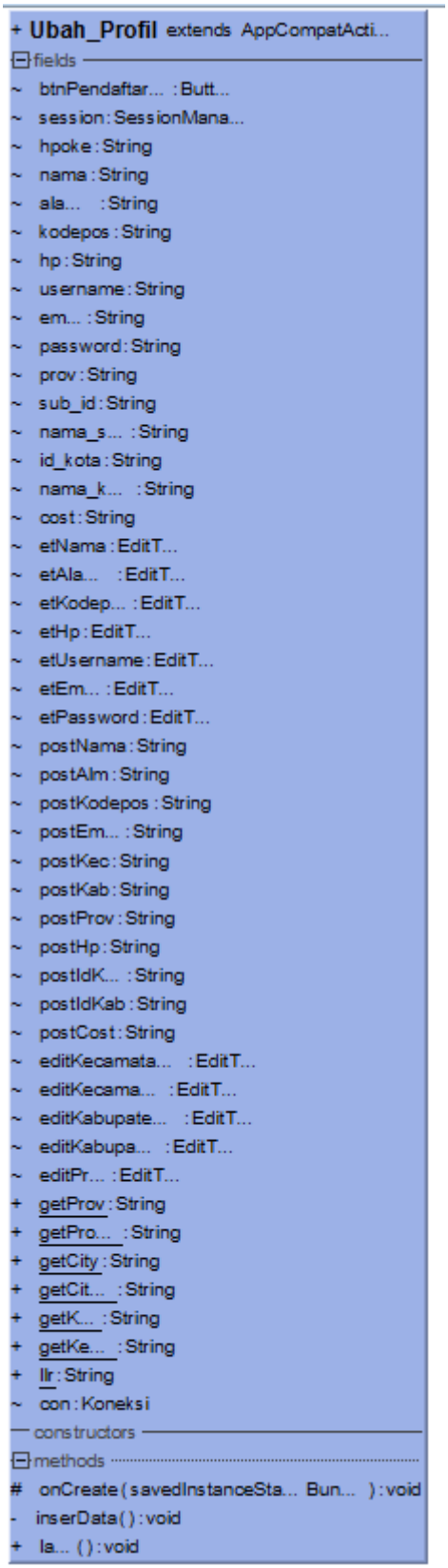
Gambar 3.13 Detail Class Diagram (CD) Lanjutan

Gambar 3.14 Detail Class Diagram (CD) Lanjutan

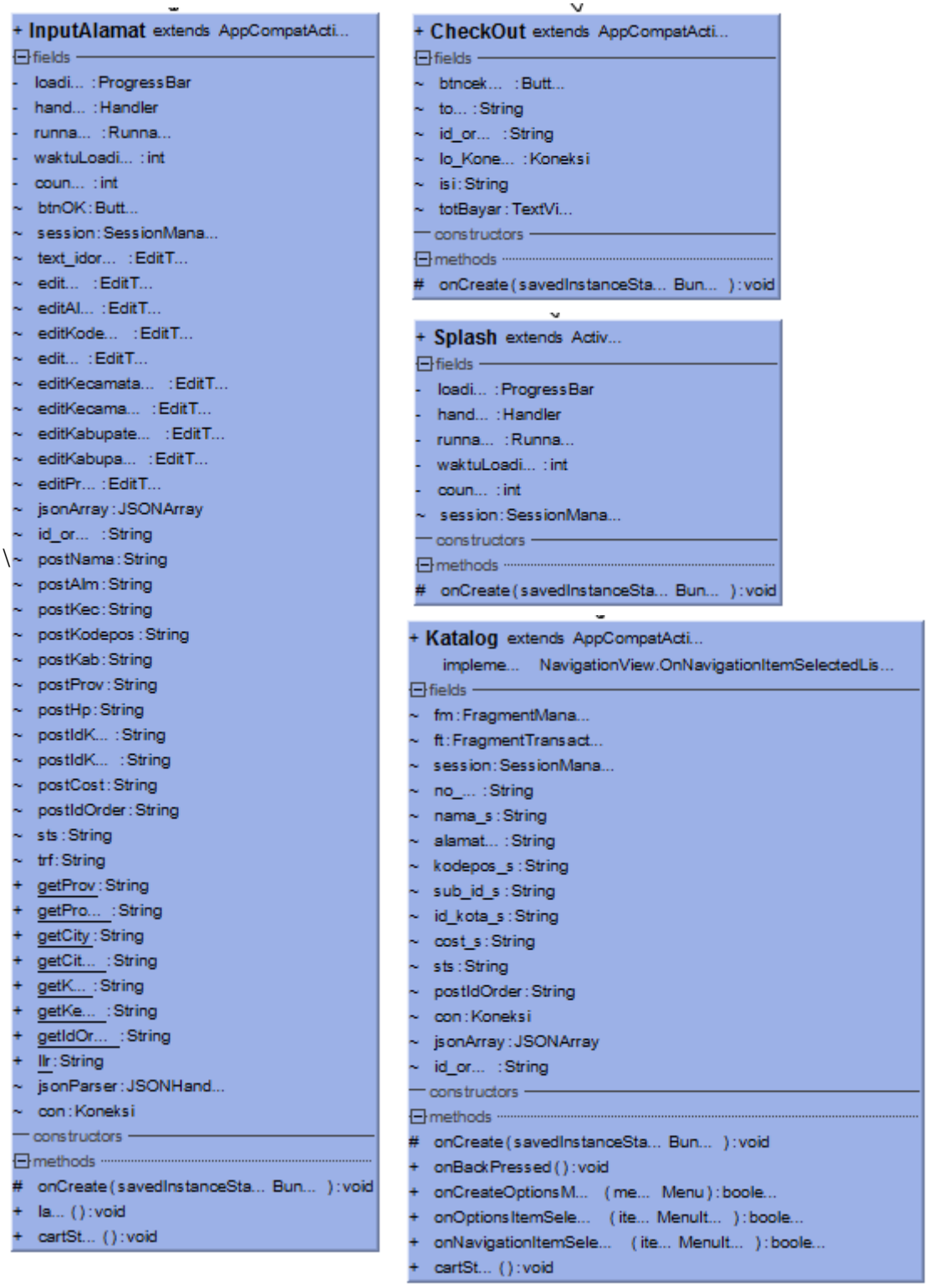


Gambar 3.15 Detail Class Diagram (CD) Lanjutan

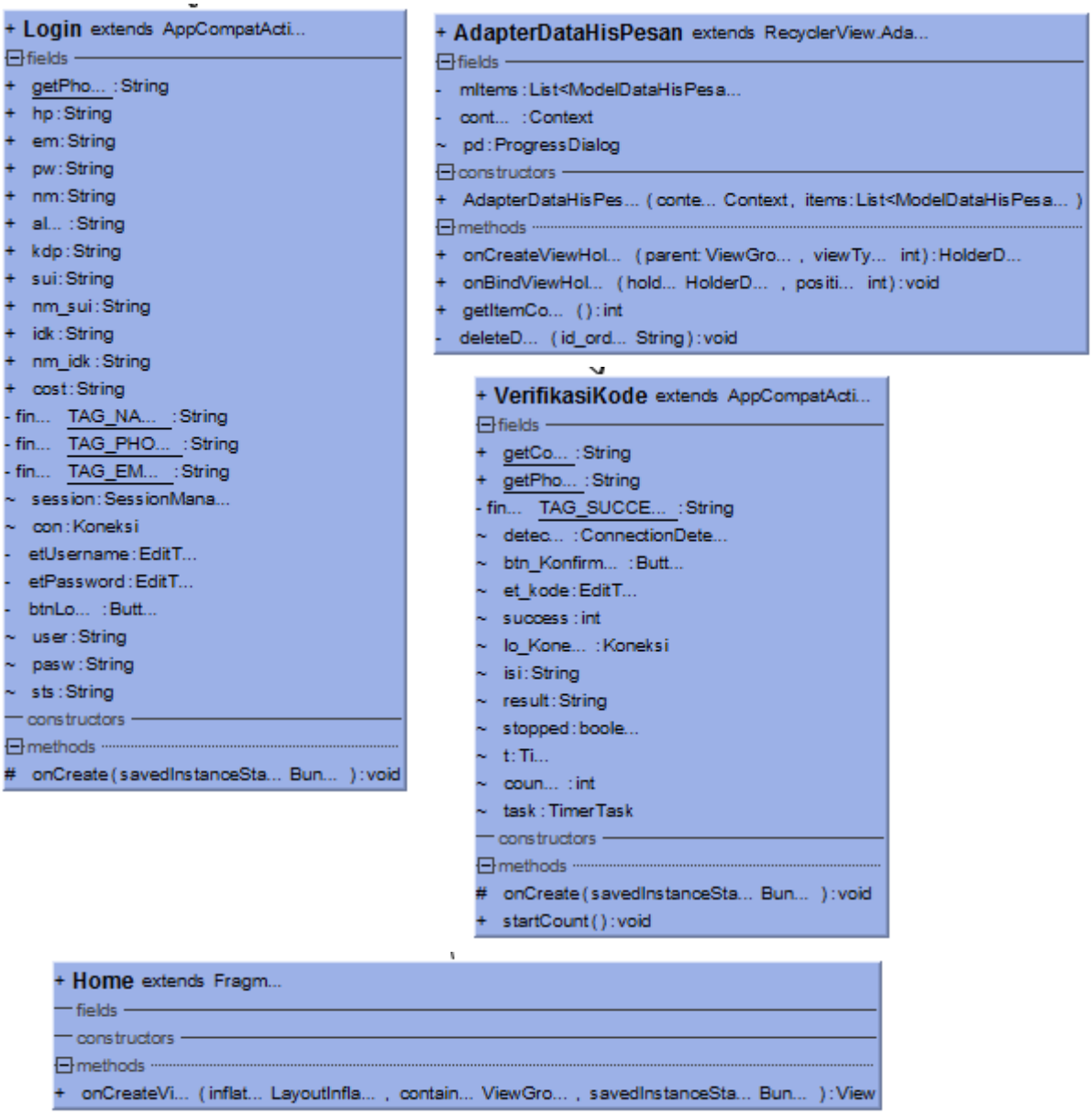
Gambar 3.16 Detail Class Diagram (CD) Lanjutan



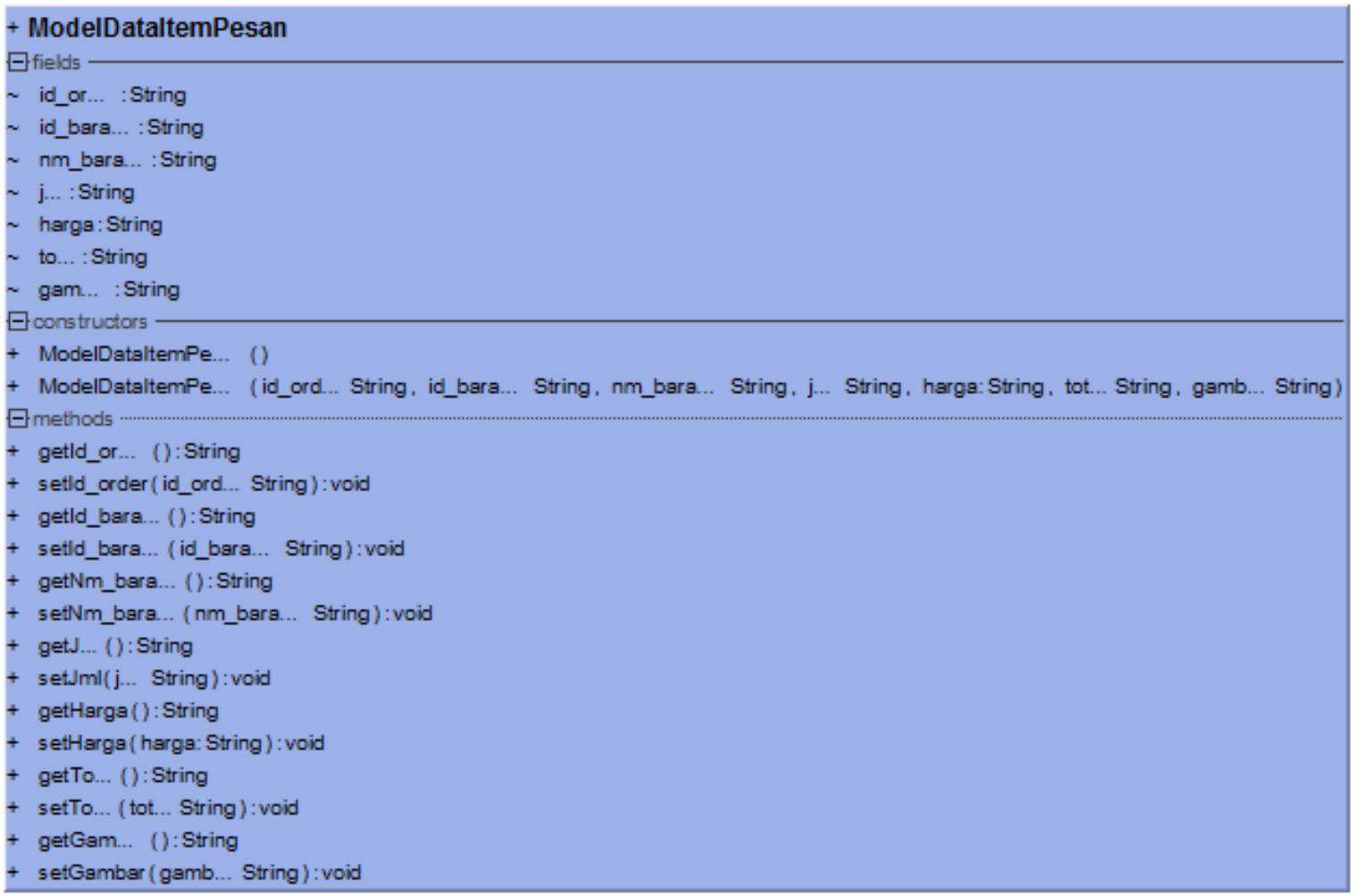
Gambar 3.17 Detail Class Diagram (CD) Lanjutan



Gambar 3.18 Detail Class Diagram (CD) Lanjutan

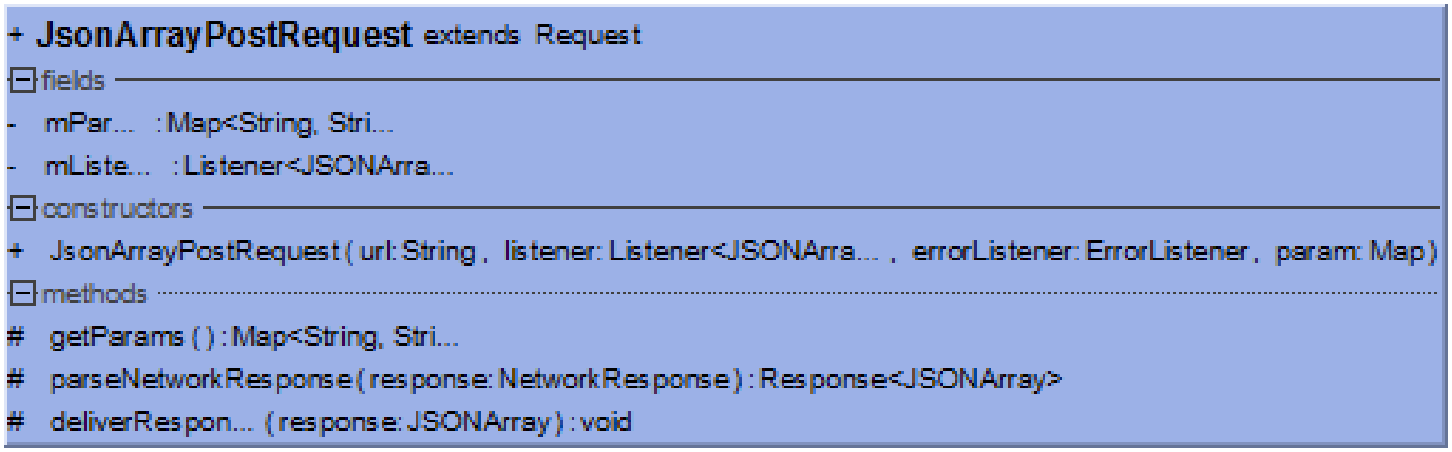


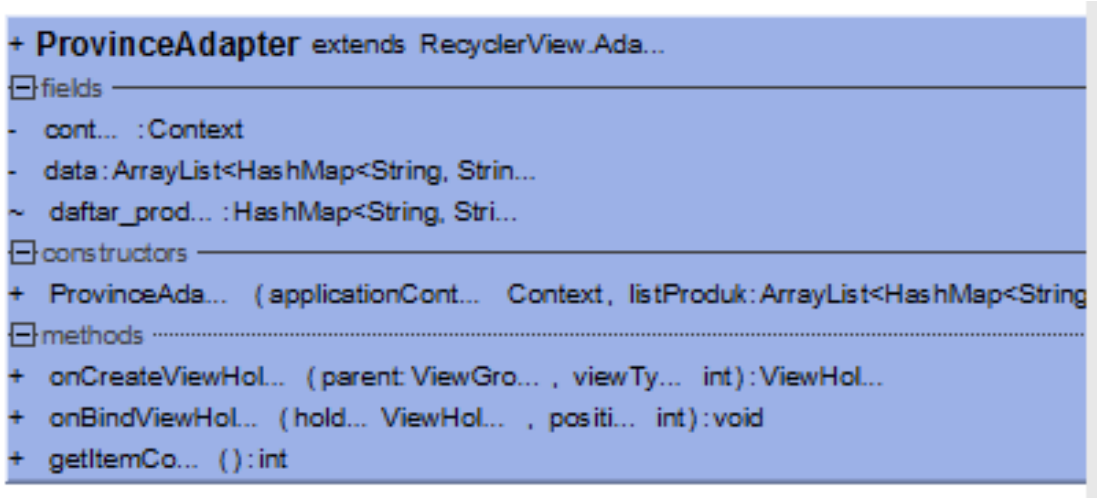
Gambar 3.19 Detail Class Diagram (CD) Lanjutan



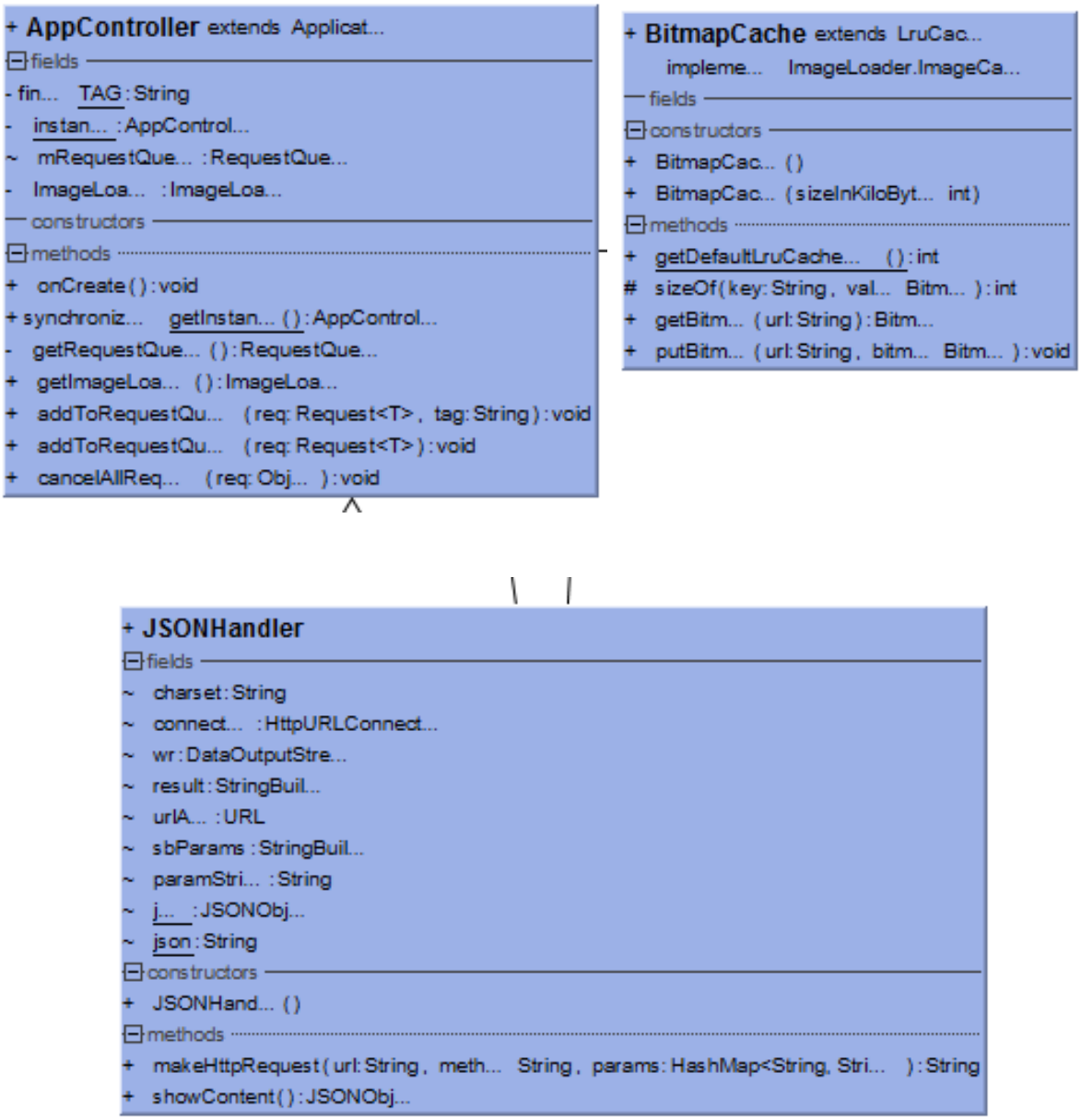
Gambar 3.20 Detail Class Diagram (CD) Lanjutan

Gambar 3.21 Detail Class Diagram (CD) Lanjutan

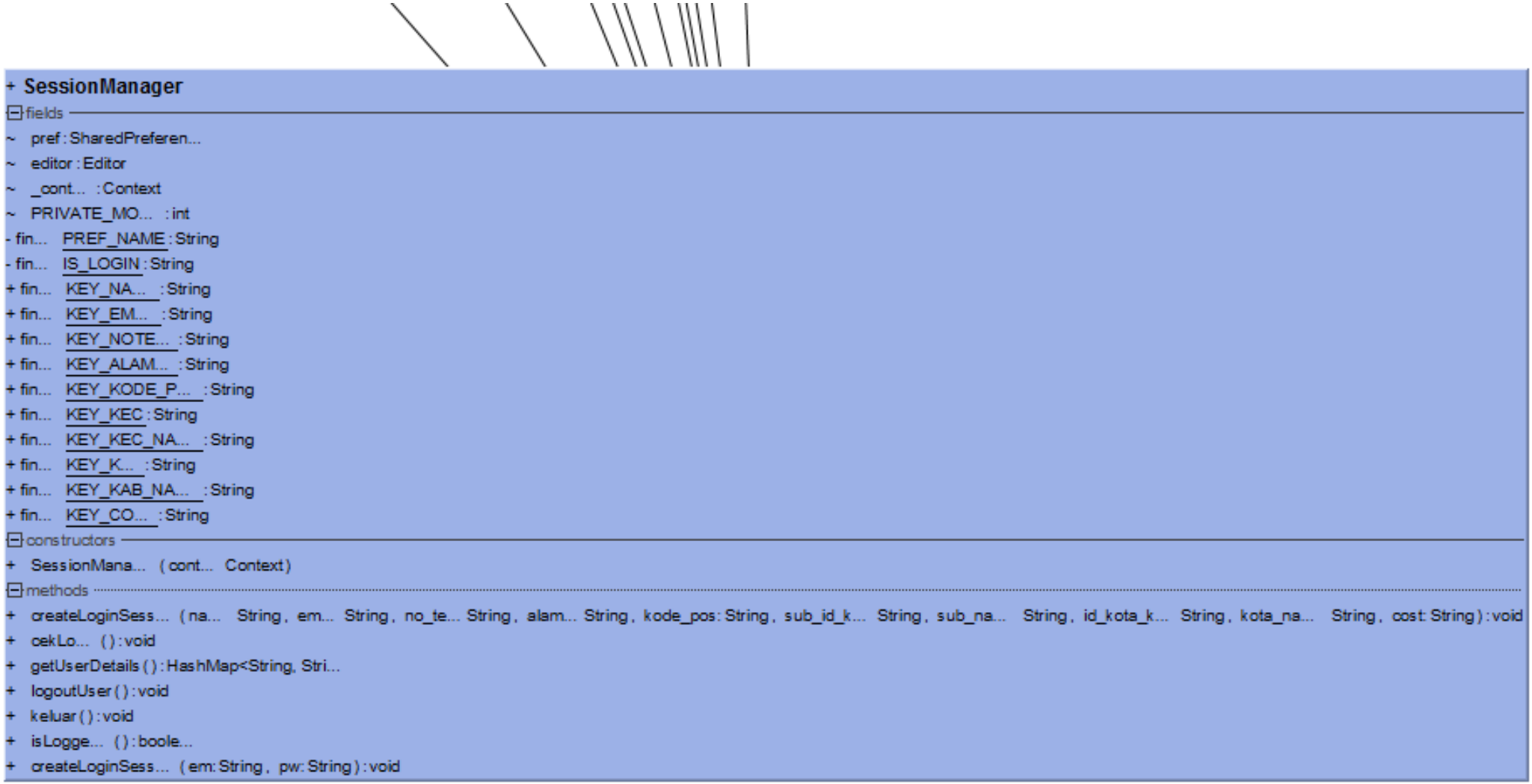




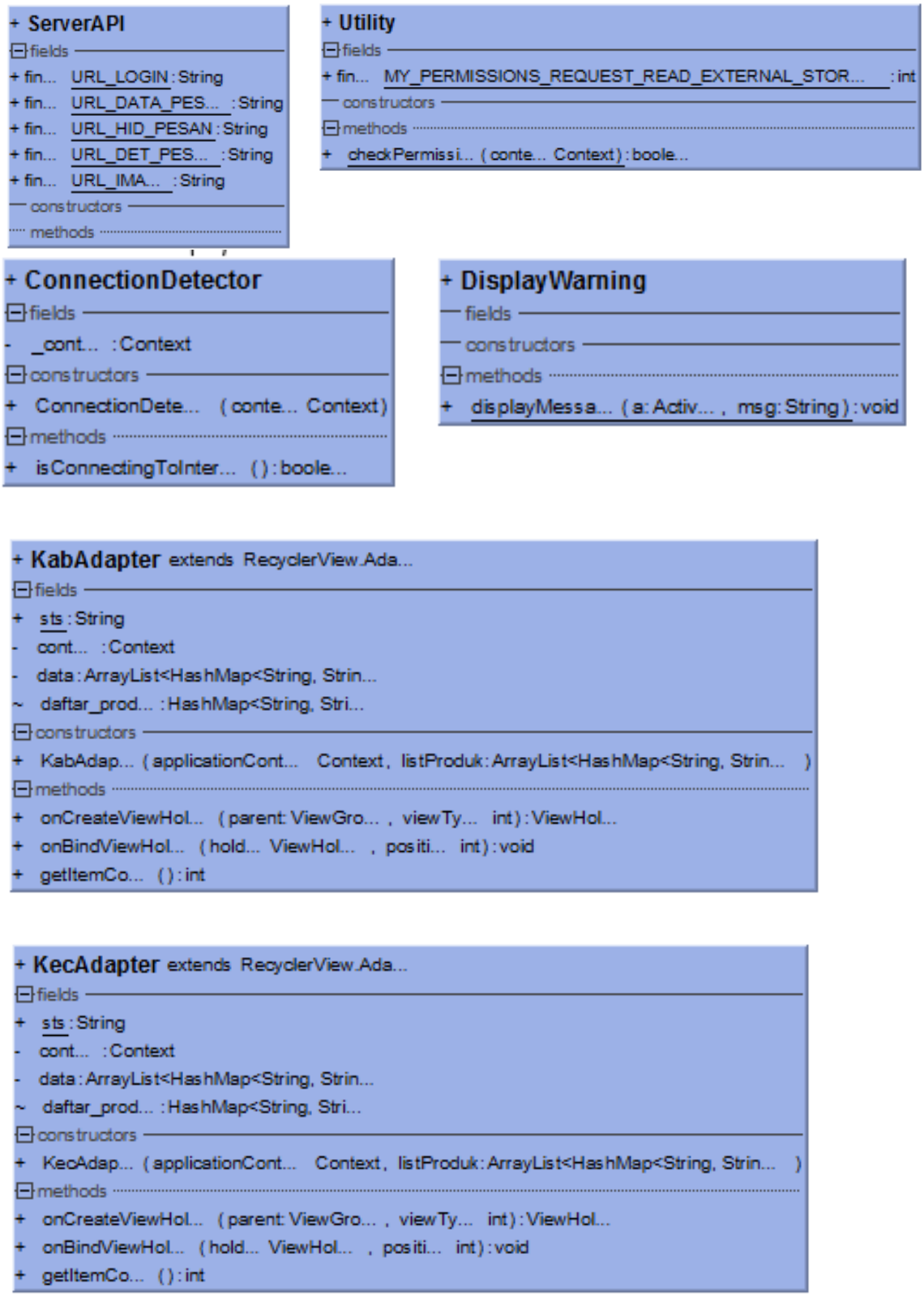
Gambar 3.22 Detail Class Diagram (CD) Lanjutan



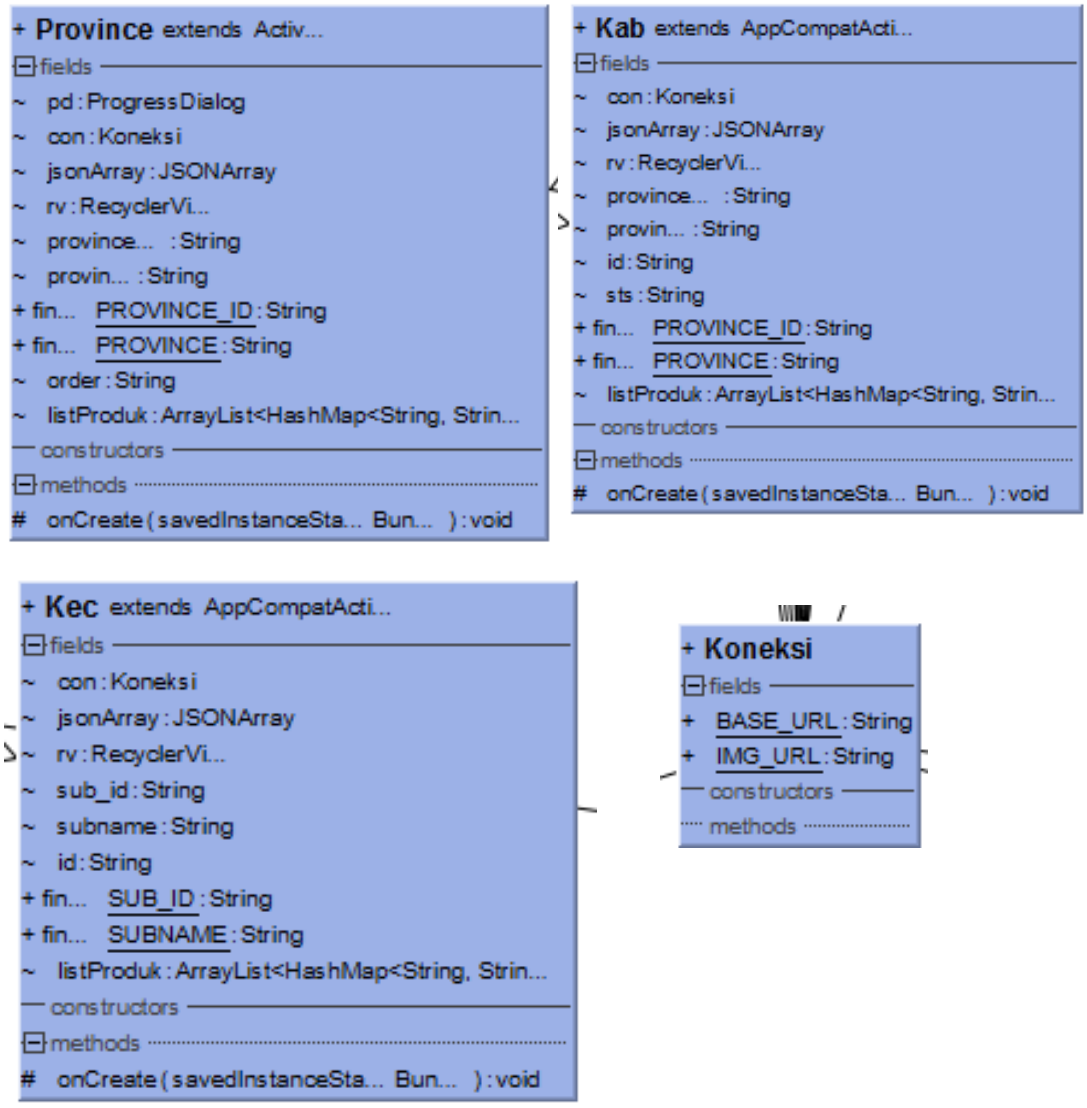
Gambar 3.23 Detail Class Diagram (CD) Lanjutan



Gambar 3.24 Detail Class Diagram (CD) Lanjutan



Gambar 3.25 Detail Class Diagram (CD) Lanjutan



Gambar 3.26 Detail Class Diagram (CD) Lanjutan

Berikut penjelasan dari gambar 3.7-3.26:

Gambaran *class diagram* yang digunakan dalam aplikasi dapat dilihat pada Gambar 3.7-3.26, dan berikut penjelasannya:

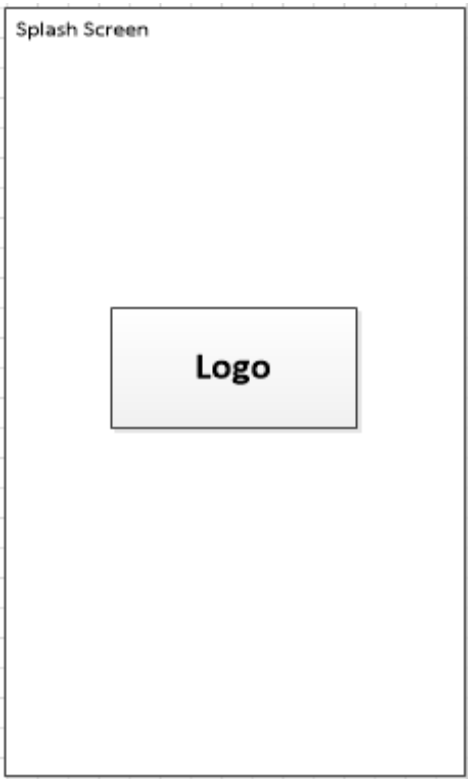
1. Splash berfungsi menampilkan screen aplikasi awal.
2. VerifikasiHp berfungsi menampilkan input no hp yang akan diverifikasi.
3. VerifikasiKode berfungsi menampilkan input kode verifikasi supaya dapat mengetahui no hp yang sebelumnya di input, aktif atau tidak, sehingga aplikasi dapat dijalankan.
4. Pendaftaran berfungsi menampilkan input data pendaftaran member baru yang harus sesuai dengan ktp.
5. Login berfungsi menampilkan halaman login dengan menginput email dan password.
6. Katalog berfungsi menampilkan navigasi berupa home, menu user dan kategori.
7. Ubah_profil berfungsi menampilkan profil yang dapat dirubah.
8. PeralatanRumahTangga berfungsi menampilkan produk-produk yang dijualkan di toko 23.
9. DeskripsiProduk berfungsi menampilkan penjelasan dan stok dari produk yang dijual.
10. DetKeranjang berfungsi menampilkan penyelesaian pembayaran yang berupa data alamat yang akan dikirim, pemilihan kurir dan total pembayaran yang harus dibayarkan.
11. Histori_Pesan berfungsi menampilkan kode barang yang sudah dibeli serta alamat kirim dan total pembayaran.
12. Histori_Det_Pesan berfungsi menampilkan produk-produk yang sudah dibeli dan histori ini member dapat melacak keberadaan produk yang sudah dikirim telah sampai mana.

3.4 Perancangan Interface

Perancangan *interface* sistem diperlukan untuk memudahkan *user* dalam melakukan proses interaksi terhadap sistem. *Interface* menyediakan tampilan halaman sebuah sistem yang digunakan untuk proses *input* hingga menghasilkan *output* yang sesuai dengan kebutuhan. *Interface* untuk sistem yang akan dibuat adalah sebagai berikut:

3.4.1 *Halaman awal*

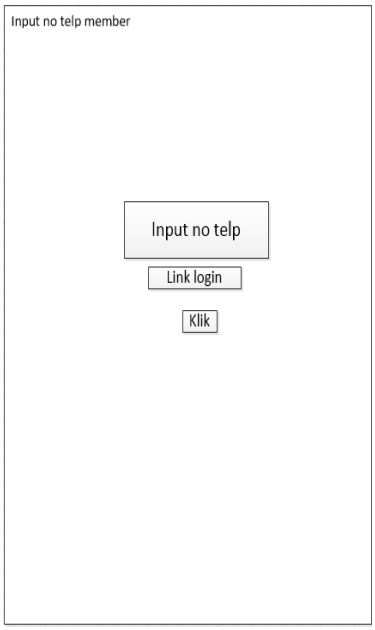
Halaman awal adalah tampilan utama dari aplikasi yang dapat dilihat oleh *member* dan *admin.* Gambaran halaman utama aplikasi dapat dilihat pada Gambar 3.27.



Gambar 3.27 Halaman Utama

2. *Halaman Member Baru*

Halaman *member* baru berisi halaman yang berkaitan dengan ketentuan pendaftaran yang sebelumnya diminta *input* nomor telphone terlebih dahulu yang dapat dilihat oleh calon *member*. Gambaran halaman *member* baru terdapat pada Gambar 3.28.



Gambar 3.28 Halaman *Member* Baru

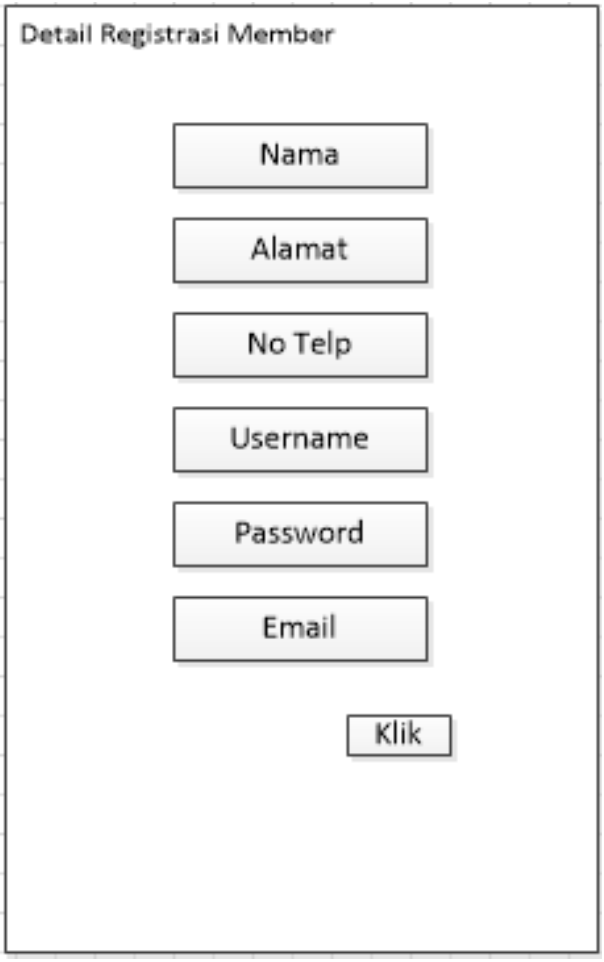
3. *Halaman Konfirmasi Member Baru*

Halaman konfirmasi *member* baru yang dapat dilihat oleh *member* ketika telah selesai melakukan *input* nomor telphone agar terdapat dalam data yang terdaftar sebagai calon pembeli yang akan membeli produk. Gambaran halaman konfirmasi *member* baru dapat dilihat pada Gambar 3.29.

Gambar 3.29 Halaman Konfirmasi *Member* Baru

4. *Halaman Registrasi Aktifasi*

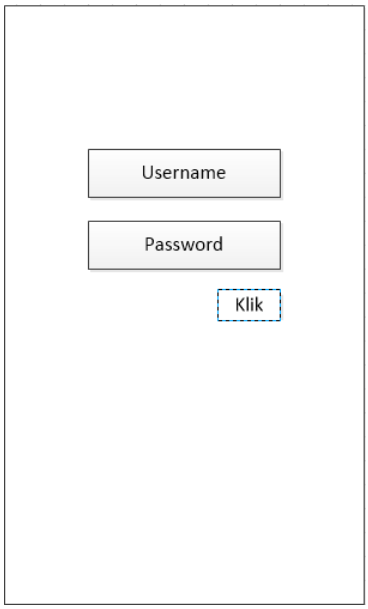
Halaman registrasi *aktifasi* yang dapat dilihat oleh *member* ketika telah selesai melakukan pendaftaran *member* baru. Terdapat *button* nama, *button* alamat, *button* nomor telphone, *button* username, *button* password dan *button* email yang harus diisi. Langkah selanjutnya *member* diwajibkan untuk membuka *email* agar akun bisa terverifikasi. Gamaran halaman *member* baru dapat dilihat pada Gambar 3.30.



Gambar 3.30 Halaman Registrasi Aktifasi

5. *Halaman* Login Member

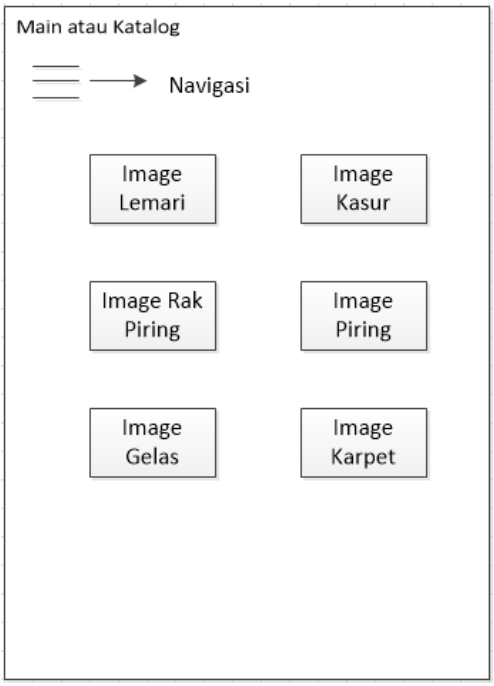
Halaman *login member* terdapat dua kolom dan satu *button* yaitu kolom *username, password* dan *button* masuk. *Member* dapat *login* sesuai *username* dan *password* yang telah dibuat sebelumnya. Gambaran halaman *login member* terdapat pada Gambar 3.31.



Gambar 3.31 Halaman *Login Member*

6. *Halaman Katalog Produk*

Halaman *katalog produk* yang menampilkan navigasi tentang profil toko, kontak toko sedangkan image produk memberikan informasi kategori produk yang tersedia. Gambar halaman katalog produk terdapat pada Gambar 3.32.



Gambar 3.32 Halaman Katalog Produk

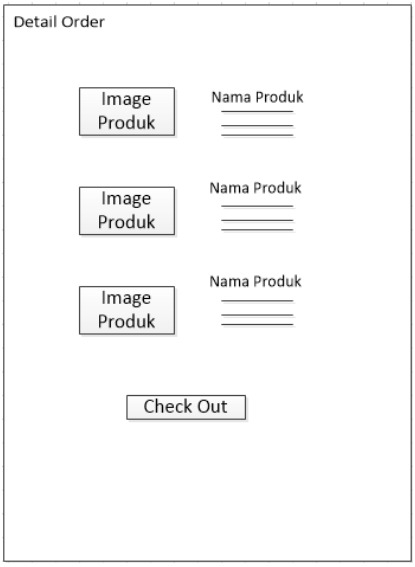
7. *Halaman Detail Produk*

Halaman detail produk merupakan tampilan produk yang akan dibeli oleh *member*, *detail* produk ini menampilkan deskripsi produk pada gambar produk yang di klik sebelumnya dan menjelaskan ukuran produk yang akan dipesan. Jika pembeli berminat membeli maka ditampilan tersebut disediakan *button order*. Gambar halaman detail produk terdapat pada Gambar 3.33.

Gambar 3.33 Halaman Detail Produk

8. *Halaman Keranjang Belanja*

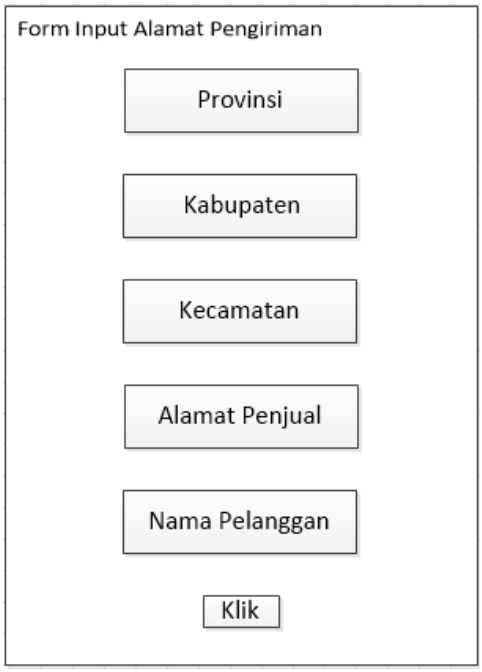
Halaman keranjang belanja terdapat data barang dengan rincian biaya yang dibeli oleh *member*, lalu terdapat juga *button check out* yang merupakan *button* untuk memastikan barang yang akan dipesan. Gambaran halaman keranjang belanja terdapat pada Gambar 3.34.



Gambar 3.34 Halaman Keranjang Belanja

9. *Halaman Alamat Pengiriman*

Halaman alamat pengiriman terdapat pengisian nama *member* alamat *member*, nomor telphone yang akan dikirimkan barang tersebut, lalu email untuk notifikasi tagihan pembayaran. Gambaran halaman alamat pengiriman terdapat pada Gambar 3.35.



Gambar 3.35 Halaman Alamat Pengiriman

3.5 Coding

Dalam tahap ini penulisan kode program atau *coding* merupakan penerjemahan *design* dalam bahasa yang bisa dikenali oleh komputer. Kode program ini biasanya dilakukan oleh *programmer* yang akan meterjemahkan transaksi yang diminta oleh *user.*

3.6 Testing

Testing disini merupakan tahapan pengujian terhadap *coding* yang telah dibuat oleh programmer dan jika terdapat kesalahan maka akan diketahui agar dapat diperbaiki.

3.7 Maintenance

Perangkat lunak yang sudah disampaikan kepada pelanggan pasti terdapat perubahan yang bisa karena perangkat lunak itu sendiri mengalami kesalahan dengan menyesuaikan lingkungan (periperal atau sistem operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional.