

APLIKASI INVENTORI OUTLET PADA PT. KONIMEX JAKARTA BERBASIS ANDROID

Luci Kanti Rahayu¹, Agustiena Merdekawati², Wasis Zakaria³

Sistem Informasi, Universitas BSI, Jakarta¹

Sistem Informasi, Universitas BSI, Jakarta²

Teknik Informatika, STMIK Nusa Mandiri, Jakarta³

Email: luci.lkr@bsi.ac.id¹, agustiena.atd@bsi.ac.id², wasiszakaria@gmail.com³

ABSTRAK

Perkembangan Perangkat seluler yang telah menjamur pada gilirannya dapat dimanfaatkan manusia sebagai media komunikasi jarak jauh yang efisien, *handphone* genggam misalnya lebih *fleksibel* dan mudah untuk dibawa kemana-mana sehingga selalu menemani pengguna dimanapun berada, perkembangan sistem operasi *handphone* pun telah mengalami kemajuan belum lama ini sistem operasi android telah berkembang pesat dan hampir menguasai pasar *mobile*. Proses pencatatan barang yang dilakukan manual pada suatu outlet terjadi dua kali, pencatatan yang dilakukan pegawai pada outlet dan selanjutnya proses pencatatan yang dilakukan pegawai pada kantor sehingga berkemungkinan adanya kesalahan pencatatan pada jumlah barang. Oleh karena itu dibutuhkan suatu aplikasi inventori outlet yang mampu mencatat data secara langsung dari pegawai yang berada pada suatu outlet ke komputer yang berada di kantor pusat yang menampung data pencatatan barang dari berbagai outlet.

Kata kunci : Darurat, Android, Lokasi

I. PENDAHULUAN

Perkembangan ilmu pengetahuan dan teknologi dewasa ini telah mendorong manusia pada kehidupan yang lebih baik. Terlebih lagi dengan adanya *smartphone* manusia diberi kemudahan-kemudahan dalam menyelesaikan pekerjaan di berbagai bidang, seperti bidang penjualan dan persediaan barang.

Penelitian sebelumnya dilakukan Wibowo dan Emha yang berjudul Pembuatan Aplikasi Inventori Pada Toko Bangunan Wahyu Wonosari Yogyakarta Berbasis Android yaitu sebuah aplikasi yang dapat membantu proses pencatatan persediaan data tanpa kendala terikat ruang dan waktu.

Menurut [1] "pada saat ini pemrograman telah berkembang menjadi teknologi internet tanpa kabel (wireless internet). Sehingga koneksi dengan perangkat *ponsel/mobile (mobile devices)* pun sudah menjadi hal yang biasa. Dan dengan teknologi yang sudah ada, aplikasi-aplikasi semacam *mobile devices* dapat dibuat dan dikembangkan dalam *Java*

Menurut [1] "proses pencatatan barang digudang terjadi dua kali, pencatatan secara manual di gudang dan selanjutnya proses penginputan data ke komputer. Sehingga proses pencatatannya tentu akan membutuhkan waktu yang lama dan membutuhkan biaya yang lebih besar."

Penelitian [2] membahas tentang Pembuatan Aplikasi Persediaan Barang Pada UD Adi Jaya Berbasis *Android*

penelitian ini membuat aplikasi persediaan barang pada gudang berbasis *android*.

Menurut [2] "Manajemen gudang yang baik tentu tidak lepas dari data-data yang banyak dan dapat berubah sewaktu-waktu, dalam hal ini adalah data-data transaksi barang masuk, maupun keluar dan tentunya data barang itu sendiri. Perubahan data tersebut harus tersimpan dengan mudah untuk dicari pada saat dibutuhkan".

Pegawai pada outlet dan pegawai kantor PT. Konimex berkeinginan agar proses mencatat barang dapat dilakukan dengan mudah, saat ini terjadi dua kali pencatatan data barang yang pertama dilakukan oleh pegawai pada outlet, kemudian dikirim melalui *Short Message Service (SMS)* dan *instant messaging (IM)*, lalu dicatat kembali di kantor. Dengan adanya aplikasi ini pihak kantor tidak perlu lagi mencatat data karena aplikasi akan membuat laporan dari pencatatan yang dilakukan oleh pegawai outlet.

Oleh karena permasalahan di atas, maka akan dibuat aplikasi yang dapat mencatat stok barang menggunakan android, sehingga memberikan beberapa manfaat, yaitu: memberikan kemudahan terhadap para pengguna dalam pencatatan barang yang keluar dan masuk pada outlet.

II. KAJIAN LITERATUR

2.1. SDLC (*Software Development Life Cycle*)

Awal melakukan penelitian ini dengan melakukan pengumpulan data untuk mendapatkan permasalahan serta data-data yang dibutuhkan, yaitu dengan cara observasi dan wawancara. Dalam observasi, mengamati sistem inventori yang dilakukan masih manual, yaitu masih menggunakan pencatatan melalui buku.

Model pengembangan sistem dalam penelitian ini menggunakan SDLC (*Software Development Life Cycle*) atau sering disebut juga dengan *System Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang teruji baik). [3]

Metode penelitian dalam penelitian ini menggunakan metode *waterfall*. Tahapan pada metode *waterfall* sebagai berikut: [3]

1. **Analisa Kebutuhan Perangkat Lunak**
Merupakan proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh user.
2. **Desain**
Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak, termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisa kebutuhan kerepresentasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.
3. **Pembuatan Kode Program**
Desain harus ditranslasikan kedalam kode program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.
4. **Pengujian**
Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.
5. **Pendukung (*Support*) atau Pemeliharaan (*maintenance*)**
Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses

pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.2. Android

Menurut [4] "Android merupakan *OS Mobile* yang tumbuh ditengah OS lainnya yang berkembang dewasa ini. OS lainnya *Windows Mobile, I-Phone OS, Symbian*, dan masih banyak lagi, juga menawarkan kekayaan isi dan keoptimalan berjalan diatas perangkat *hardware* yang ada. Akan tetapi, OS yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu dengan adanya keterbatasan dari aplikasi pihak ketiga untuk mendapatkan data asli ponsel, berkomunikasi antar proses serta keterbatasan distribusi aplikasi pihak ketiga untuk platform mereka. Andorin menawarkan sebuah lingkungan yang berbeda untuk pengembangan. Setiap aplikasi memiliki tingkatan yang sama. Android tidak membedakan antara aplikasi inti dengan aplikasi pihak ketiga. API yang disediakan menawarkan akses *har-ware*, maupun data-data ponsel sekalipun, atau data *system* sendiri. Bahkan pengguna dapat menghapus aplikasi inti dan menggantinya dengan aplikasi pihak ketiga.

2.3. Java

Menurut [5] "Java merupakan sebuah *platform* sekaligus bahasa pemrograman tingkat tinggi yang mempunyai kriteria sederhana, berorientasi objek, terdistribusi, dinamis, aman dan lainnya. Bahasa ini dikembangkan dengan model yang mirip seperti bahasa *c++* dan *Smalltalk* namun lebih mudah dipakai, juga memiliki platform *independen* yang dapat dijalankan pada sistem operasi apapun.

Sedangkan *Java Compiler* untuk program Java berbentuk *Java Development Kit (JDK)* yang diproduksi oleh Sun Microsystems. JDK menyediakan 2 program utama yaitu *javac* (program untuk meng-*compile* kode sumber) dan *Java* (program untuk meluncurkan aplikasi).

2.4. XML

Menurut [6] "Menurut *Extensible Markup Language (XML)* merupakan bahasa markip yang didesain untuk menjadi sarana yang mudah dalam mengirimkan dokumen melalui web". Berbeda dengan *Hypertext Markup Language (HTML)*. XML memungkinkan pengguna untuk mendefinisikan *custom tag*. Namun, standar XML tidak memiliki constrain semantik pada arti dari dokumen tersebut.

2.5. MySQL dan phpMyadmin

Menurut [7] "MySQL adalah *database* yang menghubungkan *script PHP* menggunakan perintah *query*

dan *escaps* character yang sama dengan *php*. *MySQL* mempunyai tampilan *client* yang memudahkan dalam mengakses *database* dengan kata sandi untuk mengizinkan proses dilakukan sedangkan *phpMyadmin* adalah sebuah *software* yang berbentuk seperti halaman situs yang terdapat pada *web server*. Fungsi dari halaman ini adalah sebagai pengendali *database MySQL* sehingga pengguna *MySQL* tidak perlu repot untuk menggunakan perintah-perintah SQL”.

2.6. Eclips

Menurut [5] “Eclips yang diluncurkan oleh IBM pada tanggal 5 November 2001 merupakan sebuah IDE yang gratis dan *open source* atau yang dapat dikembangkan dan digunakan untuk membangun sebuah program komputer dan dapat dijalankan di semua Platform”.

2.7. Android SDK (Software Development Kit)

Menurut [8] “Android SDK adalah *tools API (Application Programming Interface)* yang diperlukan untuk mulai mengembangkan aplikasi pada *platform Android* menggunakan bahasa pemrograman Java. Android merupakan *subset* perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang di- *release* oleh google saat ini di sediakan ANDROID (*Software Development Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada *platform Android* menggunakan bahasa pemrograman java. Sebagai *platform* aplikasi-netral, Android memberi anda kesempatan untuk membuat aplikasi yang kita butuhkan yang merupakan bukan aplikasi bawaan *handphone* atau *smartphone*.

2.8. Android Development Tools (ADT)

Menurut [8] “*Android Development Tools (ADT)* adalah *plugin* yang didesain untuk IDE Eclipse yang memberikan kita kemudahan dalam mengembangkan aplikasi android dengan menggunakan IDE Eclipse. Dengan menggunakan ADT untuk Eclipse akan memudahkan kita dalam membuat aplikasi *project android*, membuat GUI aplikasi, dan menggunakan komponen-komponen lainnya, begitu juga kita dapat melakukan *running* aplikasi menggunakan SDK melalui *eclipse*. Dengan ADT juga kita dapat melakukan pembuatan *package android (.apk)* yang digunakan untuk distribusi aplikasi android yang kita rancang.

2.9. Algoritma Sequential Search

Algoritma *sequential search* adalah salah satu algoritma yang digunakan untuk memecahkan masalah pencarian data pada suatu larik/ array. Cara kerja dari algoritma ini adalah dengan menelusuri elemen-elemen array dari awal sampai akhir, dimana data tidak perlu diurutkan terlebih dahulu. Kemungkinan terbaik dari

algoritma ini adalah jika data yang dicari berada pada elemen array yang terdepan sehingga waktu yang dibutuhkan untuk pencarian data semakin singkat.

2.10. Flow Chart

Menurut [9] “*Flowchart* merupakan gambar atau bagan yang memperlihatkan urutan dan hubungan atau proses beserta instruksinya. Gambaran ini dinyatakan dengan simbol. Dengan demikian setiap simbol menggambarkan proses tertentu. Sedangkan hubungan antar proses digambarkan dengan garis penghubung.

2.11. UML

Menurut [9] “*Unified Modelling Language (UML)* adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak”.

UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Seperti bahasa lainnya, UML mendefinisikan notasi dan *syntax* atau *semantik*. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya : *Grady Booch OOD (Object Oriented Design)*, *Jim Rumbaugh OMT (Object Modeling Technique)*, dan *Ivar Jacobson OOSE (Object Oriented Software Engineering)*.

III. METODELOGI PENELITIAN

Teknik yang digunakan untuk mengumpulkan data dalam penelitian ini adalah sebagai berikut:

- a. *Observaion* (Pengamatan)
Penulis melakukan peninjauan langsung pada PT. Konimex, Jakarta dengan cara melihat bagaimana proses pencatatan barang yang selama ini dilakukan.
- b. *Interview* (Wawancara)
Metode ini merupakan cara mengumpulkan data dengan tanya-jawab langsung dengan pegawai yang bersangkutan.
- c. *Library Research* (Studi Pustaka)
Metode pengumpulan data dengan cara penulis mempelajari buku-buku, jurnal, internet ataupun sumber-sumber lainnya untuk mendapatkan dasar teori dan penjelasan yang berhubungan dengan masalah yang dibahas.

Tahap pengembangan sistem dalam penelitian inia adalah:

1. Analisa Kebutuhan Perangkat Lunak

Pengembangan rencana pengembangan proyek dan dokumen perencanaan lainnya. Menyediakan dasar untuk mendapat sumber data (*resources*) yang dibutuhkan untuk memperoleh solusi. Setelah penulis mendapatkan semua informasi atau data yang diperlukan guna pengembangan *software*, penulis melakukan analisis terhadap data guna membuat model pembuatan aplikasi yang akan diangkat. Dalam penelitian ini penulis melakukan analisa terhadap aplikasi inventori outlet berbasis *android*.

2. Desain

Bertujuan untuk mendesain sistem aplikasi yang akan dirancang dan mengidentifikasi komponen-komponen sistem informasi yang akan didesain secara rinci. Tahap ini dilakukan setelah tahap analisa kebutuhan *software* selesai dilakukan, karena semua informasi atau data telah didapatkan dan disusun. Desain aplikasi ini meliputi *Flowchart* dan diagram UML seperti diagram *use case*, diagram *activity*, diagram *sequence*, dan diagram *class*.

3. Pembuatan kode program

Termasuk pada persiapan implementasi, implementasi perangkat lunak pada lingkungan produksi dan menjalankan resolusi dari permasalahan yang teridentifikasi dari fase integritas dan pengujian.

4. Pengujian

Proses pengujian dilakukan setelah pembuatan program selesai dibuat, adapun bentuk pengujian yang digunakan yaitu menggunakan *blackbox* dan *whitebox testing*.

IV. HASIL DAN PEMBAHASAN

4.1. Analisa Kebutuhan

Sebelum membuat aplikasi, ada baiknya menganalisa kebutuhan apa saja yang diperlukan dalam pembuatan program, kebutuhan yang akan dianalisa mencakup kebutuhan dari sisi pengguna, kebutuhan dari sistem, yaitu sebagai berikut:

a. Analisa kebutuhan pengguna

Dalam *Menu* utama pengguna akan memilih *menu* outlet yang diperlukan untuk memilih outlet pengguna yang sesuai dengan outlet dimana pengguna ditugaskan. Setelah memilih outlet pengguna akan memilih barang untuk di input sesuai dengan barang yang pengguna kelola dengan mudah.

b. Analisa kebutuhan sistem

Untuk analisa kebutuhan sistem sebagai berikut:

1. Sistem dapat menampilkan daftar outlet pengguna dengan akurat.
2. Sistem dapat menampilkan barang yang akan di input sesuai dengan barang yang dikelola oleh pengguna.

4.2. Desain

Setelah melakukan analisa kebutuhan, tahap selanjutnya membuat desain. Ada beberapa desain yang dibuat oleh penulis, yaitu:

1. Rancangan Algoritma

Metode algoritma program yang penulis gunakan pada aplikasi ini yaitu metode algoritma *sequential*, pada bagian algoritma ini hanya akan dijelaskan sebagian kasus yang dipakai pada algoritma program yang telah penulis dibuat, karena algoritma kasus pada contoh lainpun mempunyai penjelasan yang sama, berikut adalah desain algoritma pada kasus pencarian outlet.

a) Inisialisasi button *btnOutlet*.

```
Button btnOutlet=(Button)findViewById(R.id.btnOutlet);
btnOutlet.setOnClickListener(new View.OnClickListener() {
    public void onClick(View arg0) {
        Intent i = new Intent(MenuUtama.this,listOutlet.class);
        i.putExtra("kode_pengguna", kode_pengguna);
        i.putExtra("kode_order", kode_order);
        i.putExtra("kode_kategori", kode_kategori);
        startActivity(i);
    }
});
```

Keterangan:

Pada saat button outlet di klik system akan membuka halaman list outlet.

b) Inisialisasi XML Parser

```
Utils ip=new Utils();
String myip=ip.getIP();
kode_pengguna=kode_pengguna.replace(" ", "_");
String URL = myip+"outlet/xml.php?kode_pengguna="+kode_pengguna+"&ket=ok";
Log.v("URL",URL);
xml = parser.getXmlFromUrl(URL);
```

Keterangan:

Pada saat button outlet di klik system akan mengirimkan kode pengguna.

c) Record XML Parser Outlet.

```

    $s = "select * from 'tboutlet' where 'kode_pengguna' like '%$kode_pengguna%'";
    $q=mysql_query($s);
    if(mysql_num_rows($q)>0){
        print "<outlet>\n";
        while($d=mysql_fetch_array($q){
            $kode_outlet=$d["kode_outlet"];
            $nama_outlet=$d["nama_outlet"];
            $alamat_outlet=$d["alamat_outlet"];
            $kode_pengguna=$d["kode_pengguna"];
            $telepon_outlet=$d["telepon_outlet"];
            $keterangan_outlet=$d["keterangan_outlet"];

            print "<record>\n";
            print " <kode_outlet>$kode_outlet/</kode_outlet>\n";
            print " <nama_outlet>$nama_outlet/</nama_outlet>\n";
            print " <alamat_outlet>$alamat_outlet/</alamat_outlet>\n";
            print " <kode_pengguna>$kode_pengguna/</kode_pengguna>\n";
            print " <telepon_outlet>$telepon_outlet/</telepon_outlet>\n";
            print " <keterangan_outlet>$keterangan_outlet/</keterangan_outlet>\n";
            print "</record>\n";
        }
    }

```

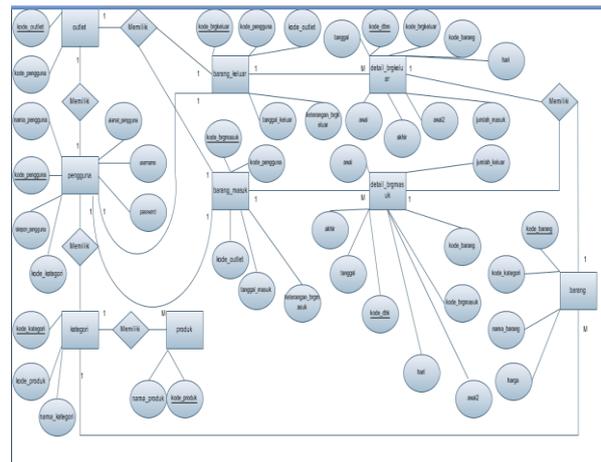
Keterangan:
Saat halaman list outlet terbuka system akan mencari daftar outlet pengguna berdasarkan kode pengguna.
d) Menampilkan Data Outlet ke List Outlet

```

for (int i = 0; i < jd; i++) {
    HashMap<String, String> map = new HashMap<String, String>();
    Element e = (Element) nl.item(i);
    String l_kode_outlet=parser.getValue(e, KEY_kode_outlet);
    if(l_kode_outlet.length()<1){l_kode_outlet="-";}arr_kode_outlet[i]=l_kode_outlet;
    String l_nama_outlet=parser.getValue(e, KEY_nama_outlet);
    if(l_nama_outlet.length()<1){l_nama_outlet="-";}arr_nama_outlet[i]=l_nama_outlet;
    String l_alamat_outlet=parser.getValue(e, KEY_alamat_outlet);
    if(l_alamat_outlet.length()<1){l_alamat_outlet="-";}arr_alamat_outlet[i]=l_alamat_outlet;
    String l_kode_pengguna=parser.getValue(e, KEY_kode_pengguna);
    if(l_kode_pengguna.length()<1){l_kode_pengguna="-";}arr_kode_pengguna[i]=l_kode_pengguna;
    String l_telepon_outlet=parser.getValue(e, KEY_telepon_outlet);
    if(l_telepon_outlet.length()<1){l_telepon_outlet="-";}arr_telepon_outlet[i]=l_telepon_outlet;
    String l_keterangan=parser.getValue(e, KEY_keterangan);
    if(l_keterangan.length()<1){l_keterangan="-";}arr_keterangan[i]=l_keterangan;
    map.put(KEY_kode_outlet, l_kode_outlet);
    map.put(KEY_nama_outlet, l_nama_outlet);
    map.put(KEY_alamat_outlet, l_alamat_outlet);
    map.put(KEY_kode_pengguna, l_kode_pengguna);
    map.put(KEY_telepon_outlet, l_telepon_outlet);
    map.put(KEY_heterangan, l_keterangan);
    recordsList.add(map);
}

```

Keterangan:
Halaman list outlet akan menampilkan daftar outlet pengguna.
2. Rancangan Database
Rancangan database ini menggunakan ERD (*Entiy Relationship Diagram*). Bentuk rancangan sebagai berikut:



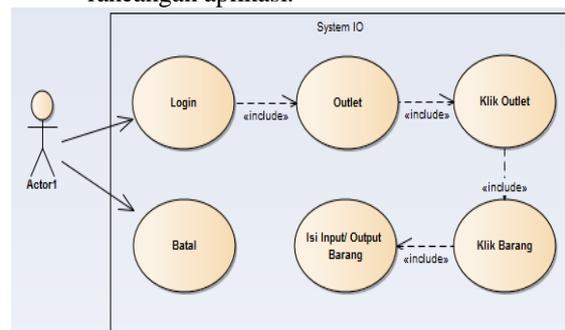
Gambar 4.1. Desain ERD

3. Rancangan Perangkat Lunak

Rancangan perangkat lunak ini, menggambarkan struktur hirarki dari komponen program, cara bagaimana komponen tersebut berinteraksi dan struktur data yang digunakan oleh komponen. Gambaran bagaimana elemen atau komponen fungsional perangkat lunak disusun, diorganisasi dan distrukturkan. Berikut rancangan perangkat lunak:

a) Usecase Diagram

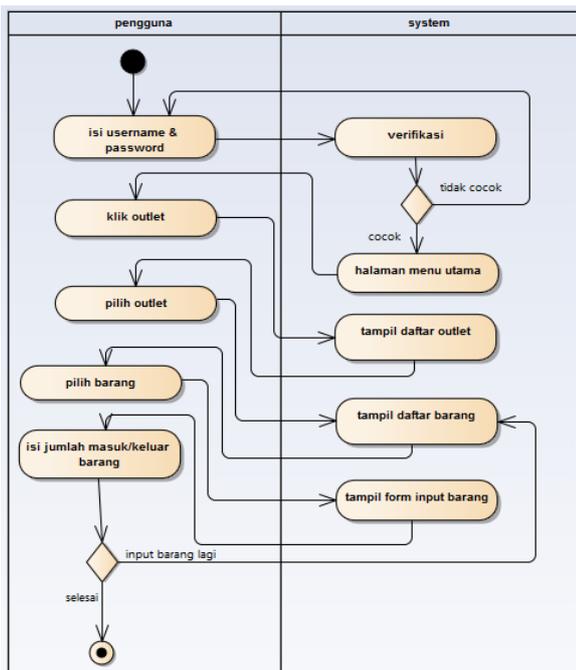
Usecase Diagram ini menggambarkan fungsionalitas sistem, berikut usecase diagram rancangan aplikasi:



Gambar 4.2. Desain Usecase Diagram

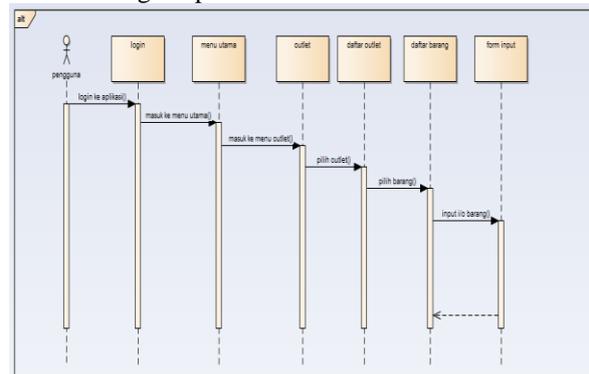
b) Activity Diagram

Activity Diagram menggambarkan rancangan aktifitas dari sistem. Berikut activity diagram rancangan aplikasi:



Gambar 4.3. Activity Diagram Outlet

Sequence Diagram, menggambarkan rancangan interaksi dengan sistem. Berikut sequence diagram rancangan aplikasi:

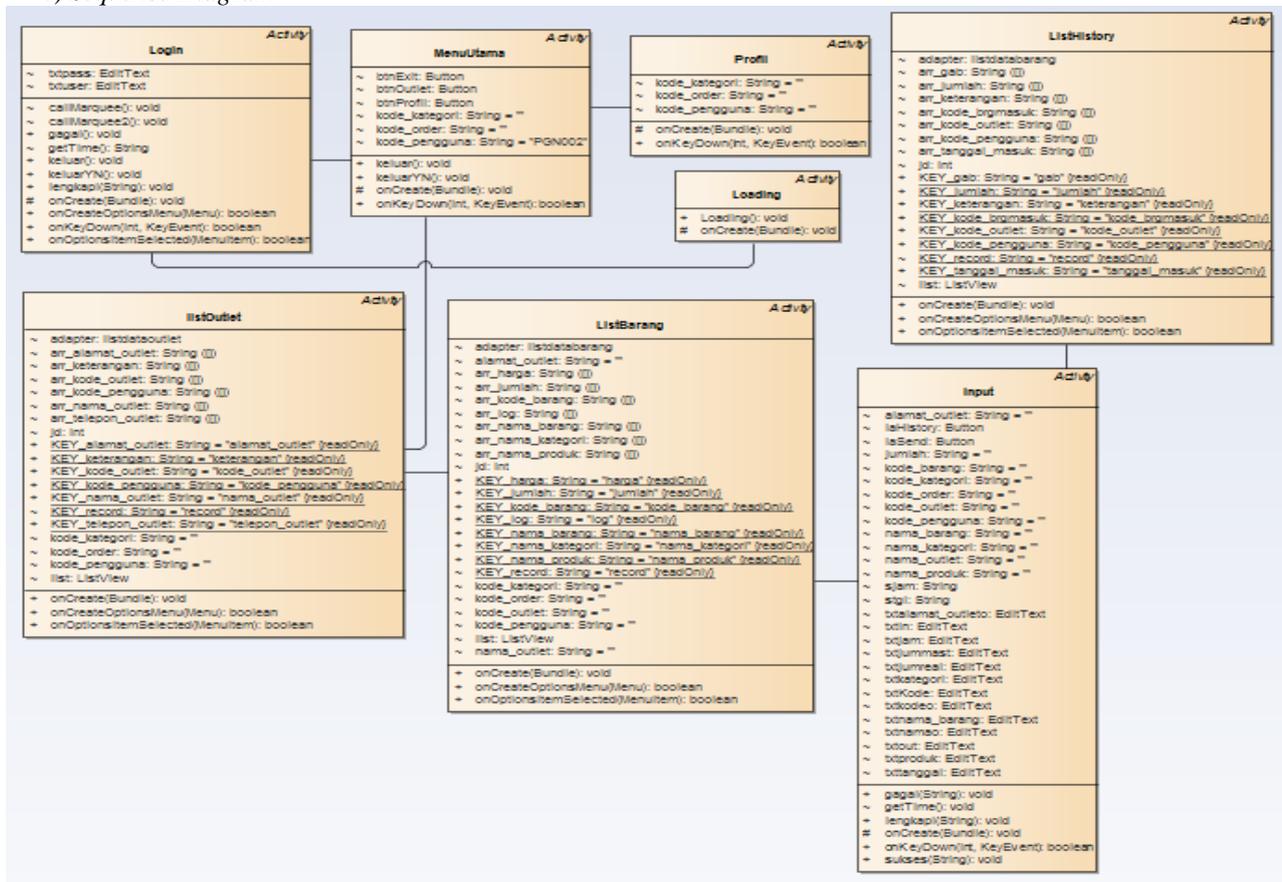


Gambar 4.4. Desain Sequence Diagram

d) Class Diagram

Berikut desain class diagram dari penelitian ini:

c) Sequence Diagram



Gambar 4.5. Class Diagram

4.3. Pembuatan Kode Program

Pembuatan koding program dari pembuatan aplikasi ini menggunakan bahasa pemrograman java, xml dan Mysql. Hasil dari pengkodean program yaitu berupa *user interface*. Berikut *user interface* dari aplikasi ini adalah:

1. *User interface* halaman login



Gambar 4.6. User Interface Login

Gambar di atas adalah tampilan login aplikasi Inventori Outlet. Pada halaman ini terdapat *Menu* Button.

2. *User interface* halaman menu utama



Gambar 4.7. User interface Menu Utama

Gambar di atas adalah tampilan dari *Menu Utama*, pada tampilan ini pengguna akan memilih antara dua

menu button yaitu outlet yang akan menampilkan daftar outlet pengguna dan log out untuk keluar dari aplikasi.

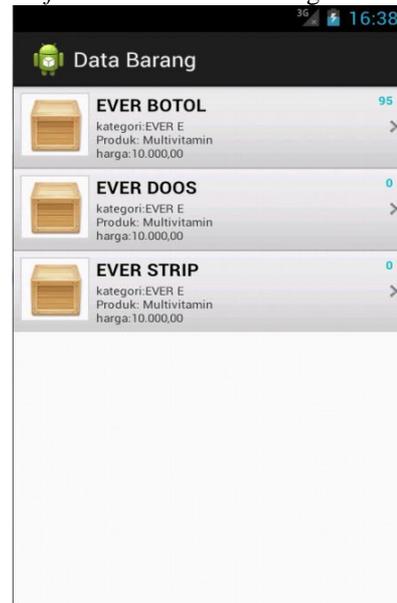
3. *User interface* halaman daftar outlet



Gambar 4.8. User Interface Daftar Outlet

Gambar di atas adalah tampilan dari Halaman Daftar Outlet, pada tampilan ini pengguna dapat melihat outlet yang pengguna tempati.

4. *User interface* halaman data barang



Gambar 4.9. User Interface Data Barang

5. *User Interface* halaman input barang



Gambar 4.9. User Interface Input Barang

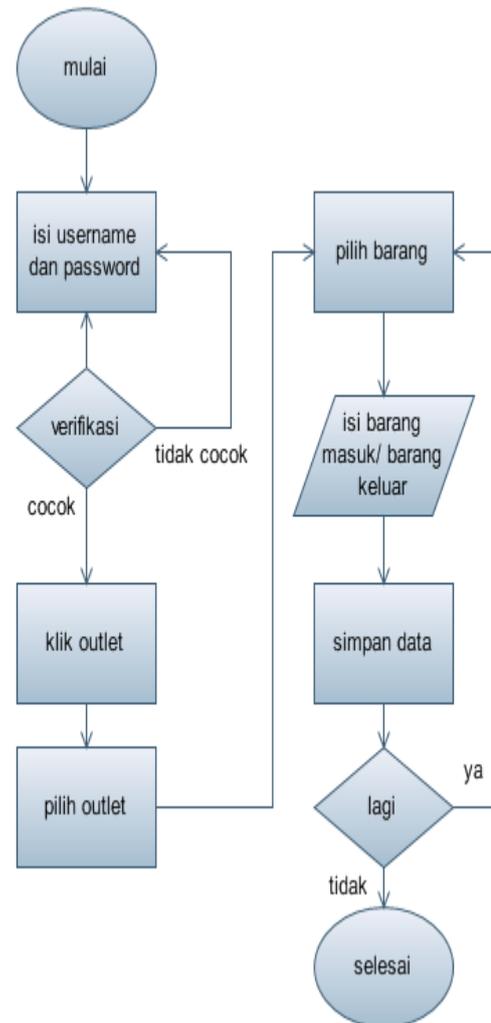
Gambar di atas adalah tampilan dari Halaman Input Barang pada tampilan ini pengguna akan menginput jumlah barang yang keluar/ masuk pada outlet.

4.4. Pengujian

Pengujian *White Box* adalah menguji perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi-fungsi, masukan dan keluaran yang sesuai dengan spesifikasi kebutuhan. Pengujian dilakukan terhadap pengiriman data yaitu sebagai berikut:

1. Pada Halaman Login isi username dan password.
2. Pada Halaman Menu Utama klik Outlet.
3. Pada Halaman Daftar Outlet pilih Outlet.
4. Pada Halaman Data Barang pilih Barang
5. Masukan jumlah barang masuk.
6. Masukan jumlah barang keluar.
7. Klik *Button Send*

Untuk lebih jelasnya, berikut ini penulis gambarkan dengan sebuah Bagan Alir.



Gambar 4.10. Bagan alir Pemakaian Aplikasi

Tabel 4.1
Pengujian *Black Box* Aplikasi Inventori Outlet

No	Test Case	Hasil yang diharapkan	Hasil yang didapatkan	Keterangan
1	Pengujian Login	Sistem dapat mencocokkan username dan password	Sistem dapat mencocokkan username dan password	Berhasil
2	Pengujian <i>Menu</i> Outlet	Sistem akan menampilkan layout Daftar Outlet	Sistem menampilkan menampilkan layout Daftar Outlet	Berhasil

3	Pengujian <i>Menu Logout</i>	Sistem akan menutup aplikasi	Sistem menutup aplikasi	Berhasil
4	Pengujian <i>List Daftar Outlet</i>	Sistem akan menampilkan layout Daftar Barang	Sistem menampilkan layout Daftar Barang	Berhasil
5	Pengujian <i>List Daftar Barang</i>	Sistem akan menampilkan layout Input Barang	Sistem menampilkan layout Input Barang	Berhasil
6	Pengujian <i>Send Input Barang</i>	Sistem akan menyimpan data	Sistem menyimpan data	Berhasil

V. KESIMPULAN

Secara umum terdapat beberapa kesimpulan dari penulisan penelitian ini, yaitu :

- Aplikasi berbasis android ini merupakan salah satu alternatif aplikasi yang dapat digunakan untuk mendata persediaan barang pada outlet.
- Dengan adanya aplikasi ini diharapkan dapat memudahkan proses pendataan barang sehingga aktifitas pendataan barang menjadi lebih efisien.

Berdasarkan hasil evaluasi penulisan ini, terdapat saran untuk pengguna sistem dan dalam pengembangan aplikasi ini secara lanjut antara lain adalah :

- Aplikasi Inventori Outlet ini berbasis Android ini harus diintegrasikan dengan persediaan barang pada gudang pemasok barang.
- Aplikasi ini dapat dikembangkan dengan menambahkan fitur penanda lokasi keberadaan pengguna untuk mengetahui posisi pengguna.

- [8] S. H. Nazruddin, *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Penerbit Informatika, 2012.
- [9] V. Yasin, *REKAYASA PERANGKAT LUNAK BERORIENTASI OBJEK Pemodelan, Arsitektur dan Perancangan (Modeling, Architecture and Design)*. Jakarta: Mitra Wacana Media, 2012.

REFERENSI

- [1] D. Wibowo and E. Luthfi, "Pembuatan Aplikasi Inventori Pada Toko Bangunan Wahyu Wonosari Yogyakarta Berbasis Android," *Data Manaj. dan Teknol. Inf.*, vol. 12, no. 4, p. 28, 2011.
- [2] A. Nugroho, A. Sunyoto, and K. Kunci, "PADA UD ADI JAYA BERBASIS ANDROID Hasil dan Pembahasan," vol. 12, no. 3, pp. 16–18, 2011.
- [3] R. A. Sukanto and Shalahuddin, *Rekayasa Perangkat Lunak*. Bandung: Informatika Bandung, 2013.
- [4] S. S. Hermawan, *Mudah Membuat Aplikasi Android*. Yogyakarta: CV Andi Offset, 2011.
- [5] A. Setyaputra and E. M. Aritonang, *Java for Beginners With eclipse 4.2 Juno*. Jakarta: PT. Elex Media Komputindo, 2012.
- [6] N. Ibrahim, "Pengembangan Aplikasi Semantic Web Untuk Membangun Web yang Lebih Cerdas," *J. Inform.*, vol. 3 No.1, pp. 27–39, 2007.
- [7] M. Sadeli, *Aplikasi Bisnis Dengan PHP dan MySQL*. Palembang, 2014.