BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis dan perancangan sistem memerlukan tahapan yang sistematis untuk mendapatkan aplikasi yang baik dan bersesuaian dengan kegunaan dan tujuannya. Tahap awal dari analisis adalah menganalisis kebutuhan-kebutuhan sistem mulai dari kebutuhan pengguna, kebutuhan non fungsional, dan kebutuhan fungsional. Sedangkan untuk tahap perancangan aplikasi yaitu perancangan *database*, dan perancangan antarmuka.

3.1.1 Analisis Masalah

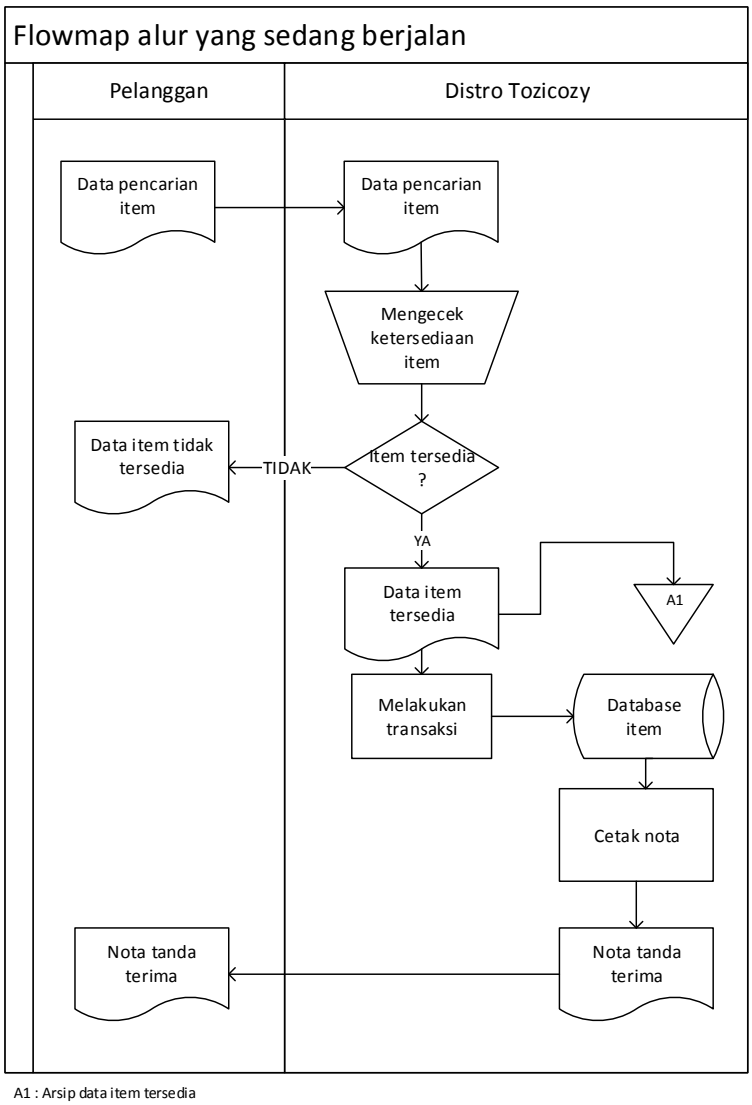
Antusias masyarakat terutama kalangan muda terhadap berdirinya distro sangat besar hal ini di karenakan harga dan kualitas dari produk-produk yang ditawarkan sangat bersaing. Selain hal tersebut, pelayanan untuk pelanggan menjadi suatu prioritas yang diutamakan. Akan tetapi fenomena yang terjadi sekarang ini masih terdapat kekurangan dalam hal kreatifitas untuk melayani pelanggan. Terkadang para pelanggan menemui kesulitan-kesulitan pada saat mereka berbelanja. Misalnya, jika pelanggan menemukan model pakaian yang disukai akan tetapi ukuran atau warna dari pakaian tersebut tidak sesuai dengan keinginan pelanggan maka pelanggan harus mengunjungi distro cabang yang lain dimana akan menyita banyak waktu dan tidak praktis.

Melihat permasalahan diatas, distributor outlet (distro) seharusnya memiliki sebuah sistem yang efektif, cepat dan mudah digunakan oleh pelanggan dalam menyampaikan informasi yang dibutuhkan oleh pelanggan. Dan hal ini pun dapat meningkatkan penjualan dan tingkat kepuasan pelanggan. Oleh karena itu dibutuhkan suatu sistem yang dapat menjawab kebutuhan ini, yaitu sistem yang memanfaatkan fitur peta digital dan pembangkit rute serta cepat dan mudah digunakan dalam memberikan informasi tersebut yaitu informasi yang disajikan melalui media aplikasi *mobile*.Karena menggunakan

perangkat *mobile* dapat digunakan kapan saja dan dimana saja dengan jaringan internet.

3.1.2 Sistem Yang Sedang Berjalan

Pada bagian ini akan menjelaskan tentang prosedur yang sedang berjalan dalam proses pembelian pakaian yang dilakukan antara konsumen dengan Distro. Berikut adalah Gambar 3.1 *Flowmap* Prosedur pembelian pakaian yang sedang berjalan :



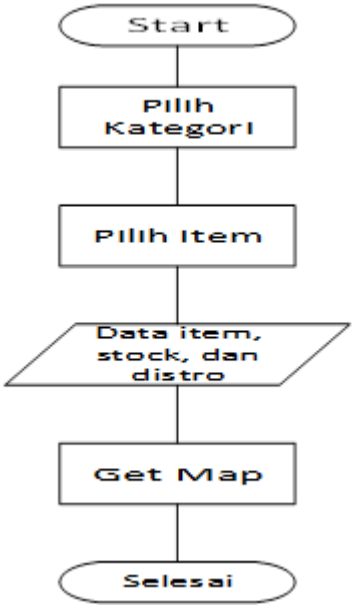
Gambar 3.1 *Flowmap* Alur sistem kerja Tozicozy yang sedang berjalan

Pada Gambar 3.1 dijelaskan *Flowmap* Alur sistem kerja Tozicozy yang sedang berjalan saat ini. Berikut adalah penjelasan dari setiap tahapan – tahapannya:

1. Pelanggan mendatangi Distro ToziCozy.
2. Pelanggan menanyakan ketersediaan item yang di cari kepada pegawai.
3. Pegawai distro mengecek ketersediaan item.
4. Jika item yang dicari oleh pelanggan itu ada maka proses transaksi dilakukan.
5. Jika item yang dicari pelanggan tidak ada atau kosong maka pelanggan di berikan data bahwa item tidak tersedia.
6. Transaksi selesai.

3.1.3 Sistem Yang Diajukan

Perancangan aplikasi *online service* pada distro Tozicozy ini memiliki beberapa alur proses dimana pada aplikasi ini pengguna dapat dimudahkan dalam proses informasi data barang, data stock dan data distro yang memiliki barang-barang ToziCozy. Penggunaan *online service* pada pembangunan aplikasi ini juga sangat memudahkan pengguna dalam melakukan pemesanan dan mengetahui letak lokasi Tozicozy. Berikut adalah gambaran alur proses sistem yang digambarkan dalam bentuk *flowchart* pada gambar 3.2 :



Gambar 3.2 *Flowchart* Aplikasi

3.1.4 Analisis Aturan Bisnis Pada Aplikasi *Mobile*

Analisis aturan bisnis berisi tentang pemaparan proses bisnis perangkat lunak Distro ToziCozy. Berikut adalah penjelasan setiap tahapannya :

1. Pengguna harus mengaktifkan fitur lokasi (*GPS)* pada *smartphone android*.
2. Pada menu *home* pengguna dapat melihat kategori jenis item yang dicari.
3. Pengguna mencari item yang diinginkan pada kolom penjabaran kategori.
4. Pengguna dapat langsung memilih item.
5. Sistem akan menampilkan informasi pakaian, ketersediaan stok dan distro yang menyediakan.
6. Pada menu *store* pengguna dapat mengetahui lokasi cabang – cabang distro.
7. Pengguna dapat mengetahui rute menuju cabang distro.
8. Pengguna keluar dari aplikasi dengan memilih fitur keluar.

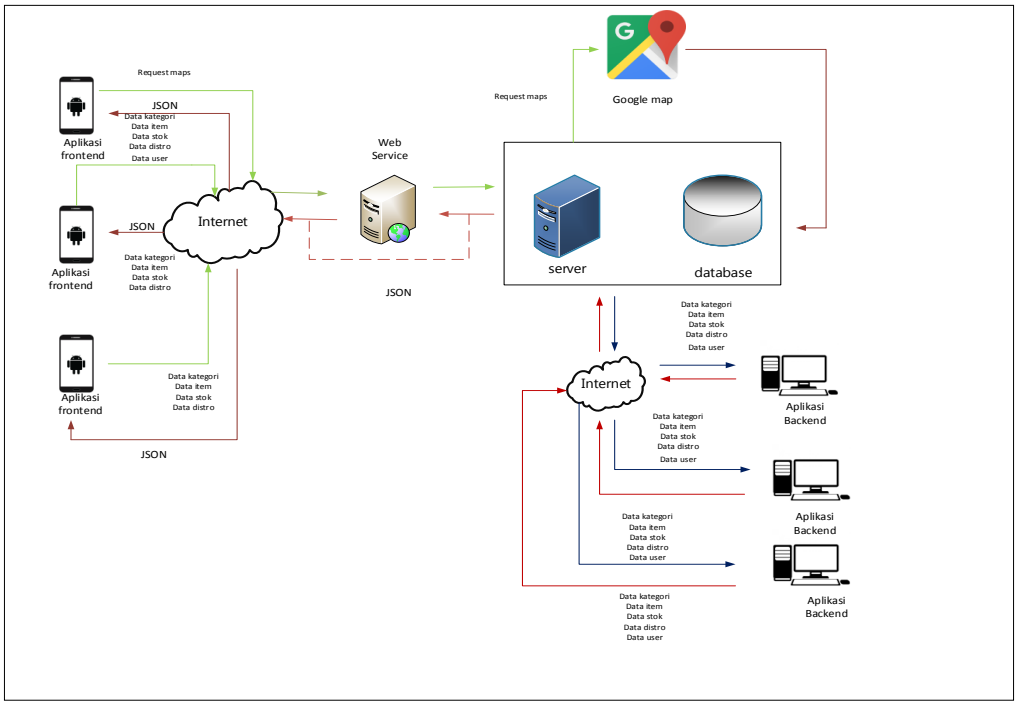
3.1.5 Analisis Aturan Bisnis Pada *Web*

Analisis aturan bisnis adalah pemaparan proses bisnis pada perangkat lunak berbasis *web*. Berikut ini adalah setiap tahapannya :

1. User diharuskan melakukan *login* terlebih dahulu untuk proses autentifikasi.
2. Pada halaman utama aplikasi terdapat beberapa pilihan menu yaitu :
    1. Master Item yang berfungsi sebagai data item. Pada menu ini user dapat melakukan proses *input, update* dan *delete* data.
    2. Master Kategori yang berfungsi sebagai data kategori. Pada *menu* ini user dapat melakukan proses *insert* dan *update* data kategori yang akan terhubung dengan *menu* master *item* .
    3. Master Distro yang berfungsi sebagai data distro. Pada menu ini user dapat melakukan proses *insert, update* dan *delete* data distro.
    4. Master *User* yang berfungsi sebagai data informasi *user*.
    5. Menu transaksi berfungsi untuk menginputkan data transaksi yang telah menjadi nota atau bukti pembayaran untuk mengurangi stok yang di inputkan secara manual oleh user.

3.1.6 Perancangan Arsitektur Sistem

Analisis arsitektur sistem bertujuan untuk mengidentifikasi arsitektur yang akan dibangun berdasarkan dua buah sistem *web* dan *mobile*. Berikut adalah Gambar 3.3 yang mengambarkan arsitektur perangkat lunak secara keseluruhan.



Gambar 3.3 Perancangan Arsitektur Sistem Keseluruhan

Keterangan :

- Request data dari platform mobile ke server
- Respon data dari server ke platform mobile JSON
- Request data dari platfrom web ke server
- Respon data dari server ke platform web

Berikut adalah deskripsi dari Gambar 3.3 Perancangan Arsitektur Sistem Keseluruhan :

1. Aplikasi *Frontend*

   Aplikasi *frontend* merupakan aplikasi yang berfungsi sebagai antarmuka antara pelanggan dengan sistem yang menampilkan informasi tentang item yang terdapat pada database server. Pada aplikasi *frontend* ini terdapat beberapa menu yaitu ada menu lihat kategori, pilih item , lihat stok,dan menu lihat distro.

2. Internet

   Internet merupakan penghubung terjadinya permintaan data ke server dan mengirimkan data yang diminta melalui jaringan internet ke komputer dan pengguna *android*.

3. *Web Service*

   Dengan *web service*, aplikasi *frontend* dapat mengakses data dari *backend*, ketika data yang ada terdapat dalam *backend* diubah maka data yang terdapat pada aplikasi *frontend* pun ikut berubah tanpa harus melakukan *updating database* sistem lokal *frontend* .

4. Aplikasi *Backend*

   Aplikasi *backend* berfungsi sebagai penghubung antara aplikasi *frontend* dengan database server. Pengguna aplikasi backend ini adalah user dari masing-masing setiap Distro. Pada aplikasi backend ini terdapat beberapa tampilan yaitu login, master item, master kategori, master distro , input stok.

5. *Google Maps*

   Jika server menerima permintaan lokasi maka permintaan data akan diteruskan ke server *google maps*. Untuk mengetahui lokasi cabang distro terdekat.

3.2 Analisis JSON

JSON (*JavaScript Object Notation*) merupakan format yang digunakan untuk pertukaran atau parsing data seperti halnya *XML*. Formatnya berbasi teks dan mudah dibaca oleh manusia serta digunakan untuk mempresentasikas struktur data sederhana dan larik asosiatif (disebut objek). Format JSON sering digunakan untuk mentransmisikan data terstruktur melalui suatu koneksi karingan pada suatu proses yang disebut serialisasi.

Langkah-langkah parsing data dengan *JSON*:

1. Aplikasi *backend* mengakses data dari *database*

   Pada *PHP versi 5.2.x* sudah mengikutsertakan ektensi *JSON* jadi tidak perlu lagi menambahkan alat bantu untuk penerapan *JSON* di *PHP*. Selain itu *PHP* juga sudah menyertakan dua fungsi untuk menulis dan membaca format *JSON*

yaitu *json_encode* dan *json_decode* untuk mengubah variabel *PHP* kedalam format *JSON*. *PHP* akan mengakses data yang ada dalam *database* dan menampilkannya dalam bentuk *array*. Selanjutnya *array* tersebut akan di *encode* ke dalam format *JSON* dengan perintah *json_encode*, maka akan menghasilkan *array* dalam bentuk format *JSON*.

2. Aplikasi *frontend* mengakses aplikasi *backend*

   Disini aplikasi *frontend* akan mengakses alamat URL dari file *PHP* yang ada pada aplikasi *backend* dan mengambil data yang telah diubah ke dalam format *JSON* untuk disimpan sementara pada *database frontend* sebelum ditampilkan kepada konsumen sehingga konsumen bisa melakukan pemesanan makanan dan minuman dari informasi yang ditampilkan.

3.2.1 Analisis Web Service

*Web Serv*ice dapat menjalankan operasi-operasi termasuk akses data, *update database*, mengatur integritas data, mengirim respon dari *request* yang dikirim oleh *user*. *Web serv*ice tidak memiliki tampilan karena *web serv*ice termasuk dalam *Bussiness-Serv*ice tier. Artinya di dalam *web serv*ice hanya tersedia fungsi-fungsi yang nantinya dapat digunakan oleh aplikasi lainnya. Pada aplikasi ToziCozy berguna sebagai perantara untuk mendapatkan data yang akan diambil dari *database server* yang diakses melauli *mobile dev*ice. Data yang diminta akan menampilkan informasi dengan format *JSON*.

3.2.2 Analisis Longitude Latitude

Longitude dan latitude adalah suatu sistem koordinat geografis yang digunakan untuk menentukan lokasi suatu tempat di permukaan bumi. Latitude atau garis lintang adalah garis yang menentukan lokasi berada di sebelah utara atau selatan ekuator. Garis lintang diukur mulai dari titik 0 derajat dari khatulistiwa sampai 90 derajat di kutub. Longitude atau garis bujur adalah digunakan untuk menentukan lokasi di wilayah barat atau timur dari garis utara selatan yang sering disebut juga garis meridian. Garis bujur diukur dari 0 derajat di wilayah Greenwich sampai 180 derajat di International Date Line.Implementasi proses logitude dan latitude ini digunakan pada pemakaian Google Maps. Aplikasi Google Maps dibuat dengan

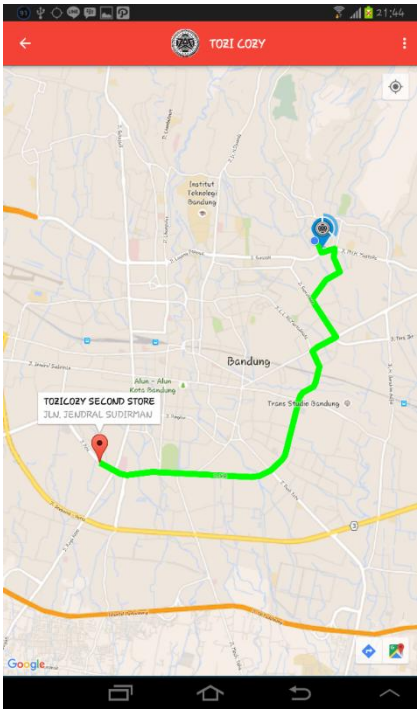
menggunakan kombinasi dari gambar peta, database serta obyek-obyek interaktif yang dibuat dengan bahasa pemrograman HTML, Javascript, AJAX, dan bahasa pemrograman lainnya.

Gambar- gambar peta yang muncul pada layar merupakan hasil komunikasi dari pengguna dengan database pada web server google untuk menampilkan gabungan dari potongan – potongan gambar yang diminta. Keseluruhan citra yang ada diintegrasikan ke dalam suatu database pada google server, yang nantinya akan dipanggil sesuai kebutuhan permintaan. Bagian – bagian gambar map merupakan gabungan dari gambar – gambar yang berukuran 256 x 256 pixels.

Tiap 256 x 256 tile mewakili gambar tertentu dalam longitude latitude dan zoom level tertentu. Informasi tersebut dirangkum dalam alamat URL dari tile. Sehingga untuk menampilkan peta suatu daerah tertentu, dapat dilakukan dengan mengirim URL tile yang diinginkan. Pada aplikasi online service distro tozi cozy ini proses penghitung koordinat langitude dan latitude sangat membantu dalam pencarian letak lokasi yang ingin dituju. Google maps akan membantu megintegrasikan permintaan user dengan database dalam google server. Selanjutnya mengolahnya dalam bentuk citra yang merupakan gabungan gambar – gambar dalam titik koordinat longitude dan latitude.

3.2.3 Directions

Directions adalah utility yang menangani masalah pencarian rute. Directions ini digunakan untuk memandu rute jalan dari lokasi pelanggan menuju tempat lokasi distro yang menyediakan stok item pada peta *Google Maps* seperti di gambar 3.4



Gambar 3.4 Contoh Penggunaan *Direction*

3.2.4 Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan perangkat lunak terbagi ke dalam dua kebutuhan yaitu kebutuhan fungsional dan kebutuhan non fungsional. Kebutuhan fungsional dapat dilihat pada tabel 3.1 Spesifikasi Kebutuhan Fungsional, sebagai berikut :

Tabel 3.1 Spesifikasi Kebutuhan Fungsional

Kode SKPL	Spesifikasi Kebutuhan Perangkat Lunak
SKPL-F-001	Sistem *mobile* menyediakan fasilitas bagi pengguna untuk memilih kategori
SKPL-F-002	Sistem *mobile* menyediakan fasilitas bagi pengguna mencari pakaian yang diinginkan
SKPL-F-003	Sistem *mobile* menyediakan fasilitas bagi pengguna untuk mencari lokasi distro

Kode SKPL	Spesifikasi Kebutuhan Perangkat Lunak
SKPL-F-005	Sistem *web* distro menyediakan fasilitas *login* ke dalam sistem untuk *administrator*
SKPL-F-006	Sistem *web* disro menyediakan fasilitas untuk admin dalam mengolah data *item*
SKPL-F-007	Sistem *web* distro menyediakan fasilitas untuk admin mengolah data distro
SKPL-F-008	Sistem *web* distro menyediakan fasilitas untuk admin dalam mengolah data stok
SKPL-F-009	Sistem *web* distro menyediakan fasilitas untuk admin dalam mengolah data stok

Kebutuhan non Fungsional pada perangkat lunak yang dibangun dapat dilihat pada tabel 3.2 Spesifikasi Kebutuhan non Fungsional, sebagai berikut :

Tabel 3.2 Spesifikasi Kebutuhan non Fungsional

Kode SKPL	Spesifikasi Kebutuhan Perangkat Lunak
SKPL-NF-001	DBMS yang digunakan adalah *My Sql*
SKPL-NF-002	Sistem *backend* dapat diakses melalui browser *Google Chrome, Mozilla Firefox.*
SKPL-NF-003	Sistem *mobile* yang dibangun menggunakan sistem operasi android versi 4.2 *jellybean*

3.2.5 Analisis Kebutuhan non Fungsional

Analisis kebutuhan non-fungsional merupakan analisis yang dibutuhkan untuk dapat menentukan spesifikasi dari kebutuhan sistem. Spesifikasi ini meliputi elemen atau perangkat-perangkat yang dibutuhkan untuk sistem yang akan dibangun sampai sistem tersebut dapat diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk

mengolah masukan sehingga dapat menghasilkan suatu keluaran yang diinginkan. Kebutuhan non-fungsional terbagi menjadi beberapa analisis yaitu analisis perangkat keras, analisis perangkat lunak dan analisis pengguna.

3.2.5.1 Analisis Perangkat Keras (*Hardware*)

Perangkat keras yang dibutuhkan pada perancangan aplikasi ini dilihat berdasarkan kebutuhan minimal yang harus dipenuhi antara lain :

a. Perangkat Keras Pendukung Aplikasi *Backend*

Pada perancangan aplikasi *online service* distro Tozicozy ini mengacu pada model tampilan *backend* yang merupakan tampilan khusus untuk mengelola segala aktifitas di dalam *web*. Secara umum, aplikasi *backend* ini sering disebut juga sebagai aplikasi administrator. Untuk mengakses model tampilan *backend* pengguna harus melakukan proses login terlebih dahulu. Berikut model perangkat komputer yang digunakan dalam menjalankan aplikasi :

Tabel 3.3 Spesifikasi Perangkat Keras Backend

No	Perangkat Keras	Spesifikasi
1	Monitor	14”
2	*Processor*	Dual-Core E-450
3	Memori / RAM	2 GB
4	*Hardisk*	320GB
5	VGA Card	VGA On-Board
6	*Keyboard*	Standar

b. Perangkat Keras Pendukung Aplikasi *Mobile*

Selain model aplikasi berbasis *web*, perancangan *online service* Distro ToziCozy juga dirancang untuk bisa diakses dalam perangkat berbasis *mobile* dengan model sistem operasi android. Untuk mendukung

pembangunan aplikasi berbasis *mobile* ini dibutuhkan perangkat keras yang dapat mendukung pembangunan aplikasi ini yaitu:

Tabel 3.4 Spesifikasi Perangkat Keras frontend

No	Android	Spesifikasi
1	Sistem Operasi	Android Versi Jelly Bean
2	Display	240 x 320
3	Jaringan	Wi-Fi 802.11 b/g/n (2.4Ghz), HSPA, EDGE Rx (850/900/1800/1900 MHZ), Wi-Fi Direct available,gsm
4	Memori	2 GB
5	Chipset	Single Core Application Processor Single Core Application Processor
6	Baterai	1,200 mAh Battery Capacity
7	Lokasi	Assisted GPS

3.2.5.2 Analisis Kebutuhan Perangkat Lunak

Pada perancangan aplikasi *online service* distro tozicozy ini dibutuhkan perangkat lunak yang dapat membantu dalam pembangunan aplikasi ini. Kebutuhan perangkat lunak yang dibutuhkan adalah sebagai berikut:

Tabel 3.5 Spesifikasi perangkat lunak

No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Windows
2	Aplikasi	IDE *Android Studio*
3	Bahasa Pemograman *web*	PHP (Hyper Text Preproccessor)
4	Bahasa Pemograman *mobile*	Java
5	DBMS	MySQL

6	Browser	Mozila firefox

c. Analisis Pengguna Sistem (User)

Pada perancangan aplikasi *online service* distro tozicozy ini pengguna yang mengakses aplikasi ini merupakan administrator yang merupakan pemilik distro dan konsumen tozicozy sendiri. Dalam menggunakan sistem ini, admin terlebih dahulu harus melakukan *login* untuk autentifikasi data dan pengguna diharuskan memiliki koneksi GRPS/EDGE/UMTS dan fasilitas GPS untuk bisa menggunakan aplikasi ini. Berikut ini karakteristik pengguna sistem yang ditunjukan oleh tabel 3.4 dan 4.5

Tabel 3.6 Karakteristik Pengguna Backend

Pengguna	**Karakteristik yang dibutuhkan**
Pegawai Distro	1. Mengerti dalam menggunakan Komputer 2. Mengerti cara mengoperasikan perangkat lunak *web-based*

Tabel 3.7 Karakteristik Pengguna Aplikasi Frontend

Pengguna	**Karakteristik yang dibutuhkan**
Pelanggan	1. Mampu mengoperasikan Smartphone Android 2. Mengerti menggunakan GPS

3.2.5.3 Analisis Pengkodean

Pada bagian ini akan dibahas tentang pengkodean yang dihasilkan oleh sistem untuk mendefinisikan item.

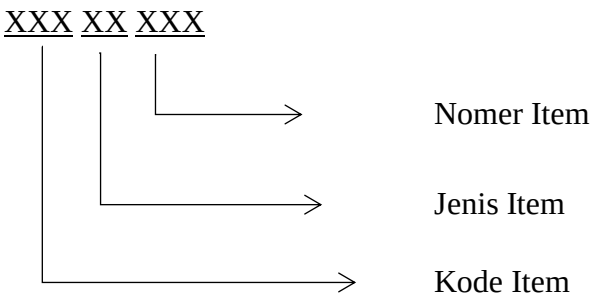
Kode_Item :

Kode item yang diinputkan kedalam sistem berupa inisial singkatan dari item tersebut dan kode ini hanya memiliki 3 digit, Contoh “tozicozy” menjadi “TOS”. Kode item ini hanya sebagai alat bantu ketika user melakukan input item kedalam sistem.

XXX → Format untuk kode pesan berdasarkan inisial item

Kode_jenis item :

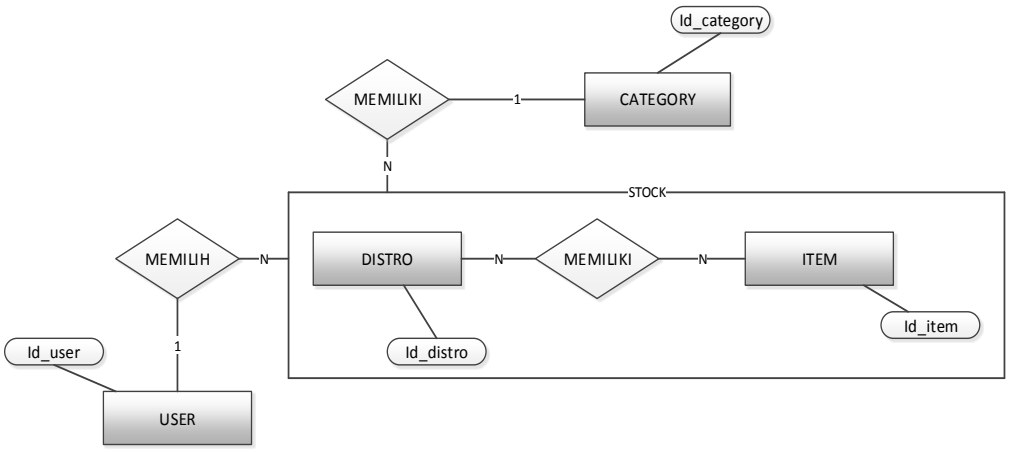
Kode yang dihasilkan oleh sistem memiliki 8 digit yaitu :



Contoh : kode ToziCozy ialah TOZ dan jenis item Tshirt menjadi TS, serta nomer item 001. Maka kode item yang dihasilkan : TOS TS 001.

3.2.6 Analisis Basis Data

Entity Relationship yang digunakan untuk perancangan basis data dalam rancangan sistem web. Dapat dilihat pada gambar 3.4



Gambar 3.5 Entity Relationship Diagram

Tabel 3.8 Tabel Kamus Data ERD

Entitas	Atribut
User	{username,password,}
Distro	{id_distro, nama_distro,alamat, telepon,}
Item	{id_item,nama_item,ukuran,}
Kategori	{jenis_kategori, keterangan}
Stok	{kode_stok,jumlah_stok}

3.2.7 Analisis Kebutuhan Fungsional

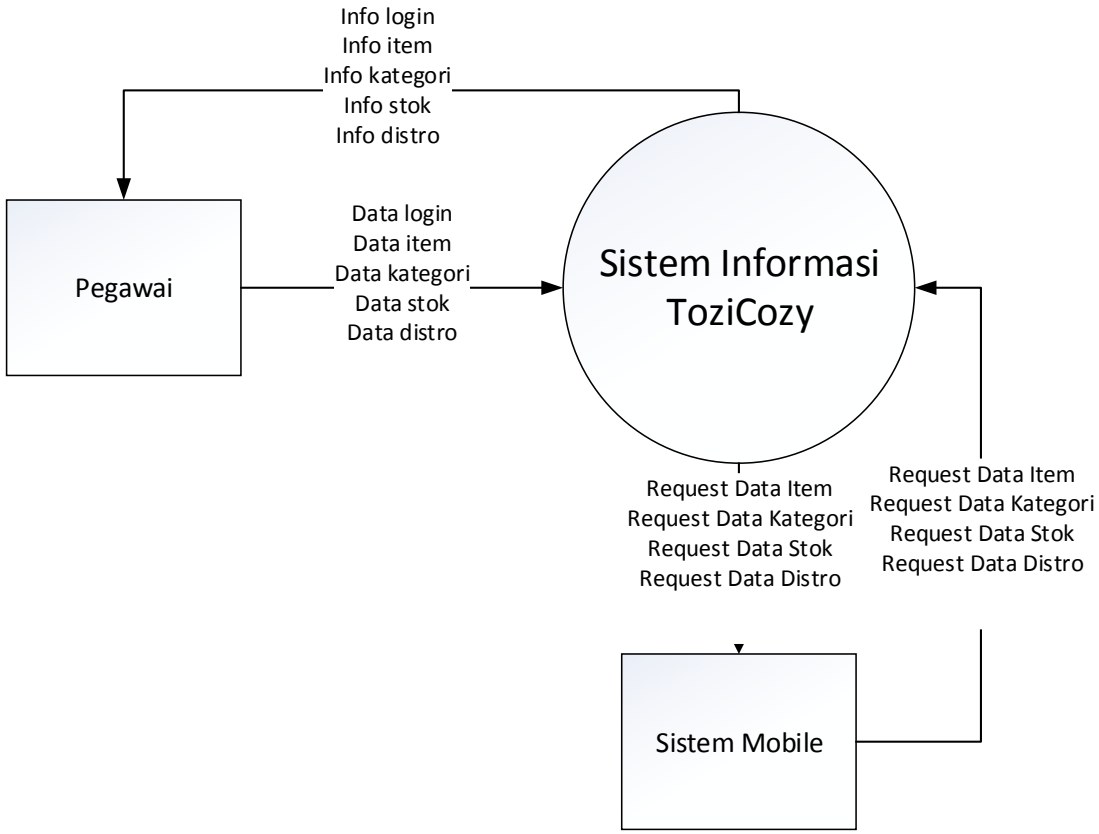
Analisis kebutuhan fungsional sistem yang akan dibahas pada penelitian ini terbagi menjadi dua bagian yaitu Analisis Kebutuhan Fungsional *Web* dan Analisis Kebutuhan *Mobile.*

3.2.8 Analisis Kebutuhan Fungsional Web

Analisis kebutuhan fungsional *web* pada perancangan aplikasi *online service* tozicozy ini meliputi perancangan diagram konteks, data *flow diagram* yang terbagi menjadi beberapa level dan spesifikasi proses yang berfungsi untuk menjelaskan aliran data yang terjadi dalam pembangunan sistem.

1. Diagram Konteks

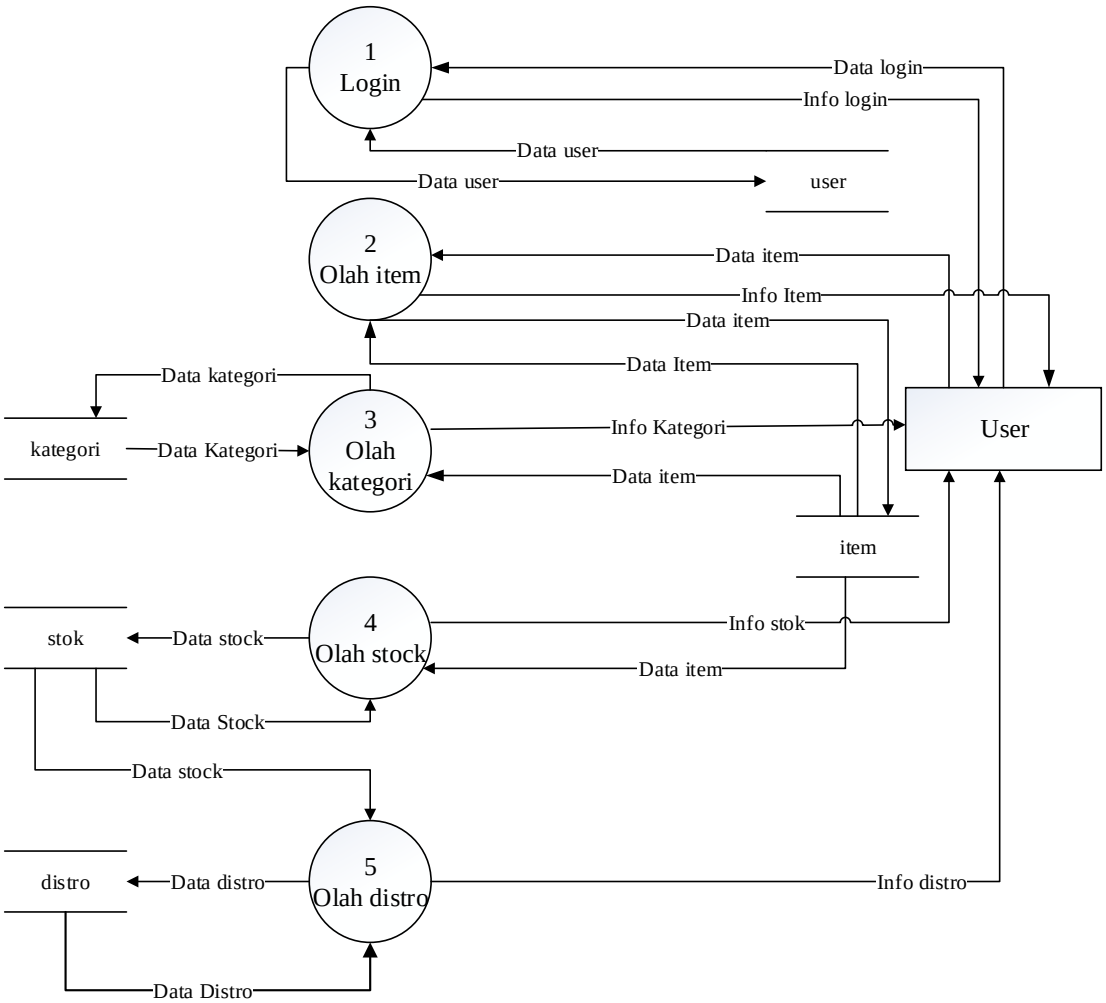
   Diagram Konteks merupakan kejadian tersendiri dari suatu diagram alir data dimana satu lingkaran mempresentasikan seluruh sistem. Diagram konteks merupakan tingkatan tertinggi dalam diagram aliran data dan hanya memuat satu proses, menunjukkan sistem secara keseluruhan. Berikut adalah gambar Diagram Konteks untuk perancangan perangkat lunak yang akan dibangun pada aplikasi *online service* distro tozicozy.



Gambar 3.6 Diagram Konteks

2. Data *Flow Diagram*

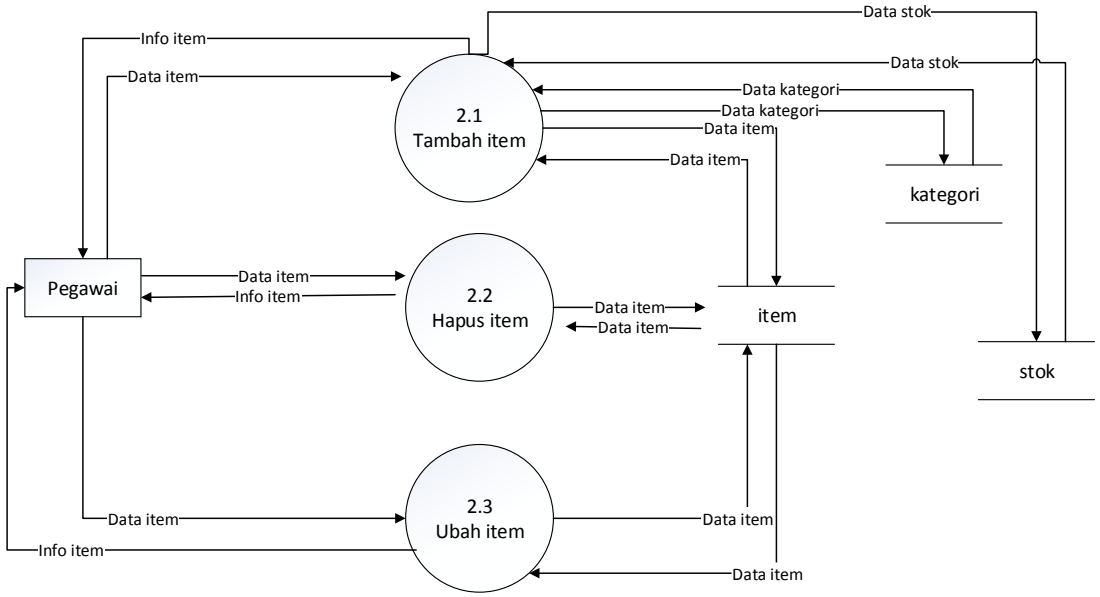
Penyajian desain sistem adalah bentuk data model logikal yang berisi diagram arus data yang mengalir pada sistem. Proses yang terjadi pada model logikal ini berkaitan dengan data, dan sistem yang terkait sehingga mampu membantu dalam menjalankan sistem. Dengan pemodelan sistem ini diharapkan dapat diperoleh gambaran tentang bagaimana sistem ini akan bekerja. Data flow diagram merupakan model logikal yang memberikan gambaran secara umum dari sistem yang akan dibuat. Di dalam proses ini akan mengalir aliran-aliran data dari entitas luar menjadi informasi yang dibutuhkan. Berikut adalah gambaran data *flow diagram* level 1 dari perancangan aplikasi *online service* distro Tozicozy.



Gambar 3.7 DFD Level 0

3. DFD Level 2 proses 2 Pengolahan Item

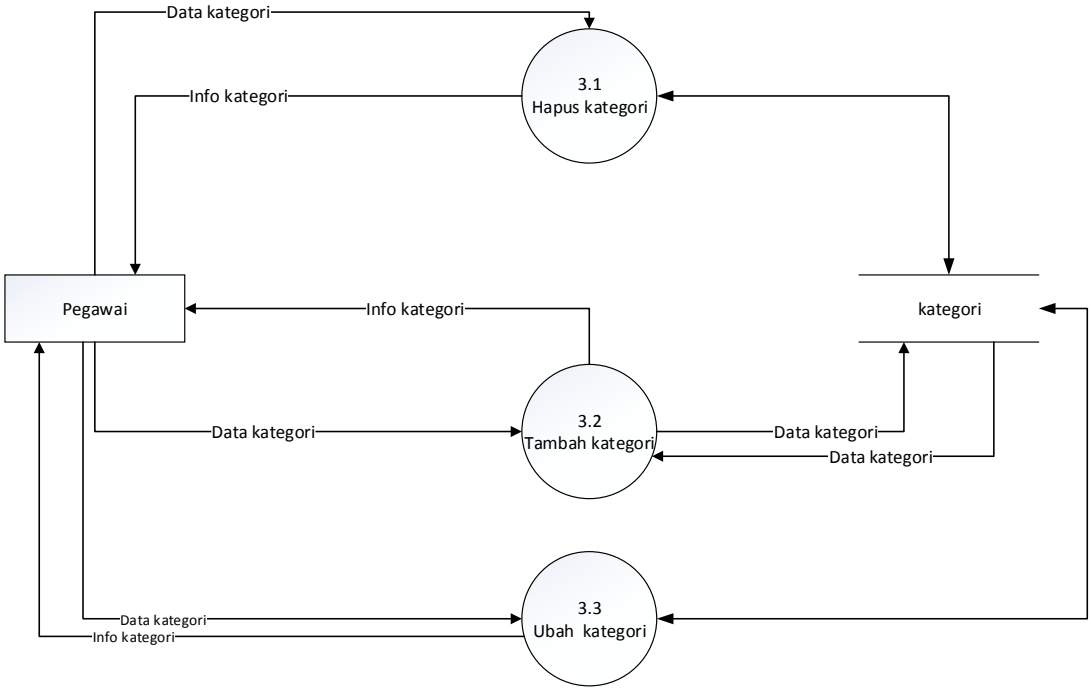
   Pada Gambar 3.7 dijelaskan penggambaran data *flow diagram* Level 2 proses 2 dalam proses pengolahan item pada perangkat lunak yang akan dibangun. Data *flow diagram* level 2 merupakan hasil dekomposisi dari proses-proses yang ada di data *flow diagram level 1*.



Gambar 3.8 DFD Level 1

4. DFD Level 2 proses 3 Pengolahan Kategori

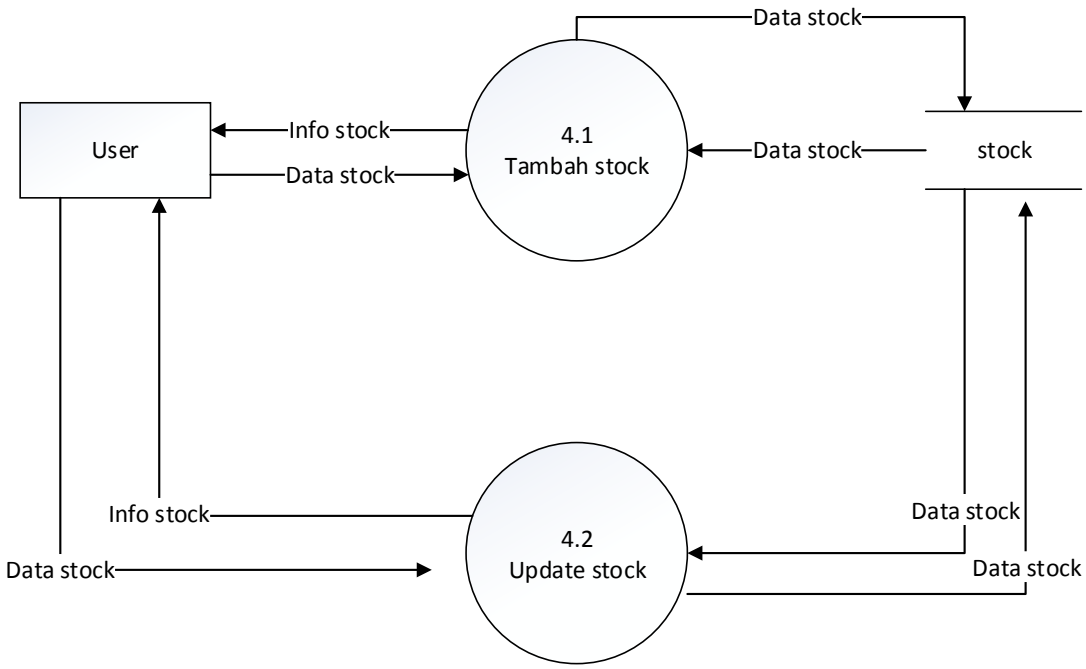
   Berikut adalah gambaran data flow diagram dalam proses pengolahan kategori yang akan dibangun dijelaskan pada Gambar 3.8. Pada proses data flow diagram ini terdapat 2 proses pengolahan data berupa hapus dan tambah kategori yang terhubung dengan penyimpanan data kategori dan data item.



Gambar 3.9 DFD Level 3

5. DFD Level 2 proses 4 Pengolahan Stok

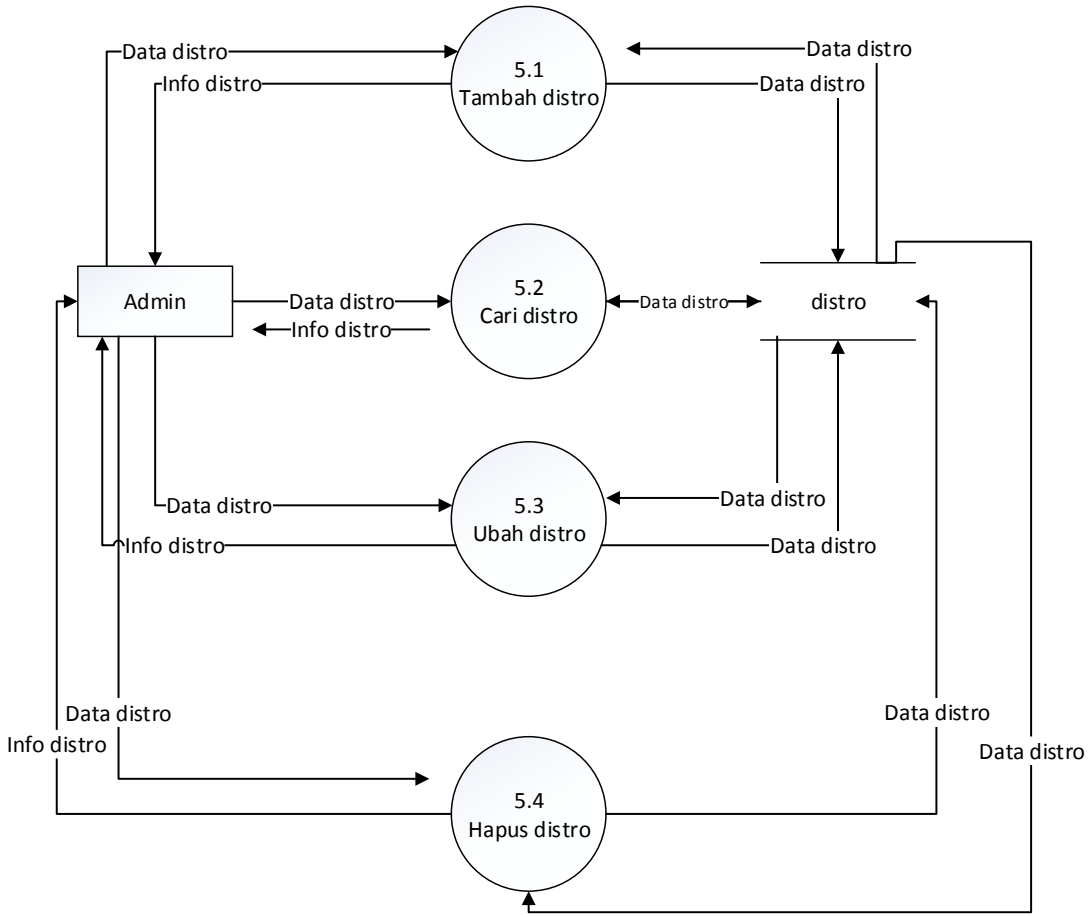
   Berikut adalah gambaran dari proses data *flow diagram* level 2 pengolahan stok pada perangkat lunak yang akan dibangun. Dijelaskan pada gambar 3.9.



Gambar 3.10 DFD Level 2 proses 4

6. DFD Level 2 proses 5 Pengolahan Distro

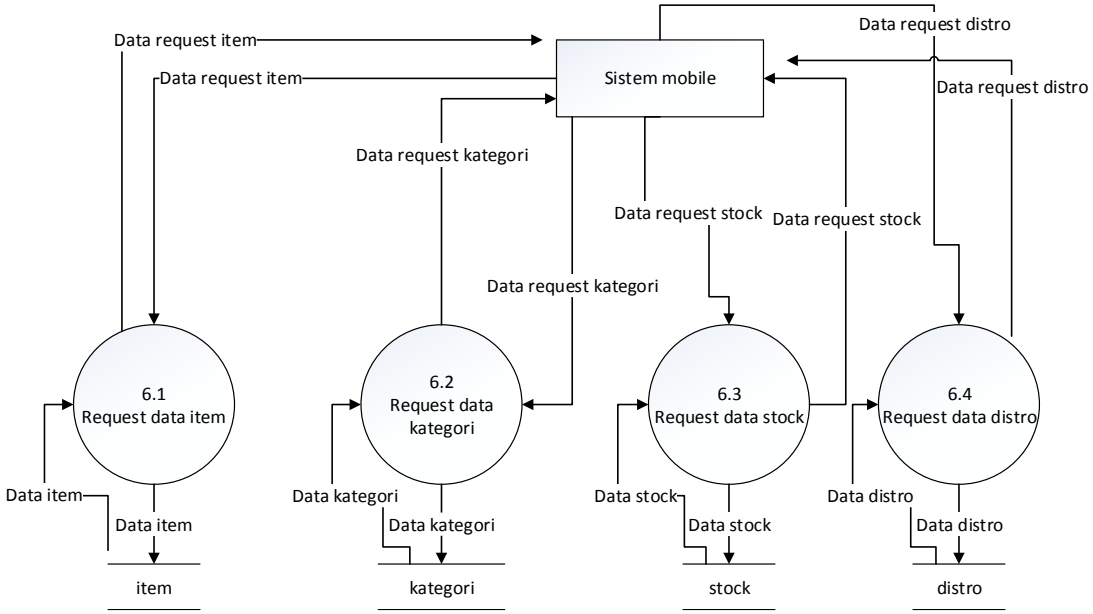
   Berikut adalah gambaran dari proses data *flow diagram* level 2 proses 5 pengolahan distro pada perangkat lunak yang akan dibangun. Dijelaskan pada gambar 3.10.



Gambar 3.11 DFD Level 2 proses 5

7. DFD level 1 Penyajian *web service*

   Berikut adalah Gambar Data *Flow Diagram* level 1 tentang penyajian *web service* dari proses pembangunan sistem dijelaskan pada Gambar 3.11.



Gambar 3.12 DFD Level 1 Penyajian *Web Service*

1. Spesifikasi Proses *Level* 1 *Login*

   Pada spesifikasi proses *Level* 1 *Login* digunakan untuk menggambarkan proses *login* yang terdapat pada DFD *Level 1 Login*. Berikut tabel yang menjelaskan spesifikasi dari alur data diagram yang telah digambarkan pada gambar 3.9

Tabel 3.9 Spesifikasi Proses Level 1 Login

No	Proses	Keterangan
1.	No Proses	1.1
	Nama Proses	*Login*
	Source (Sumber)	Admin
	Input	Data *Login*
	Output	Info *Login*
	Destination	User
	Logika Proses	1. User mengisi form dengan data *login* User

No	Proses	Keterangan
		2. Sistem akan memeriksa data *login* user yang dimasukan 3. Jika data *login* benar maka akan menampilkan halaman utama admin 4. Jika data *login* salah maka akan menampilkan pesan "*user*name atau *password* salah"

2. Spesifikasi Proses *Level* 2 Pengolahan *Item*

   Pada spesifikasi proses *Level* 2 Pengolahan *Item* digunakan untuk menggambarkan proses pengolahan *item* yang terdapat pada *DFD Level* 2 Pengolahan *Item*. Berikut tabel yang menjelaskan spesifikasi dari alur data diagram yang telah digambarkan pada gambar 3.10

Tabel 3.10 Spesifikasi Proses Level 2 Pengolahan Item

No	Proses	Keterangan
1.	No Proses	2.1
	Nama Proses	Tambah *Item*
	Source (Sumber)	User
	Input	Data *Item*
	Output	Info *Item*
	Destination	User
	Logika Proses	1. User memilih tambah *item*. 2. Sistem menampilkan *form* tambah *item*. 3. User memasukan data *item* sesuai *form.*

No	Proses	Keterangan
		4. Sistem melakukan validasi pada data masukan, jika data yang dimasukan kurang lengkap maka user diharuskan mengisi kembali. 5. Sistem menyimpan masukan dalam *database*.
2.	No Proses	2.2
	Nama Proses	Hapus *Item*
	Source (Sumber)	User
	Input	Data *Item*
	Output	Info *Item*
	Destination	User
	Logika Proses	1. User memilih hapus *item* 2. Sistem menampilkan konfirmasi penghapusan data item dari *database* 3. Jika user memilih ya maka sistem akan menghapus data item dari *database* 4. Jika user memilih tidak maka sistem akan membatalkan proses penghapusan data.

3. Spesifikasi Proses Level 2 Pengolahan Kategori

   Pada spesifikasi proses Level 2 Pengolahan kategori digunakan untuk menggambarkan proses pengolahan kategori yang terdapat pada DFD Level 2 pengolahan kategori. Berikut tabel yang menjelaskan spesifikasi dari alur data diagram yang telah digambarkan pada gambar 3.11

Tabel 3.11 Spesifikasi Proses Level 2 Pengolahan Kategori

No	Proses	Keterangan
1.	No Proses	3.1
	Nama Proses	Hapus Kategori
	Source (Sumber)	User
	Input	Data Kategori
	Output	Info Kategori
	Destination	User
	Logika Proses	1 User memilih hapus kategori 2 Sistem menampilkan konfirmasi penghapusan data kategori dari *database* 3 Jika user memilih ya maka sistem akan menghapus data kategori dari *database* 4 Jika user memilih tidak maka sistem akan membatalkan proses penghapusan data.
2.	No Proses	3.2
	Nama Proses	Tambah Kategori
	Source (Sumber)	User
	Input	Data Kategori
	Output	Info Kategori
	Destination	User

Logika Proses	1. Pegawai memilih tambah kategori

	2. Sistem menampilkan konfirmasi tambah data kategori dari *database* 3. Jika pegawai memilih ya maka sistem akan menambah data kategori.
No Proses	3.3
Nama Proses	Ubah Kategori
Source (Sumber)	Pegawai
Input	Data Kategori
Output	Info Kategori
Destination	Pegawai
Logika Proses	1. Pegawai memilih ubah kategori 2. Sistem menampilkan konfirmasi ubah data kategori dari *database* 3. Jika pegawai memilih ya maka sistem akan mengubah data kategori.

4. Spesifikasi Proses Level 2 Pengolahan Stok

   Pada spesifikasi proses Level 2 Pengolahan stok digunakan untuk menggambarkan proses pengolahan stok yang terdapat pada DFD Level 2 pengolahan stok. Berikut tabel yang menjelaskan spesifikasi dari alur data diagram yang telah digambarkan pada gambar 3.12

Tabel 3.12 Spesifikasi Proses Level 2 Pengolahan Stok

No	Proses	Keterangan
1.	No Proses	4.1
	Nama Proses	Tambah Stok
	Source (Sumber)	User
	Input	Data Stok
	Output	Info Stok
	Destination	User
	Logika Proses	1. User memilih tambah Stok. 2. Sistem menampilkan form tambah Stok. 3. Pengguna memasukan nama item yang akan ditambahkan jumlah stoknya. 4. Sistem mencari data item dari db_item sesuai nama item yang dicari admin. 5. Sistem menampilkan data item pada form. 6. Pegawai memasukan jumlah stok pada *form* tambah stok. 7. Sistem melakukan validasi pada data masukan, jika data yang dimasukan kurang lengkap maka pegawai diharuskan mengisi kembali. 8. Sistem menyimpan masukan dalam *database.*
2.	No Proses	4.2

No	Proses	Keterangan
	Nama Proses	Update Stok
	Source (Sumber)	User
	Input	Data Stok
	Output	Info Stok
	Destination	User
	Logika Proses	1. User memilih *Update* Stok. 2. Sistem menampilkan form update stok 3. Pengguna memasukan nama item yang akan di *update* stoknya. 4. Sistem mencari nama item pada db_item jika nama item yang dimasukan tidak ada maka admin diharuskan untuk menambah stok item terlebih dahulu. 5. Sistem menampilkan data stok. 6. Pegawai menambahkan jumlah stok item pada *field* jumlah stok. 7. Sistem melakukan validasi pada data masukan, jika data yang dimasukan kurang lengkap maka admin diharuskan mengisi kembali. 8. Sistem menyimpan data masukan pada database.

5. Spesifikasi Proses *Level 2* Pengolahan Distro

Pada spesifikasi proses Level 2 Pengolahan distro digunakan untuk menggambarkan proses pengolahan distro yang terdapat pada *DFD Level* 2 pengolahan distro. Berikut tabel yang menjelaskan spesifikasi dari alur data diagram yang telah digambarkan pada gambar 3.13

Tabel 3.13 Spesifikasi Proses Level 2 Pengolahan Distro

No	Proses	Keterangan
1.	No Proses	5.1
	Nama Proses	Tambah Distro
	Source (Sumber)	Admin
	Input	Data Distro
	Output	Info Distro
	Destination	Admin
	Logika Proses	1 Admin memilih tambah distro. 2 Sistem menampilkan form tambah distro. 3 Admin memasukan data distro yang akan ditambahkan. 4 Sistem mencari data stock dari db_stock untuk ditambahkan pada data distro 6. Sistem menampilkan data stock pada form. 7. Admin memasukan jumlah stock pada form data distro. 8. Sistem melakukan validasi pada data masukan, jika data yang dimasukan kurang

No	Proses	Keterangan
		lengkap maka admin diharuskan mengisi kembali. 9. Sistem menyimpan masukan dalam *database*.
2.	No Proses	5.2
	Nama Proses	Cari Distro
	Source (Sumber)	Admin
	Input	Data Distro
	Output	Info Distro
	Destination	Admin
	Logika Proses	1. Admin memilih cari distro. 2. Admin memasukan nama distro. 3. Sistem mencari distro sesuai yang dicari oleh admin pada *database*. 4. Jika data tidak ada maka sistem akan menampilkan tabel kosong. 5. Jika data ada, sistem menampilkan informasi item yang dicari dari *database*
3.	No Proses	5.4
	Nama Proses	Hapus Distro
	Source (Sumber)	Admin
	Input	Data Distro
	Output	Info Distro
	Destination	Admin

No	Proses	Keterangan
	Logika Proses	1. Admin memilih hapus distro. 2. Sistem menampilkan konfirmasi penghapusan data distro dari *database* 3. Jika admin memilih ya maka sistem akan menghapus data distro dari *database* 4. Jika admin memilih tidak maka sistem akan membatalkan proses penghapusan data.

6. Spesifikasi Proses *Level* 1 Pengolahan *Web Service*

Pada spesifikasi proses *Level* 1 Pengolahan web service digunakan untuk menggambarkan proses model aliran *web service* yang terdapat pada DFD *Level* 1 pengolahan *web service*. Berikut tabel yang menjelaskan spesifikasi dari alur data diagram yang telah digambarkan pada gambar 3.14

Tabel 3.14 Spesifikasi Proses Level 1 Pengolahan Web Service

No	Proses	Keterangan
1.	No Proses	6.1
	Nama Proses	*Request* Data Item
	Source (Sumber)	Sistem *Mobile*
	Input	Data *Request* Item
	Output	Data *Request* Item
	Destination	Sistem *Mobile*
	Logika Proses	1. Sub Sistem *Web* melakukan *request* data item. 2. Sistem memvalidasi *request* 3. Jika *request* valid sistem akan mengirim data *request* item

No	Proses	Keterangan
		4. Jika *request* tidak valid sistem akan mengirim hasil null
2.	No Proses	6.2
	Nama Proses	*Request* Data Kategori
	Source (Sumber)	Sistem *Mobile*
	Input	Data *Request* Kategori
	Output	Data *Request* Kategori
	Destination	Sistem *Mobile*
	Logika Proses	1. Sub Sistem *Web* melakukan *request* data kategori 2. Sistem memvalidasi *request* 3. Jika *request* valid sistem akan mengirim data *request* kategori 4. Jika *request* tidak valid sistem akan mengirim hasil null
3.	No Proses	6.3
	Nama Proses	*Request* Data Stock
	Source (Sumber)	Sistem *Mobile*
	Input	Data *Request* Stock
	Output	Data *Request* Stock
	Destination	Sistem *Mobile*
	Logika Proses	1. Sub Sistem *Web* melakukan *request* data stock 2. Sistem memvalidasi *request*. 3. Jika *request* valid sistem akan mengirim data *request* stock.

No	Proses	Keterangan
		4. Jika *request* tidak valid sistem akan mengirim hasil null.
4.	No Proses	6.4
	Nama Proses	*Request* Data Distro
	Source (Sumber)	Sistem *Mobile*
	Input	Data *Request* distro
	Output	Data *Request* distro
	Destination	Sistem *Mobile*
	Logika Proses	1. Sub Sistem *Web* melakukan *request* data distro. 2. Sistem memvalidasi *request*. 3. Jika *request* valid sistem akan mengirim data *request* distro. 4. Jika *request* tidak valid sistem akan mengirim hasil null.

7. Kamus Data

Berikut adalah yang mendeskripsikan kamus data yang digunakan pada pembangunan perangkat lunak.

Tabel 3.15 Kamus Data

Nama Aliran Data	data user
How Used	1. Proses Login
Keterangan	validasi menggunakan username dan password
Struktur data	Username, password
Username Password	huruf [a..z,A..Z] kombinasi [a..z,A..Z,0..9]

Nama Aliran Data	data item
How Used	1. Proses olah item
Keterangan	Create,read,update,delete data item
Struktur data	id_item,nama_item,ukuran
id_item nama_item ukuran	Angka [0..9] huruf [a..z,A..Z] kombinasi [a..z,A..Z,0..9]

Nama Aliran Data	data kategori
How Used	1. Proses olah kategori
Keterangan	Create,read,update,delete data kategori
Struktur data	jenis_kategori, keterangan
jenis_kategori keterangan	kombinasi [a..z,A..Z,0..9] kombinasi [a..z,A..Z,0..9]

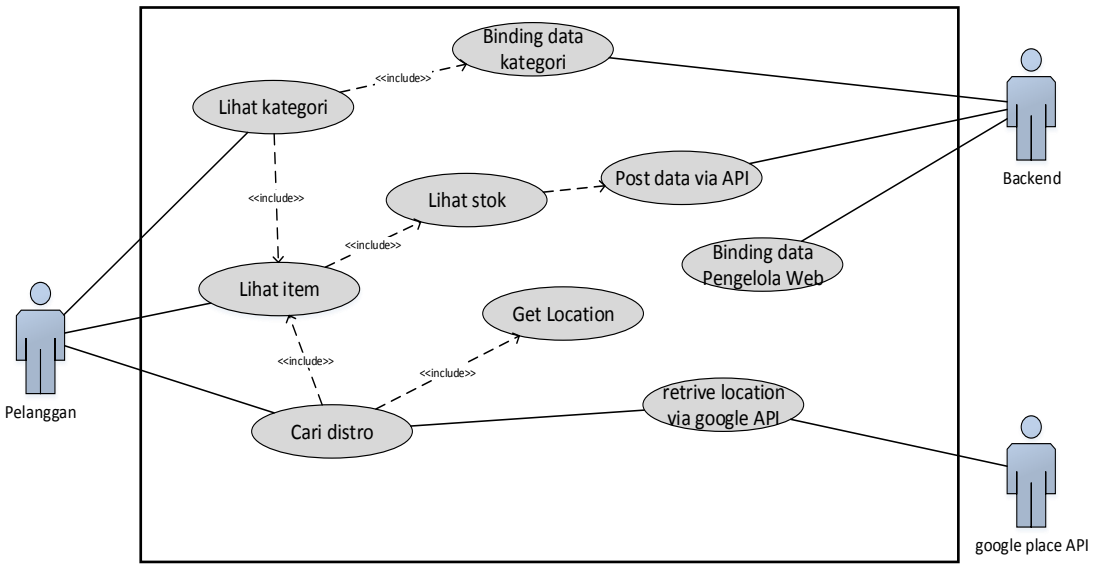
Nama Aliran Data	data distro
How Used	1. Proses olah distro
Keterangan	Create,read,update,delete data distro
Struktur data	id_distro, nama_distro,alamat, telepon
id_distro nama_distro alamat telepon	Angka [0..9] huruf [a..z,A..Z] kombinasi [a..z,A..Z,0..9] Angka [0..9]

Nama Aliran Data	data stok
How Used	1. Proses Olah Stok
Keterangan	validasi menggunakan username dan password
Struktur data	kode_stok,jumlah_stok
kode_stok jumlah_stok	Angka [0..9] Angka [0..9]

3.2.8.1 Analisis Kebutuhan *Mobile*

a. Diagram *Use Case*

Diagram Use case merupakan bagian tertinggi dari fungsionalitas yang dimiliki sistem yang akan menggambarkan bagaimana seseorang atau aktor akan menggunakan dan memanfaatkan sistem. Diagram ini juga medeskripsikan apa yang akan dilakukan oleh sistem. *UseCase* terdiri dari tiga bagian yaitu identifikasi aktor, identifikasi *UseCase*, dan skenario *UseCase*.



Gambar 3.13 Use Case Diagram Sistem

1. Identifikasi Aktor

   Identifikasi aktor dimaksudkan untuk mengetahui siapa saja aktor yang terlibat di dalam sistem aplikasi ini. Deskripsi dari setiap aktor yang terlibat dapat dilihat pada tabel 3.16.

Tabel 3.16 Identifikasi Aktor

NO	Aktor	Deskripsi
1	Pelanggan	Aktor ini merupakan pengguna aplikasi pada platform Android. Dimana dalam menggunakan aplikasi aktor harus memiliki koneksi internet yang berguna untuk parsing data ke database server.
2	*Backend*	Aktor ini hanya memiliki wewenang sebagai penyedia jembatan sumber data.

NO	Aktor	Deskripsi
3	Google place API	Aktor ini memiliki wewenang sebagai sumber data lokasi yang digunakan aplikasi dan berjalan secara background.

1. Identifikasi *Use Case*

   Identifikasi use case yang terdapat dalam sistem dapat dilihat pada tabel 3.17.

Tabel 3.17 Identifikasi Use case

No	*UseCase*	Deskripsi
1	Lihat Kategori	Fungsionalitas ini untuk menampilkan data kategori yang diurutkan berdasarkan jenis.
2	Pilih item	Fungsionalitas ini digunakan untuk memilih item (pakaian) yang di butuhkan.
3	Cari Distro	Fungsionalitas ini digunakan untuk mencari lokasi distro
4	Get Location	Sistem memanggil fungsi *get location* untuk mengetahui lokasi pengguna
5	Binding Data kategori	Fungsionalitas ini digunakan untuk menerima data kategori dari server
6	Lihat Stok	Fungsionalitas ini digunakan untuk menampilkan informasi stok item
7	Post data via API	Fungsionalitas ini digunakan untuk mengirimkan data via API
8	Binding Data pengguna web	Fungsionalitas ini digunakan untuk mengelola data pengguna web
9	*Retrieve Location Data via API*	*Google Maps* menyediakan data lokasi dari api *database Google Maps*

2. Skenario *Use Case*

   Skenario dari proses-proses yang terdapat didalam *Use Case* dapat dilihat pada tabel 3.18.

Tabel 3.18 Skenario UseCase Lihat Kategori

Identifikasi	
Nama *Use Case*	Lihat Kategori
Aktor	Pelanggan
Tujuan	Agar memudahkan pelanggan dalam memilih kategori
Keadaaan Awal	Pelanggan berada pada home awal aplikasi
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Pelanggan menekan tombol pilih kategori	2. Sistem menampilkan data kategori berdasarkan jenis item

Tabel 3.19 Skenario UseCase Pilih Item

Identifikasi	
Nama *Use Case*	Pilih item
Aktor	Pelanggan
Tujuan	Menampilkan detail item yang dipilih
Keadaan Awal	Pelanggan berada pada home awal aplikasi
Skenario Utama	
Aksi Aktor	Reaksi Sistem

1. Pelanggan memilih item	2. Menampilkan detail item yang dipilih pelanggan.

Tabel 3.20 Skenario UseCase Lihat Stok

Identifikasi	
Nama *Use Case*	Lihat Stok
Aktor	Pelanggan
Tujuan	Pelanggan dapat melihat informasi detail item yang di cari termasuk ketersedian stok dan distro yang menyediakan
Keadaan Awal	Pelanggan berada pada tampilan info item yang dicari
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Pelanggan memilih item yang dicari.	2. Menampilkan informasi stok item

Tabel 3.21 Skenario UseCase Cari Distro

Identifikasi	
Nama *Use Case*	Cari Distro
Aktor	Pelanggan
Tujuan	Pelanggan dapat melakukan pencarian informasi alamat distro
Keadaaan Awal	Pelanggan berada pada menu utama
Skenario Utama	
Aksi Aktor	Reaksi Sistem

1. Pelanggan memilih menu cari distro pada menu utama	2. Sistem menampilkan informasi alamat distro

Tabel 3.22 Skenario UseCase Get Location

Identifikasi	
Nama *Use Case*	*Get location*
Aktor	Pelanggan
Tujuan	Untuk mengetahui lokasi distro
Keadaaan Awal	Sistem terkoneksi dengan *server*
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Pengguna mengirimkan *request* untuk megetahui lokasi distro	2. Sistem menerima data lokasi dari *google* 3. Menampilkan informasi lokasi distro

Tabel 3.23 Skenario UseCase Binding Data Kategori

Identifikasi	
Nama *Use Case*	Binding data Kategori
Aktor	User
Tujuan	Untuk menampilkan kategori item
Keadaaan Awal	User telah melakukan login
Skenario Utama	
Aksi Aktor	Reaksi Sistem

1. Melakukan input data setiap kategori	2. Menyimpan data hasil input 3. Menampilkan data hasil input

Tabel 3.24 Skenario UseCase Post data via API

Identifikasi	
Nama *Use Case*	*Post data via API*
Aktor	*User*
Tujuan	Untuk mengirimkan data stok ke server
Keadaaan Awal	User telah melakukan login
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Mengirimkan data stok	2. Menerima data stok 3. Menampilkan data stok

Tabel 3.25 Skenario UseCase Retrieve Location Data API

Identifikasi	
Nama *Use Case*	Retrieve Location Data via API
Aktor	Google Maps API
Tujuan	Untuk memberikan data lokasi daro google maps
Keadaaan Awal	*Server google maps* terkoneksi dengan sistem
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Sistem mengirimkan *request* data lokasi	2. *Google maps* memeriksa *request* sistem 3. *Google maps* mengirimkan data lokasi

4. Sistem menerima data lokasi dari *google maps*	

Tabel 3.26 Skenario UseCase Binding Data Pengelola Web

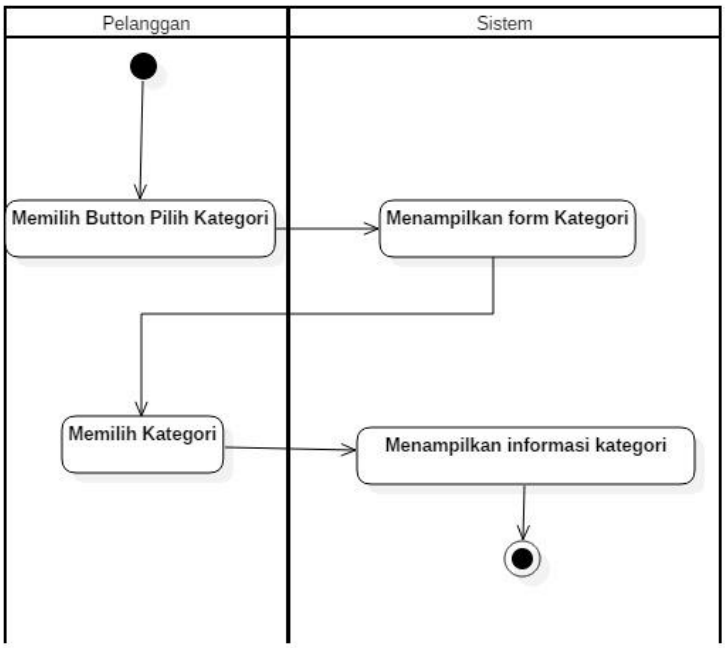
Identifikasi	
Nama *Use Case*	Binding data pengelola web
Aktor	Web
Tujuan	Untuk mengelola data web
Keadaaan Awal	Admin sudah melakukan login
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Melakukan pengelolaan data web	2. Menyimpan hasil pengelolaan data web

3. Diagram *Activity*

Activity diagram menjelaskan aliran kerja aktor pada perangkat lunak yang akan di bangun.

a. *Activity* Diagram *Pilih Kategori*

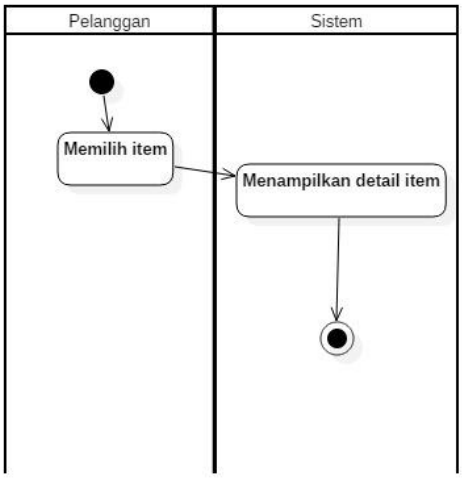
Berikut adalah Gambar 3.14 Activity Diagram pilih Kategori yang menggambarkan activity diagram pilih Kategori.



Gambar 3.14 Activity Diagram Pilih Kategori

b. *Activity* Diagram Pilih Item

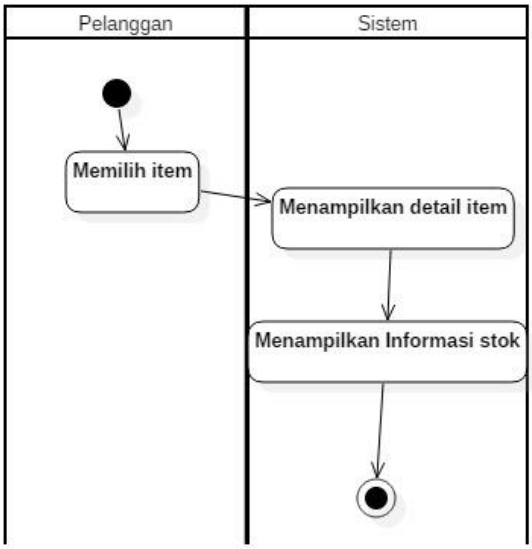
Berikut adalah Gambar 3.14 Activity Diagram pilih item yang menggambarkan activity diagram pilih item.



Gambar 3.15 Activity Diagram Pilih Item

c. *Activity* Diagram *Lihat Stok*

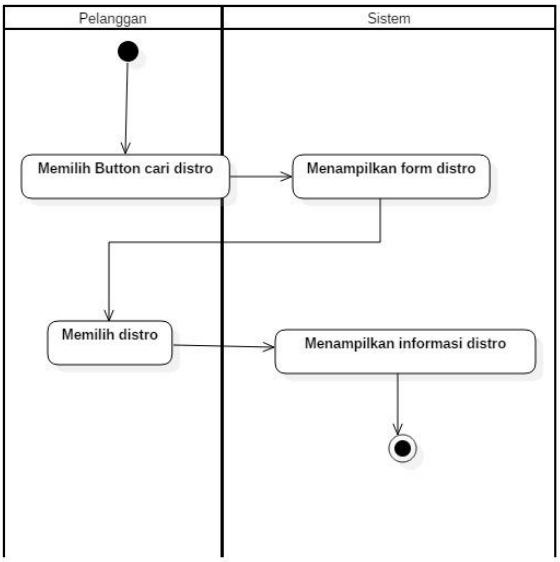
Berikut adalah Gambar 3.15 Activity Diagram lihat stok yang menggambarkan activity diagram lihat stok.



Gambar 3.16 Activity Diagram Lihat Stok

d. *Activity* Diagram *Cari Distro*

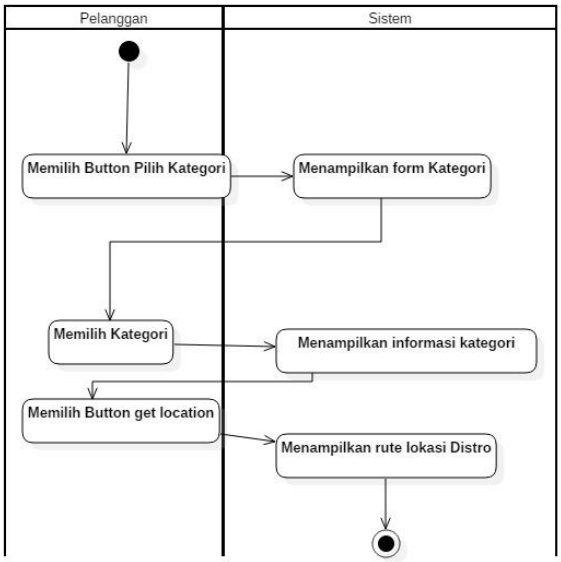
Berikut adalah Gambar 3.16 Activity Diagram cari distro yang menggambarkan activity diagram cari distro.



Gambar 3.17 Activity Diagram Cari Distro

e. *Activity* Diagram *Get Location*

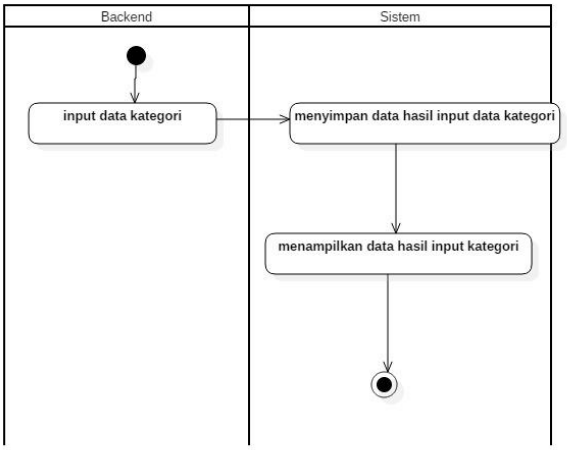
Berikut adalah Gambar 3.17 Activity Diagram get location yang menggambarkan activity diagram get location.



Gambar 3.18 Activity Diagram Get Location

f. *Activity* Diagram Binding Data Kategori

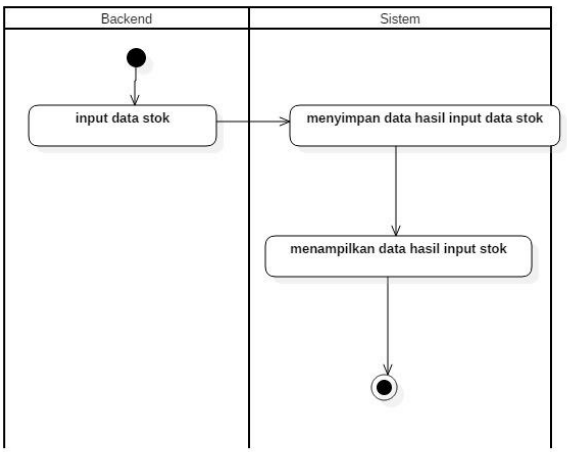
Berikut adalah Gambar 3.18 Activity Diagram binding data kategori yang menggambarkan activity diagram binding data kategori.



Gambar 3.19 Activity Binding Data Kategori

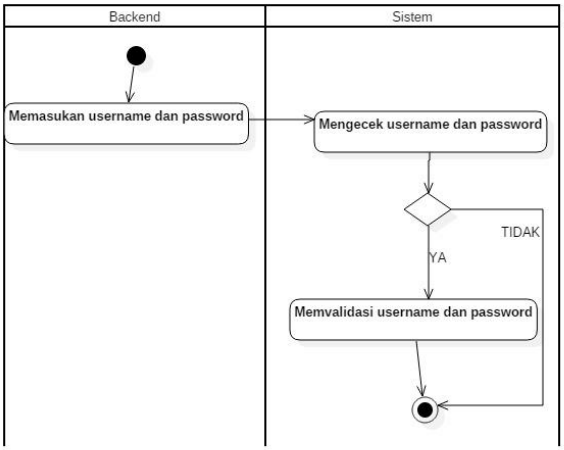
g. *Activity* Diagram *Post Data via API*

Berikut adalah Gambar 3.19 Activity Diagram Post Data via API yang menggambarkan activity diagram Post data via API.



Gambar 3.20 Activity Diagram Post Data via API

a. *Activity* Diagram *Login*



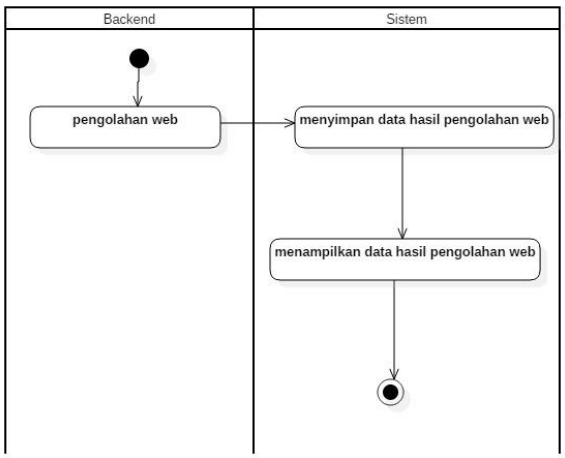
.

Gambar 3.21 Activity Diagram Login

Berikut adalah Gambar 3.20 Activity Diagram *Login* yang menggambarkan activity diagram *Login*.

a. *Activity* Diagram *Binding* Data Pengelola *Web*

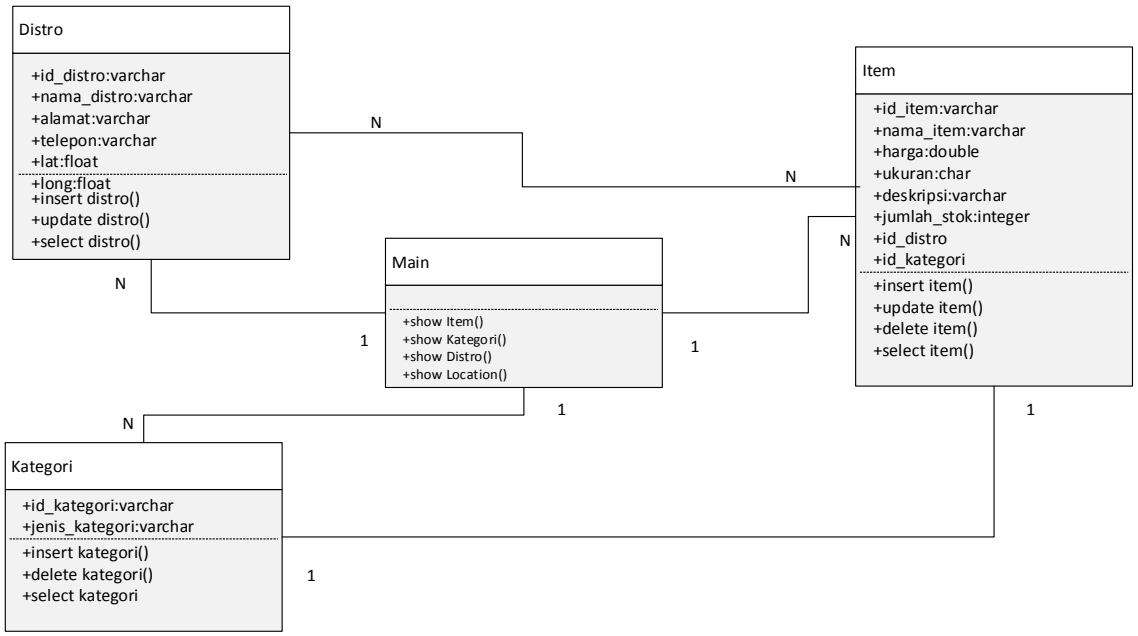
Berikut adalah Gambar 3.21 Activity Diagram Binding data pengelola *Web* yang menggambarkan activity diagram Binding Data pengelola *Web*.



Gambar 3.22 Activity Diagram Binding Data Pengelola Web

4. Diagram *Class*

Berikut adalah Gambar 3.22 Diagram Class yang menggambarkan kelas-kelas dalam sebuah sistem dan hubungannya antara satu dengan yang lain.



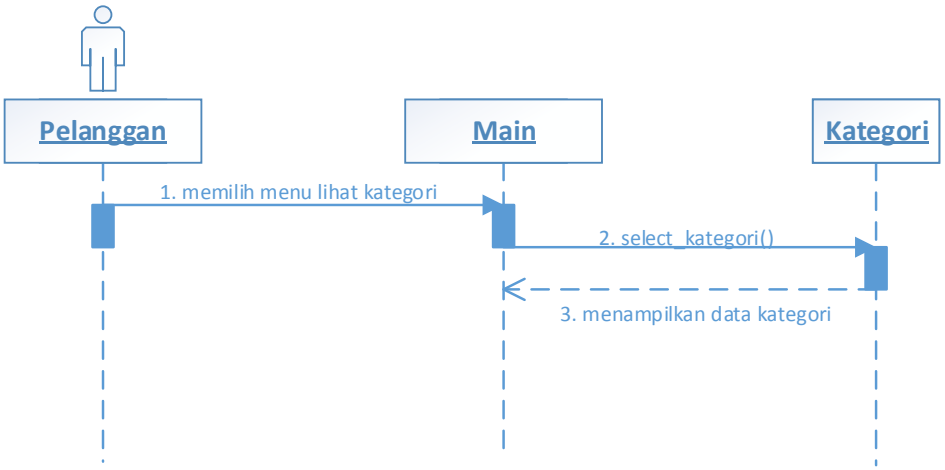
Gambar 3.23 Class Diagram

5. Diagram *Sequence*

Sequence diagram digunakan untuk menggambarkan arus pekerjaan, pesan yang di sampaikan dan bagaimana elemen-elemen di dalamnya bekerja sama dari waktu ke waktu untuk mencapai suatu hasil.

a. Sequence Diagram Lihat kategori

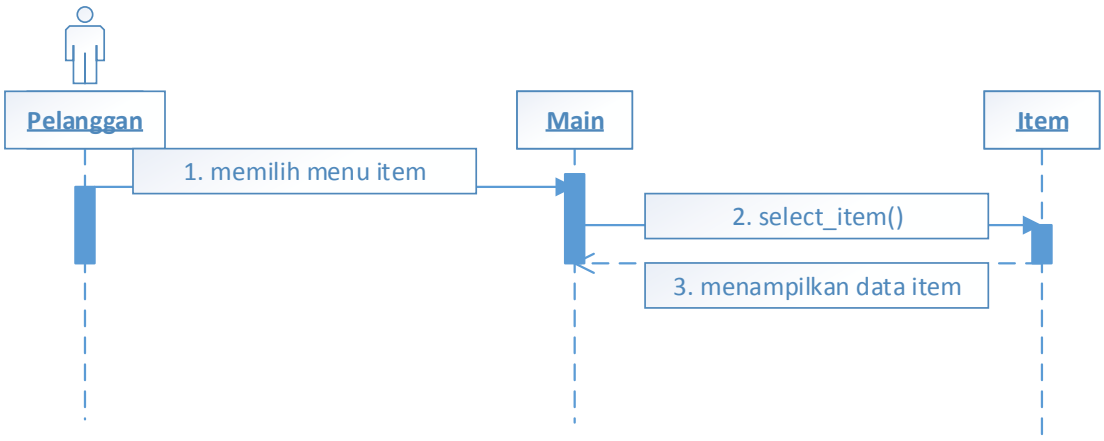
Berikut Gambar 3.23 Sequence Diagram yang menggambarkan sequence diagram Lihat kategori.



Gambar 3.24 Sequence Diagram Lihat Kategori

b. Sequence Diagram Lihat Item

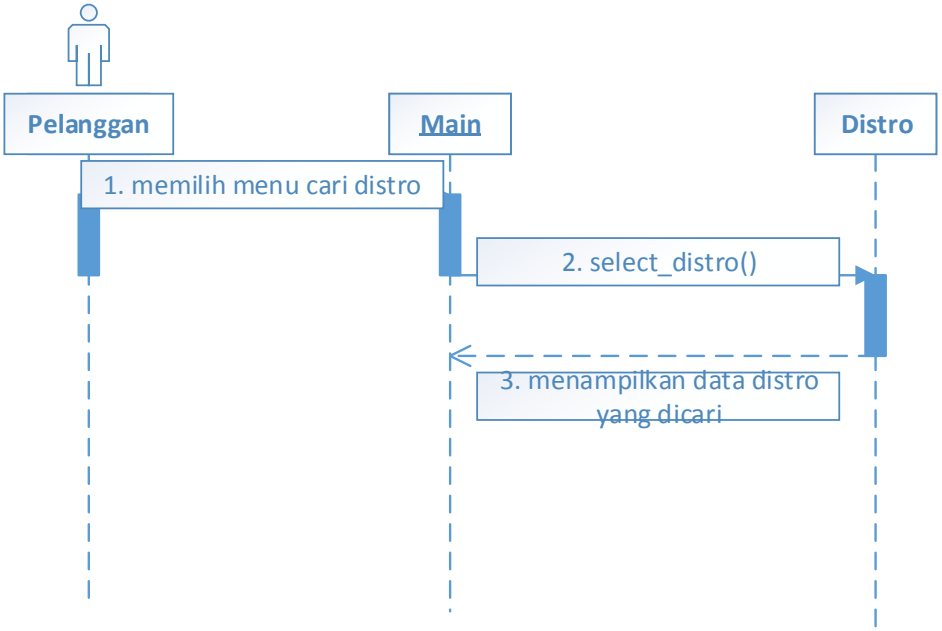
Berikut Gambar 3.24 Sequence Diagram yang menggambarkan sequence diagram Lihat item.



Gambar 3.25 Sequence Diagram Lihat Item

c. Sequence Diagram Distro

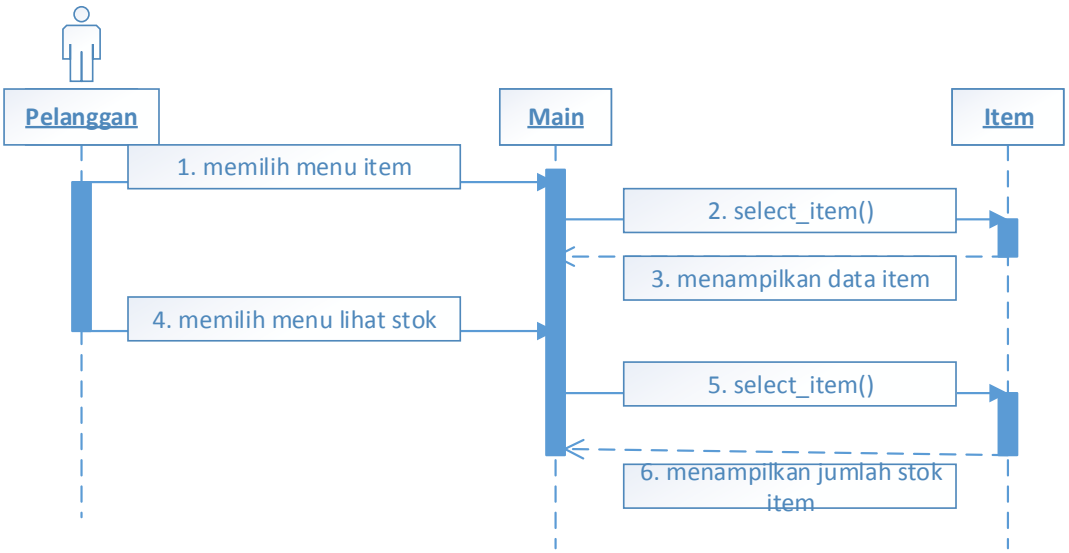
Berikut Gambar 3.25 Sequence Diagram yang menggambarkan sequence diagram Distro.



Gambar 3.26 Sequence Diagram Distro

d. Sequence Diagram Stok

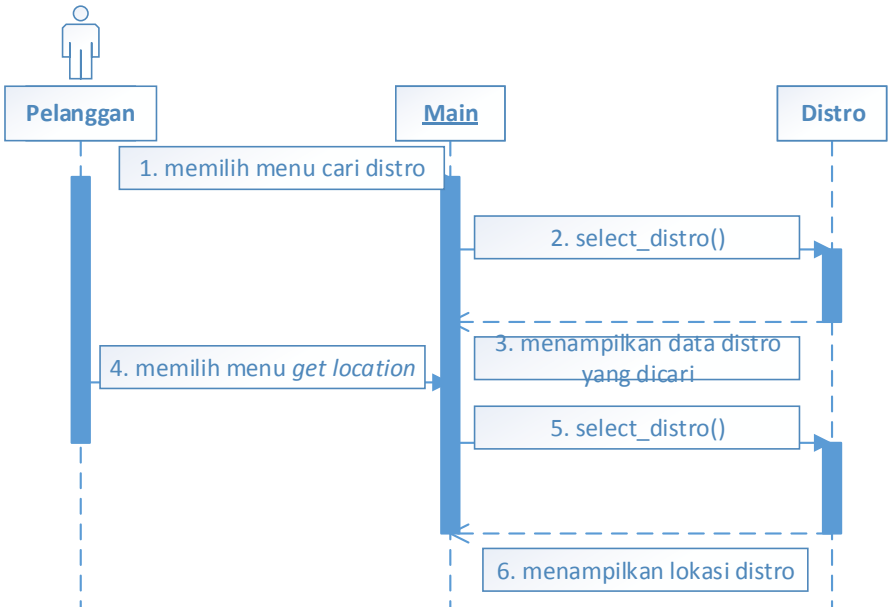
Berikut Gambar 3.26 Sequence Diagram yang menggambarkan sequence diagram Stok.



Gambar 3.27 Sequence Diagram Stok

e. Sequence Diagram Get Location

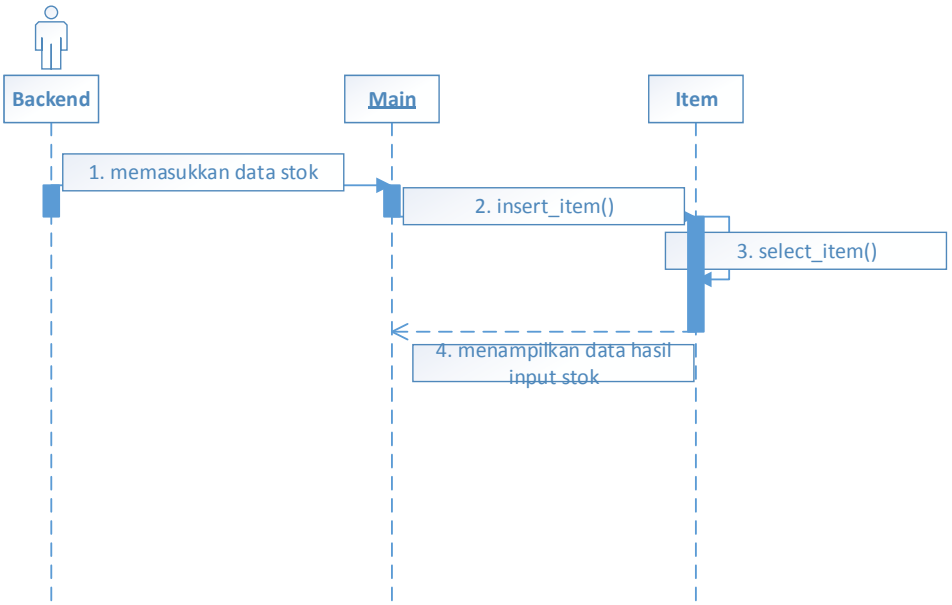
Berikut Gambar 3.27 Sequence Diagram yang menggambarkan sequence diagram get location.



Gambar 3.28 Sequence Diagram Get Location

f. Sequence Diagram Post data Via API

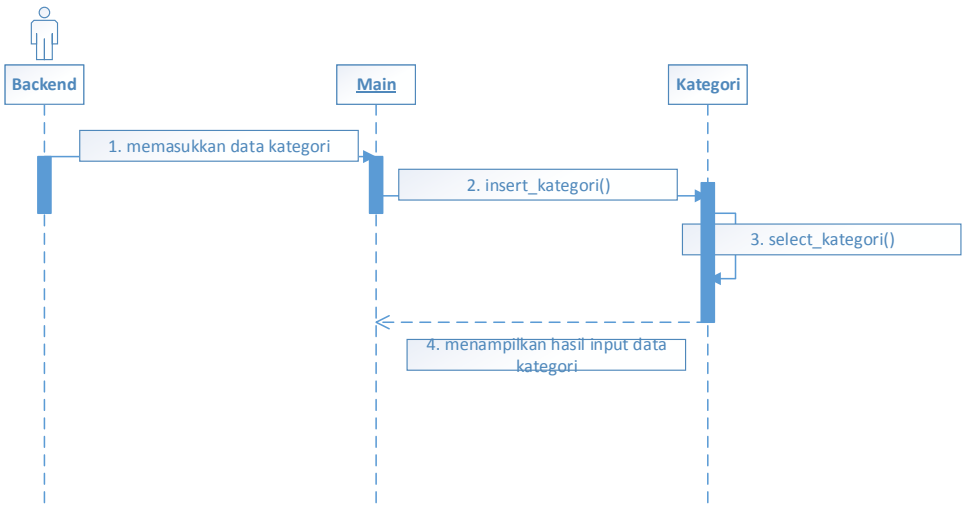
Berikut Gambar 3.28 Sequence Diagram yang menggambarkan sequence diagram post data via API.



Gambar 3.29 Sequence Diagram Post data Via API

g. Sequence Diagram Binding data kategori

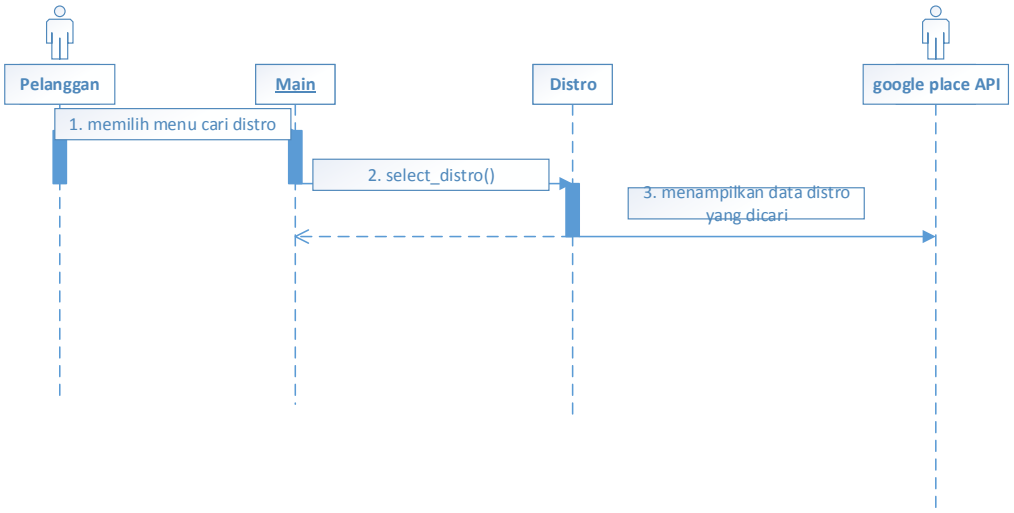
Berikut Gambar 3.29 Sequence Diagram yang menggambarkan sequence diagram binding data kategori.



Gambar 3.30 Sequence Diagram Binding data Kategori

h. Sequence Diagram Retrieve Location

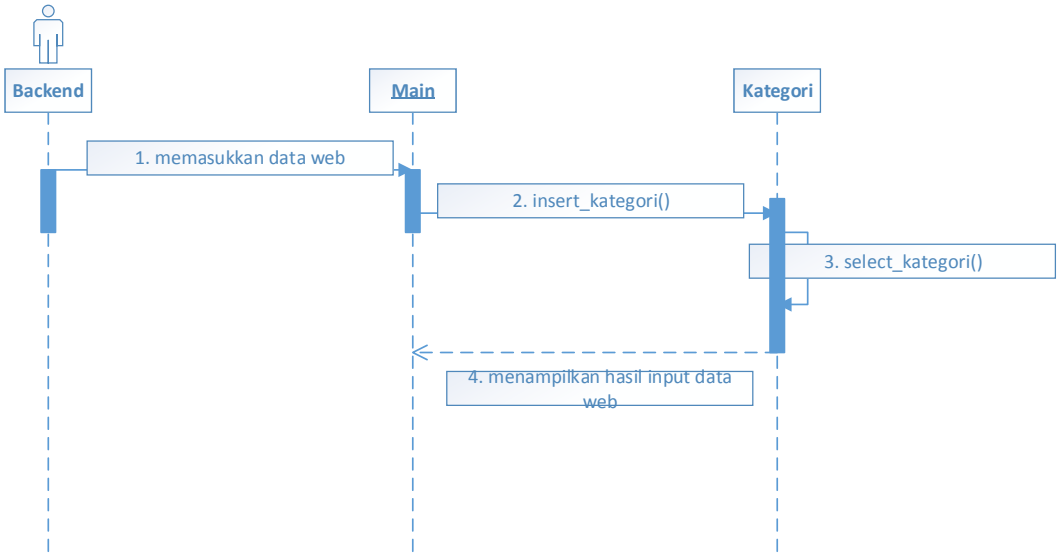
Berikut Gambar 3.30 Sequence Diagram yang menggambarkan sequence diagram retrieve location.



Gambar 3.31 Sequence Diagram Retrieve Location

i. Sequence Diagram Binding data web

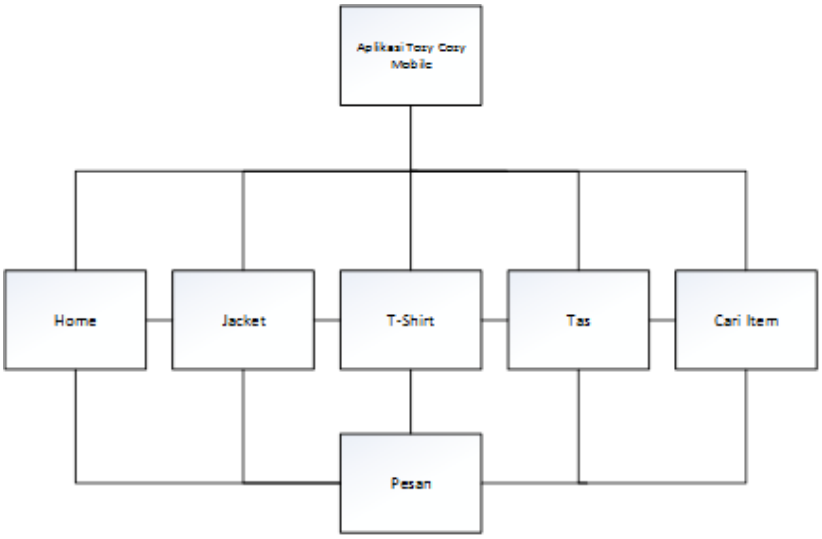
Berikut Gambar 3.31 Sequence Diagram yang menggambarkan sequence diagram binding data web.



Gambar 3.32 Sequence Diagram Retrieve Web

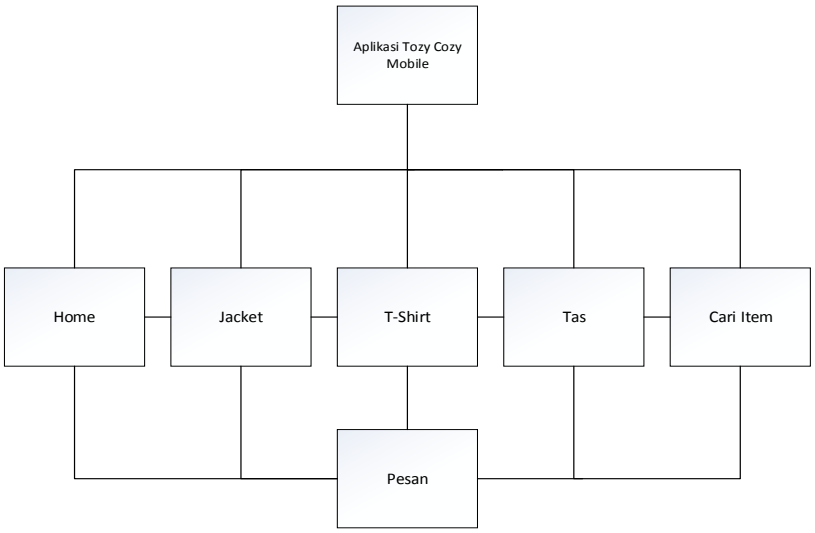
3.2.9 Perancangan Arsitektur Menu

Perancangan arsitektur menyediakan software enginer satu gambaran dari stuktur menu program atau blueprint dari perangkat lunak yang akan dibuat. Tujuan perancangan ini adalah untuk membangun struktur program secara moduler dan menggambarkan hubungan kendali diantara modul program. Berikut Gambar 3.32 Arsitektur Menu Web Admin adalah yang mengambarkan arsitektur menu web Admin.



Gambar 3.33 Perancangan Arsitektur Menu Web Admin

Berikut Gambar 3.33 Arsitektur Menu Aplikasi Mobile adalah yang mengambarkan arsitektur menu web Admin.



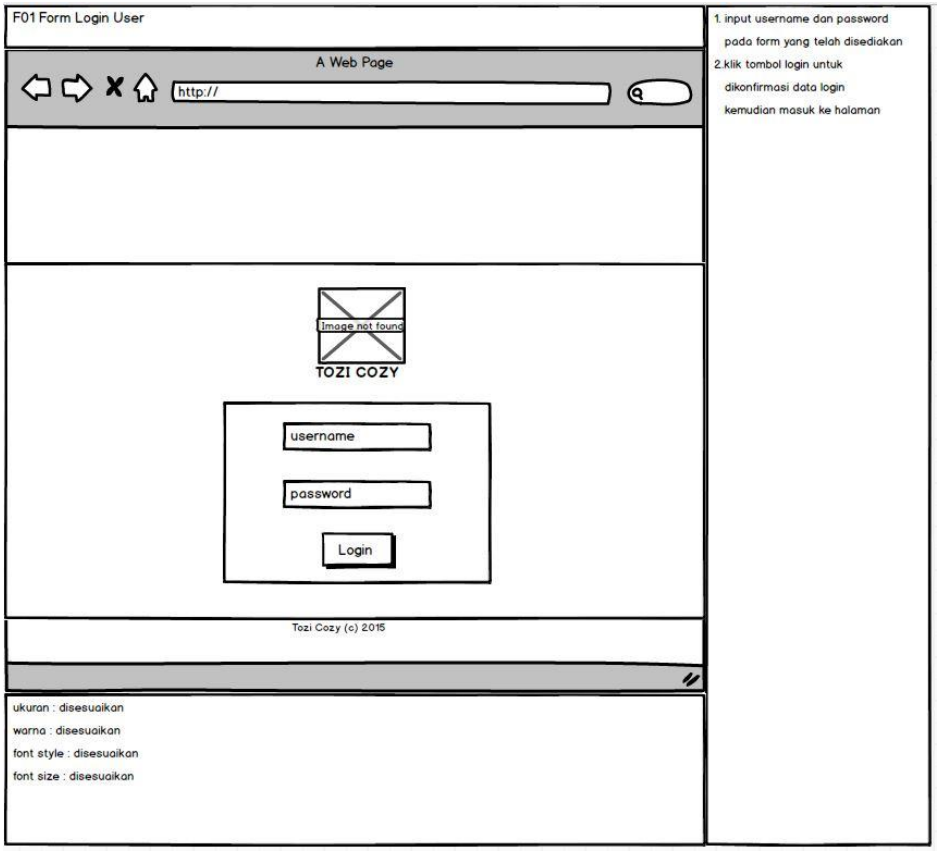
Gambar 3.34 Perancangan Arsitektur Menu Aplikasi Mobile

3.2.10 Perancangan Antar Muka

1. Perancangan Antar Muka Web

   a. Perancangan Antar Muka *Web* login

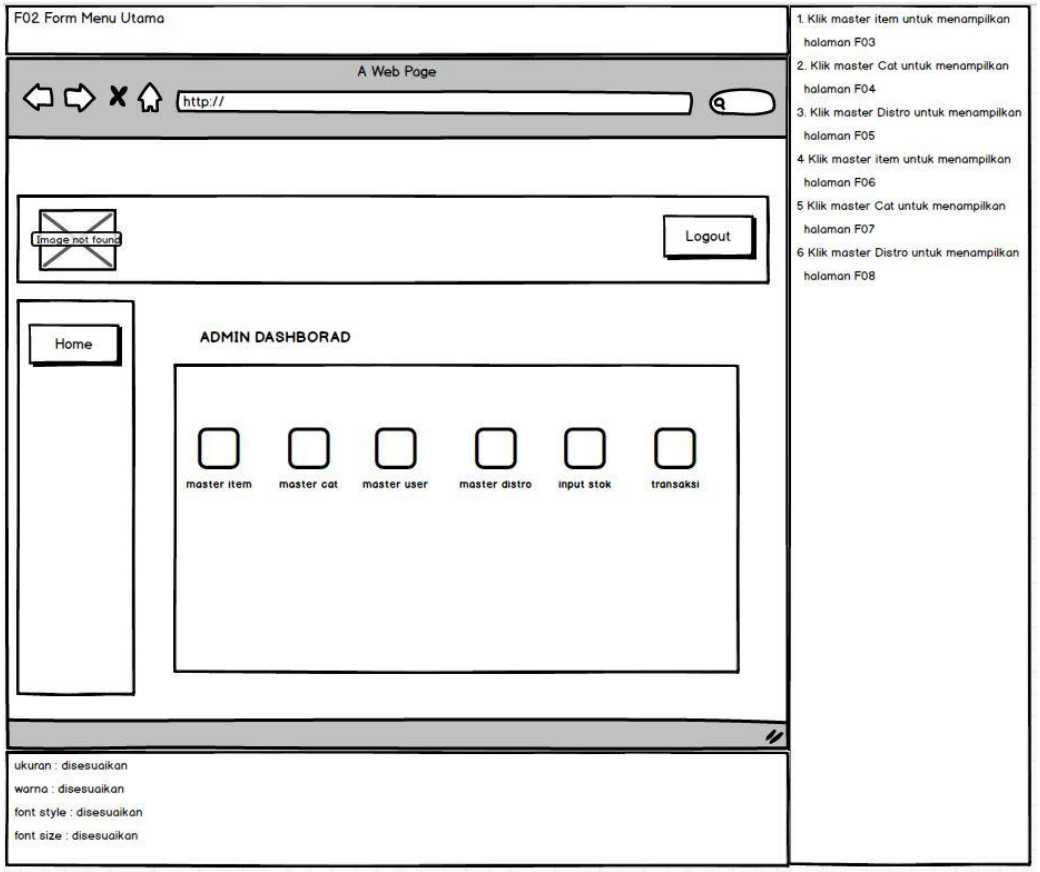
      Berikut adalah Gambar 3.34 Antar muka login yang mengambarkan antar muka login.



Gambar 3.35 Antar Muka Web Login

b. Perancangan Antar Muka Home

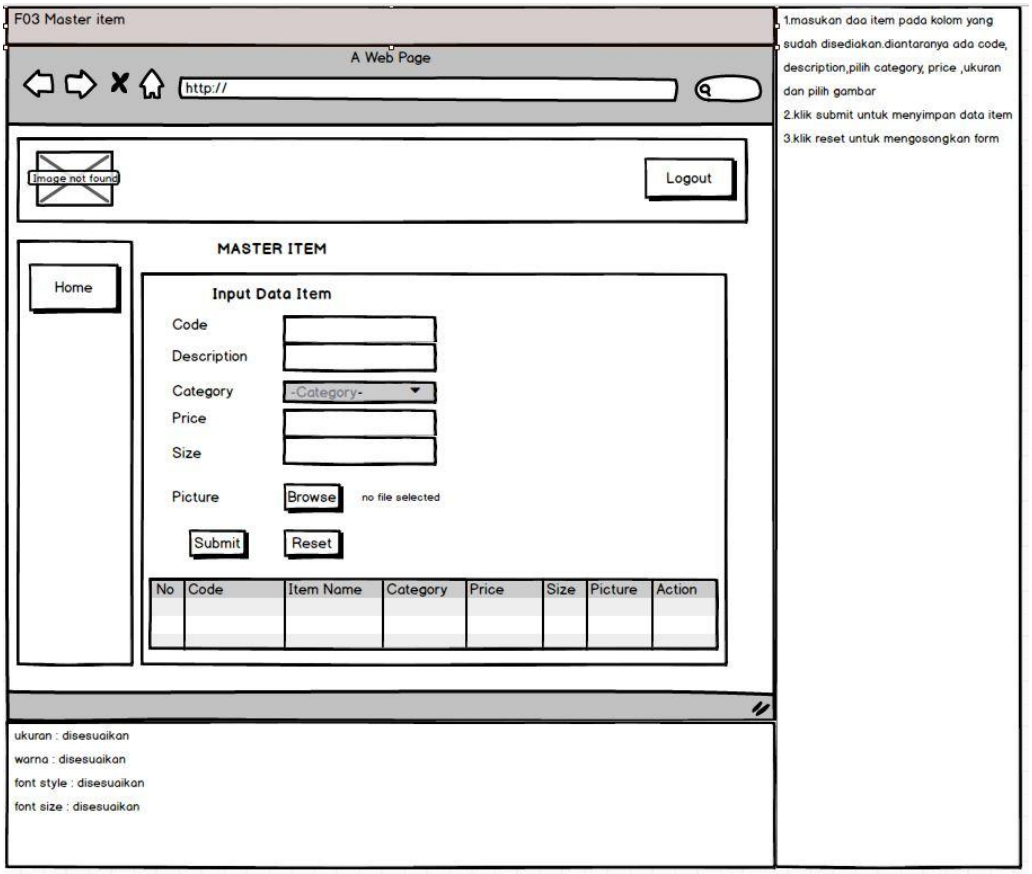
Berikut adalah Gambar 3.36 Antar muka home yang mengambarkan antar muka home.



Gambar 3.36 Antar Muka Home

c. Perancangan Antar Muka Master item

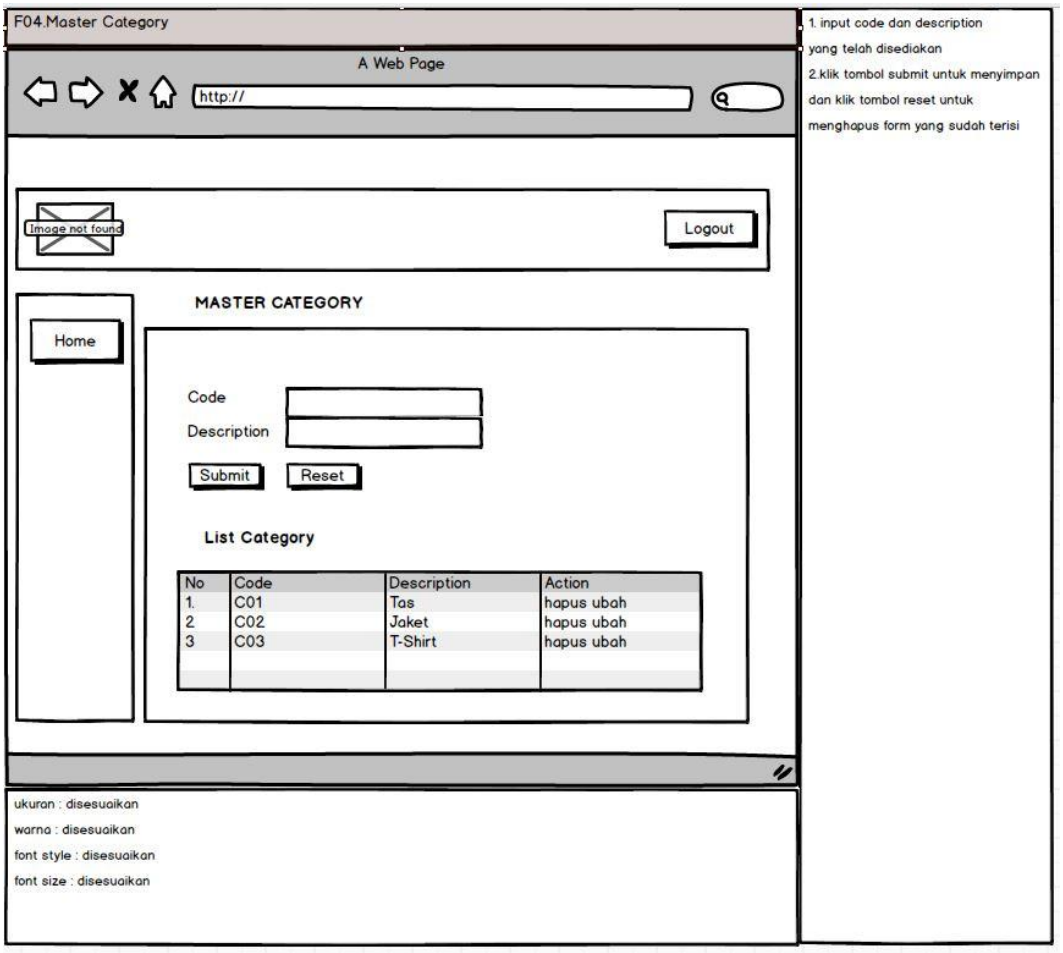
Berikut adalah Gambar 3.37 Antar muka Master Item yang mengambarkan antar muka master item.



Gambar 3.37 Antar Muka Master Item

d. Perancangan Antar Muka Master Kategori

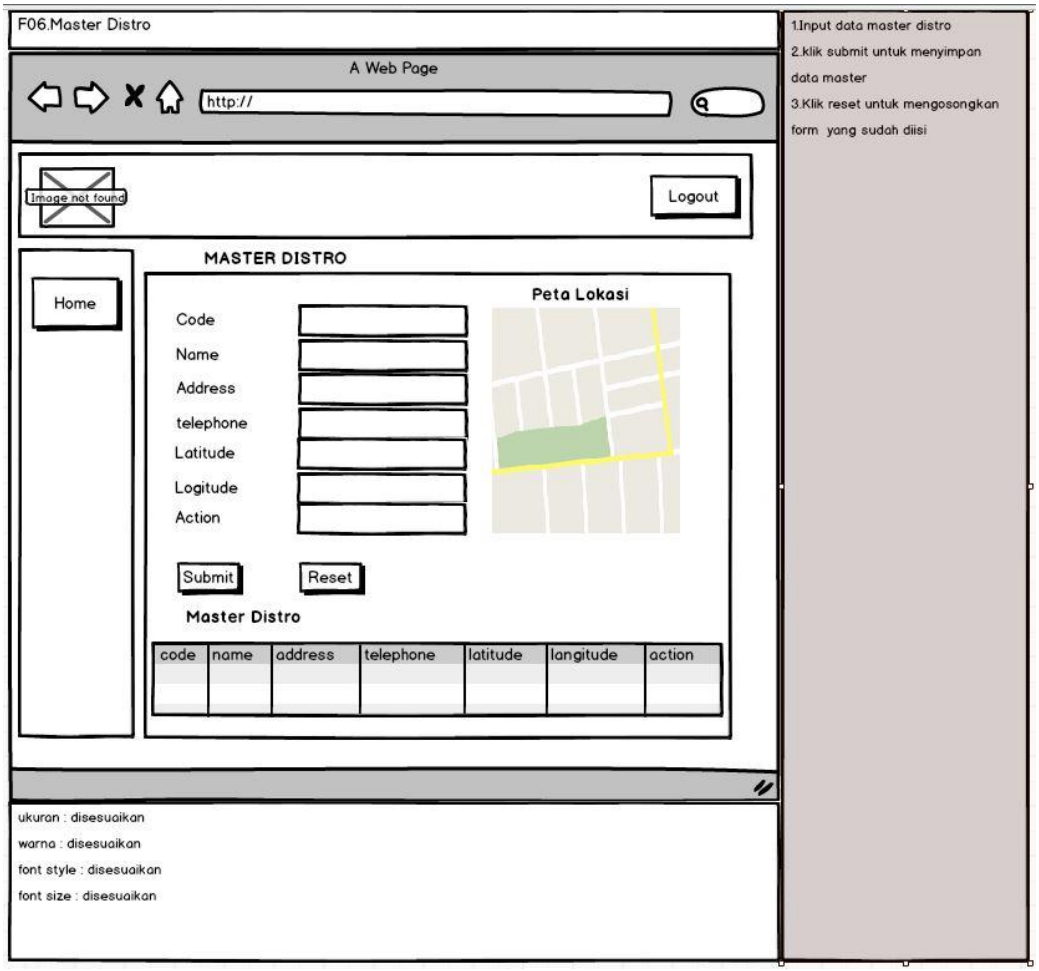
Berikut adalah Gambar 3.38 Antar muka Master Kategori yang mengambarkan antar Master kategori.



Gambar 3.38 Antar Muka Master Kategori

e. Perancangan Antar Muka Master Dsitro

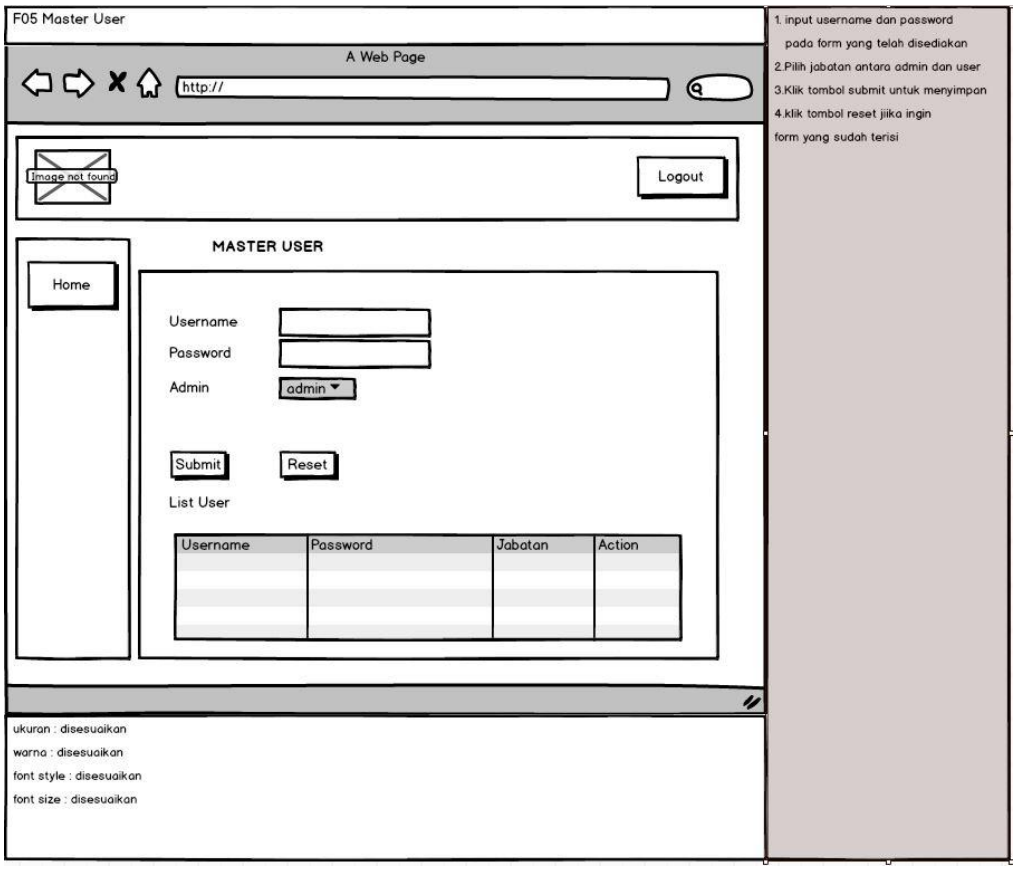
Berikut adalah Gambar 3.39 Antar muka Master Distro yang mengambarkan antar muka Master distro.



Gambar 3.39 Antar Muka Master Distro

f. Perancangan Antar Muka Master User

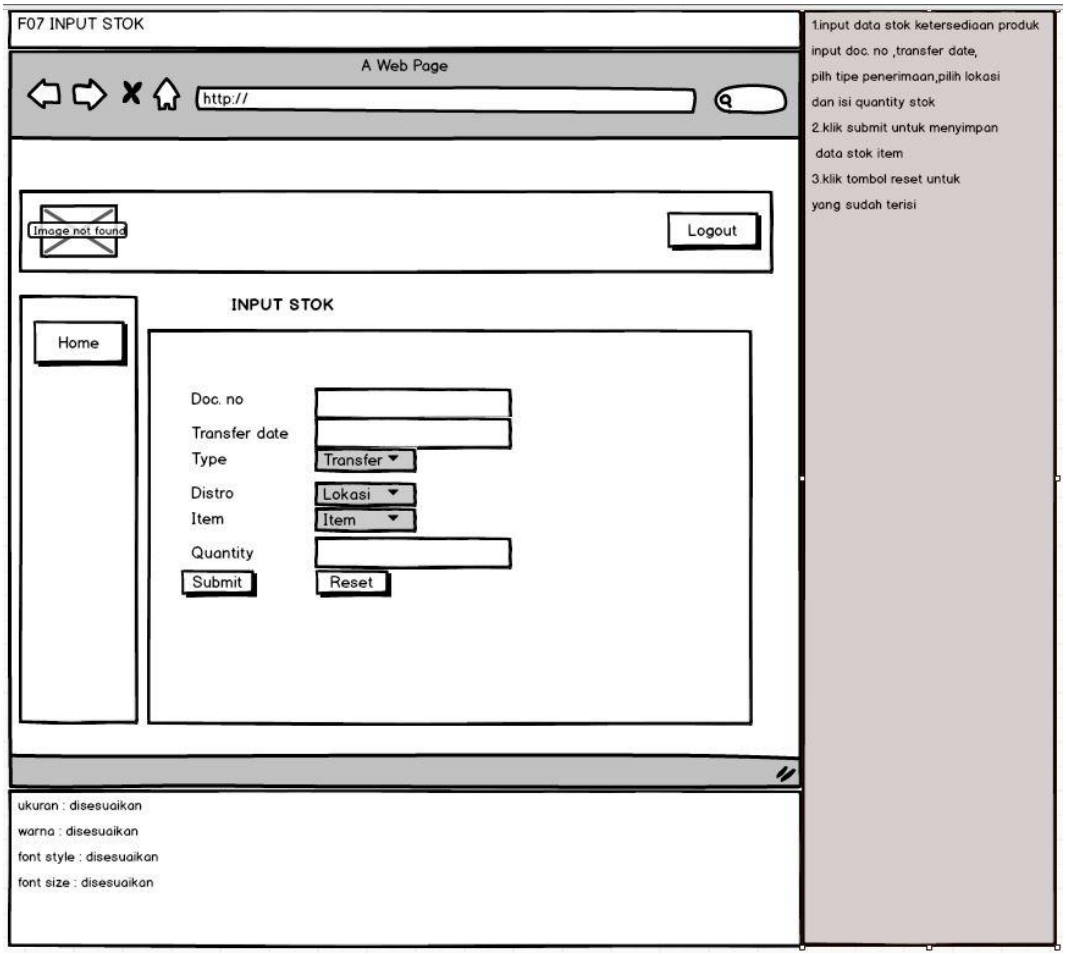
Berikut adalah Gambar 3.40 Antar muka Master user yang mengambarkan antar muka Master user.



Gambar 3.40 Antar Muka Master User

g. Perancangan Antar Muka Input Stok

Berikut adalah Gambar 3.41 Antar muka Input Stok yang mengambarkan antar muka input stok.

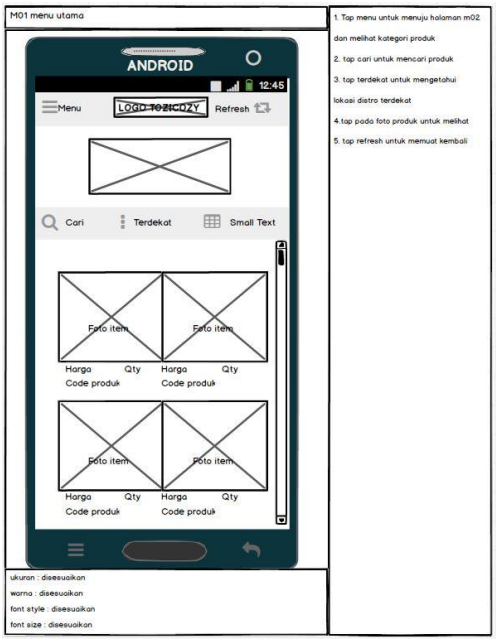


Gambar 3.41 Antar Muka Input Stok

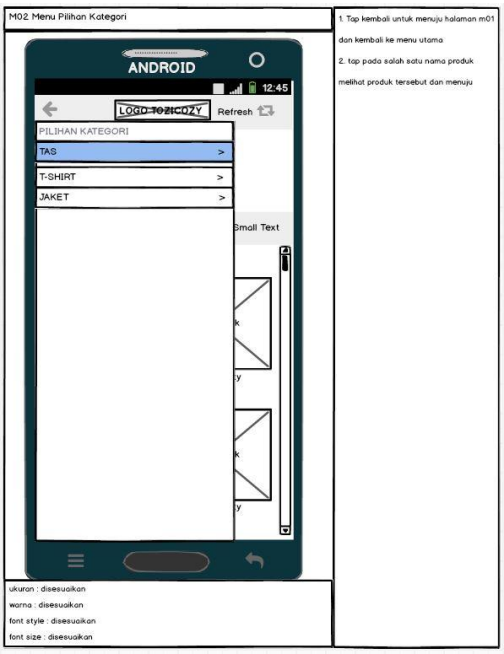
3.2.11 Perancangan Antar Muka *Mobile*

a. Perancangan Antar Muka *Mobile*

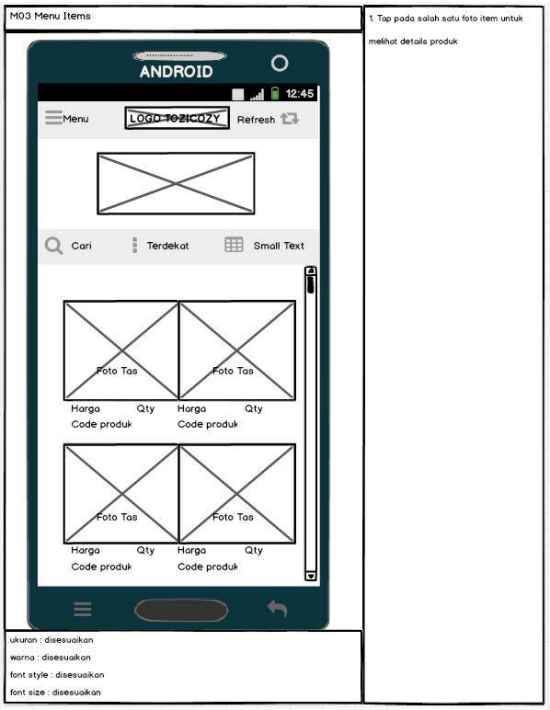
Berikut adalah Gambar 3.42 Antarmuka Home yang mengambarkan rancangan antarmuka Home



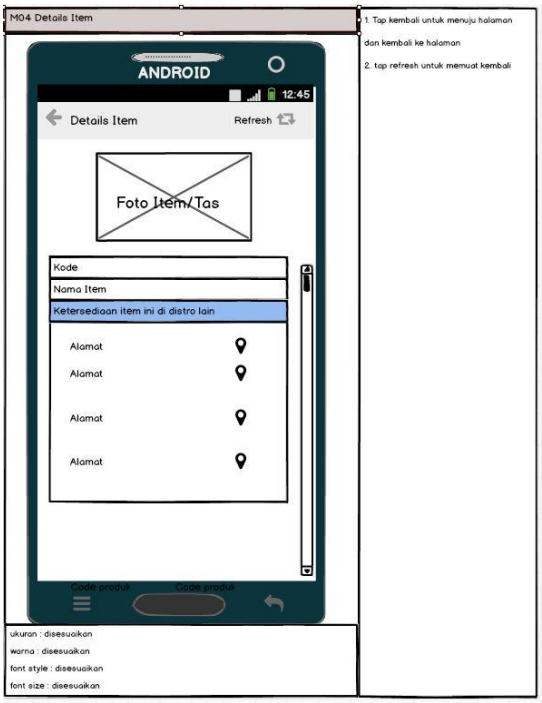
Gambar 3.42 Antar Muka Mobile Home



Gambar 3.43 Antar Muka Mobile Kategori



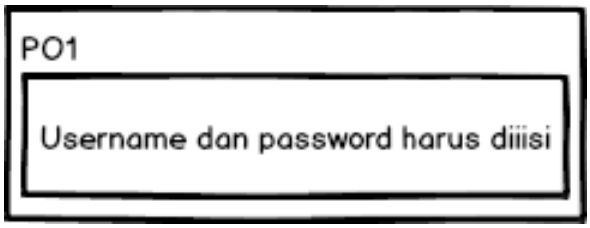
Gambar 3.44 Antar Muka Mobile Item



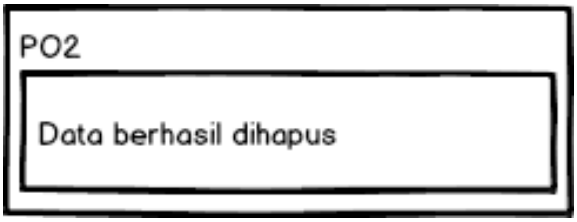
Gambar 3.45 Antar Muka Mobile Detail Item

3.2.12 Perancangan Pesan

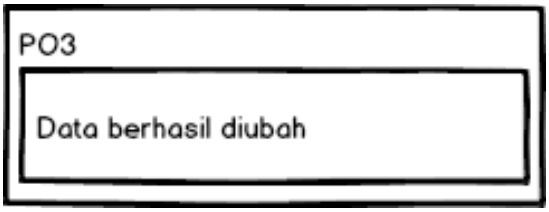
Perancangan pesan pada perangkat lunak yang dibangun terbagi menjadi dua perancangan pesan *web*.



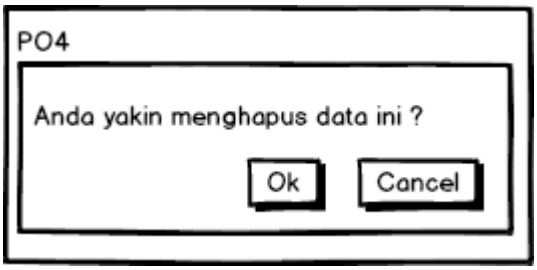
Gambar 3.46 Perancangan Pesan P01



Gambar 3.47 Perancangan Pesan P02



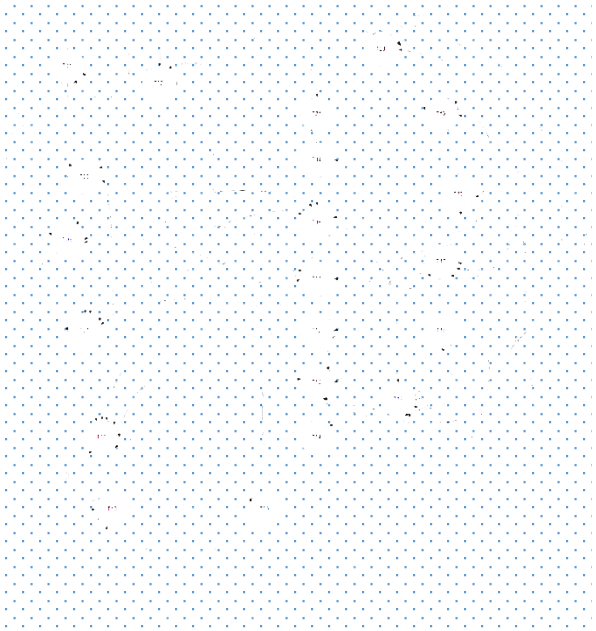
Gambar 3.48 Perancangan Pesan P03



Gambar 3.49 Perancangan Pesan P04

3.2.13 Jaringan Semantik

Jaringan Semantik merupakan gambaran pengetahuan grafis yang menunjukkan hubungan antar objek (lingkaran) dan garis yang menggambarkan formasi antar objek tersebut. Adapun jaringan semantik untuk perangkat yang dibangun dapat dilihat pada Gambar 3.49 Jaringan Semantik.



Gambar 3.50 Jaringan Semantik