Journal of Information System, Applied, Management, Accounting and Research
JISAMAR
ISSN: 2598-8719 (Online)
ISSN: 2598-8700 (Printed)
Vol. 2 No.4 November 2018

APLIKASI WAREHOUSE CONTROLING BERBASIS ANDROID

Astriana Mulyani[1], Dale[2],
Program Studi Teknik Informatika – STMIK NUSA MANDIRI JAKARTA[1]
Program Studi Teknik Informatika – STMIK NUSA MANDIRI JAKARTA[2]
[1]astriana.atm@nusamandiri.ac.id
[2]dalehasannudin89@gmail.com
,

Abstrak – *Warehouse* merupakan salah satu bagian penting sebuah industri penerbangan pengiriman jasa cargo, yang berfungsi sebagai tempat penyimpanan sementara baik *outgoing and incoming*. Dengan tingkat mobilitas pengiriman tinggi setiap harinya, meyebabkan banyak cargo keluar dan masuk gudang yang didata secara manual, untuk itu dibutuhkan suatu aplikasi management data untuk membantu mengelola data masuk dan keluar yang dilakukan secara sistem. Sehingga data tersebut terkelola dengan baik,secara *mobile*.Aplikasi ini dibangun secara *mobile* agar para pengguna aplikasi ini dapat menggunakannya kapanpun dan dimanapun (*moveable*). Aplikasi ini memanfaatkan android sebagai sistem operasinya. Karena saat ini android merupakan salah satu sistem operasi yang banyak digunakan di dalam *smartphone* dan android juga merupakan *open source* yang memungkinkan untuk di modifikasi dan di distribusikan secara bebas oleh pengembang sehingga hal tersebut yang menjadi alasan penulis dalam memilih android sebagai sistem operasi dalam pembuatan aplikasi *Warehouse Controling*. Aplikasi ini dapat dijalankan yaitu minimal pada sistem operasi android versi 4.2 (*Jelly Been).*

Kata Kunci: *Warehouse*, Teknologi, *Mobile*, Android

I. PENDAHULUAN

Warehouse dalam pengertian Gudang merupakan salah satu bagian penting sebuah industri penerbangan pengiriman jasa cargo, yang berfungsi sebagai tempat penyimpanan sementara baik cargo outgoing dan incoming. Jika dilihat dari segi fungsi dapat diketahui bahwa tingkat mobilitas cargo dalam gudang sangat tinggi setiap harinya, hampir terdapat puluhan bahkan ratusan keluar dan masuk gudang. hal inilah yang membuat kebutuhan akan sistem informasi database gudang menjadi sebuah wajib ada, sehingga nantinya keberadan cargo atau barang dalam gudang dapat terkolala dengan baik, secara mobile online setiap harinya.

Menurut (Athoillah, Muhamad dkk,2013)" Penggunaan teknologi informasi untuk kegiatan mengolah data pada berbagai aktivitas perusahan khususnya pada bagian gudang merupakan bagian penting bagi setiap perusahan, dengan teknologi ini data yang telah diolah dapat menjadi sebuah informasi yang cepat dan tepat. pada saat ini banyak perusahan yang masih menggunakan metode pencatatan manual untuk mendata barang yang masuk maupun keluar dari gudang".

Banyak permasalahan yang dihadapi dalam mendata barang masuk dan barang keluar diantaranya:

1. Semakin banyak jumlah pengiriman cargo melalui warehouse, mengakibatkan data cargo *outgoing dan incoming* kesulitan bila masih menerapkan cara manual.
2. Sulitnya pencarian rekapan atau *history data* cargo *outgoing dan incoming*, jika terjadi permasalahan."untuk mengetahui jumlah, berat cargo masuk gudang serta tanggal masuk dan keluar cargo tersebut".
3. Sulitnya pengontrolan data, khusunya untuk para pimpinan pekerja "melihat database cargo, melalui komputer diruangan", menyebabkan tidak efesien.

Melihat permasalahan diatas sehingga penulis tertarik untuk merancang sebuah aplikasi yang dapat membantu pekerja dalam menjalani aktivitas dalam bekerja. Aplikasi yang penulis buat ini merupakan sebuah aplikasi mengontrol serta mengelola data *outgoing* dan *incoming* cargo dasar pada perangkat *mobile* yang dibangun pada sistem operasi android. Android dipilih sebagai sistem operasi dalam pembuatan aplikasi ini dikarenakan pada saat ini android banyak digunakan pada *smartphone* dan juga android merupakan *open source* yang bebas untuk di modifikasi dan di distribusikan secara bebas oleh pengembang dan alasan membuat aplikasi ini secara *mobile* karena penulis berharap aplikasi ini dapat dipergunakan kapan saja dan dimana saja (*moveable*).

II. KAJIAN PUSTAKA

A. Warehouse

ISSN: 2598-8719 (Online)
ISSN: 2598-8700 (Printed)
Vol. 2 No.4 November 2018

(Triwobowo, Dodi dkk,2015) Menyimpulkan " dalam industry yang semakin perkembang, proses keluar masuknya barang perlu dicatat, ini diperlukan untuk

B. Konsep Dasar Program

(Sukamto dan M Shalahuddin,2015) Menyatakan "pemograman terstruktur adalah konsep atau paradigma atau sudut pandang pemograman yang membagi-bagi program berdasarkan fungsi-fungsi atau prosedur – prosedur yang dibutuhkan program komputer".

Sedangkan (Sianipar,2014) Mendefinisikan "program adalah kumpulan yang memuat satu atau lebih subprogram, yang dinamakan fungsi. beberapa fungsi yang terkenal dengan fungsi standar atau fungsi terpradefinisi adalah bagaian dari sistem".

Java

(Satyaputra dan Aritonang,2012) Menyimpulkan "Java merupakan sebuah *platform* sekaligus bahasa permrograman tingkat tinggi yang mempunyai kriteria sederhana, berorientasi objek, terdistribusi, dinamis, aman, dan lainnya".

(Supardi,2010) Menyatakan "perangkat lunak pemrograman Java merupakan bahasa pemrograman berorientasi objek karena seperti bahasa pengembangan C++, Java juga termasuk bahasa pemrograman PBO atau OOP (*Object Oriented Programming*) murni". Sedangkan menurut (Hariyanto,2011) "Java merupakan penyeimbangan antara *mazhab* orientasi objek murni yang memandang semua harus objek dan *mazhab* pragmatis yang menerapkan model pragmatis "*stay out my way*".

Eclipse

(Satyaputra dan Aritonang,2012) Menyimpulkan "Eclipse 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*". Eclipse yang diluncurkan oleh IBM pada tanggal 5 November 2001 telah mengeluarkan beberapa versi sejak awal peluncurannya.

Android

(Hidayat dkk,2011) Menyimpulkan "Android adalah system operasi untuk perangkat *mobile* yang pengembangannya dipimpin oleh Google".

Ada dua jenis distributor sistem operasi Android di dunia, yaitu Google *Mail Services* (GMS) yang didukung penuh oleh Google dan *Open Handset Distribution* (OHD) yang bebas dan tanpa dukungan langsung Google. Android yang disokong Google terintegrasi dengan layanan Gmail, Google *Calender*, Google *Contacts* dan Google *Voice*. Karena itu, begitu mengaktifkan android, langsung diminta memasukkan nama pengguna Google dan *password*.

C. Pengujian Sistem

Pengujian Whitebox

Menurut (Pressman,2010) "pengujian kotak putih, terkadang disebut juga pengujian kotak kaca (*glass box testing*), merupakan sebuah filosofi perancangan *test-case* yang menggunakan struktur kontrol yang dijelaskan sebagai bagian dari perancangan peringkat komponen untuk menghasilkan *test case*". Dengan mengggunakan metode pengujian kotak putih, menurut (Pressman, 2010) perekayasa sistem dapat melakukan *test case* yang :

1. Menjamin bahwa semua jalur independen di dalam modul telah dieksekusi setidaknya satu kali.
2. Melaksanakan semua keputusan logis pada sisi benar dan yang salah.
3. Melaksanakan semua *loop* pada batas mereka dan dalam batas-batas operasional mereka
4. Melakukan struktur data *internal* untuk memastikan ke*sahih*annya.

Pengujian Blackbox

(Pressman,2010) Menyatakan "pengujian kotak hitam, juga disebut pengujian perilaku, berfokus pada persyaratan fungsional perangkat lunak". Dengan demikian, teknik pengujian kotak hitam memungkinkan perakayasa perangkat lunak untuk membuat beberapa kumpulan kondisi masukan yang sepenuhnya akan melakukan semua kebutuhan fungsional untuk suatu program. Pengujian kotak hitam bukan teknik alternatif untuk teknik kotak putih. Sebaliknya ini merupakan pendekatan pelengkap yang mungkin dilakukan untuk mengungkap kelas kesalahan yang berbeda dari metode kotak putih. Sedangkan Menurut Ayuliana (2009:2) "pada *black box testing*, memfokuskan pada keperluan fungsional dari sofware ".

Menurut (Pressman,2010) pengujian kotak hitam berupaya untuk menemukan kesalahan dalam kategori berikut:

1. Fungsi yang salah atau hilang
2. Kesalahan antarmuka
3. Kesalahan dalam struktur data atau akses basis data eksternal
4. Kesalahan perilaku atau kinerja
5. Kesalahan inisialisasi dan penghentian.

Tidak seperti pengujian kotak putih yang dilakukan pada awal proses pengujian, pengujian kotak hitam cenderung diterapkan selama tahap-tahap pengujian selanjutnya. Karena pengujian kotak hitam sengaja mengabaikan struktur kendali, perhatian difokuskan pada ranah informasi.

D. UML (*Unified Modelling Languange*)

(Sukamto dan Shalahudin,2015) mendefinisikan "salah satu standar bahasa yang banyak digunakan di dunia industry untuk mendefinisikan reguirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemograman berorientasi objek.

ISSN: 2598-8719 (Online)
ISSN: 2598-8700 (Printed)
Vol. 2 No.4 November 2018

(Sukamto, Rosa A dan M.Shalahuddin,2015) UML mendefinisikan diagram-diagram sebagai berikut:

1. *Class Diagram*
   Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisikan kelas-kelas yang akan dibuat untuk membangun sistem.kelas memilki apa yang disebut atribut dan motode ata operasi.
2. *Object Diagram*
   Diagram objek menggambarkan struktur sistem dari segi penamaan obejek dan jalannya objek dalam sistem.Pada diagram objek harus dipastikan semua kelas yang sudah didefinisikan pada diagram kelas harus dipakai objeknya ,karna jika tidak,pendefinisikan kelas itu tidak dapat dipertanggungjawabakan.
3. *Component Diagram*
   Diagram komponen atau component *diagram* dibuat untuk menunjukan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem.
4. *Composite Structure Diagram*
   Composite structure diagram digunakan untuk mengambarkan struktur dari bagian-bagaian yang saling terhubung maupun mendeskripsikan struktur pada saat (berjalan) dari *instance* yang saling terhubung.
5. *Package* Diagram
   Package diagram menyediakan cara mengumpulkan elemen-elemen yang saling terkait dalam uml.
6. *Deployment* Diagram
   Deployment diagram juga menunjukan konfigurasi komponen dalam proses eksekusi aplikasi.
7. *Use Case* Diagram
   Use case atau *use case diagram merupakan permodelan untuk kelakukan (behavior)* sistem informasi yang akan dibuat. sebuah *intraksi* antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.
8. *Activity Diagram*
   Menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. mengelompokkan
9. *State Machine Diagram*
   State machine diagram atau statechart diagram diagram status digunakan untuk menggambarkan peubahan perubahan status atau transisi status dari sebuah mesin atau sistem atau objek.

E. ERD dan LRS

(Sukamto, Rosa A dan sahalahudin,2015) Menyimpulkan "bentuk paling awal dalam melakukan perancangan basis data relasional. memgambarkan hubungan antar entitas didalam basis data.

1.Entitas/ *entity*
merupakan simbol data inti yang akan disimpan, digunakan tabel pada basis data.

2. Atribut
merupakan simbol *field* atau kolom kebutuhan disimpan dalam suatu entitas.

3.Atribut kunci primer
merupakan simbol *field* atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses.

4.Atribut multinilai/*multivalue.*
merupakan simbol *field* atau kolom data yang disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.

5.*Relasi*
merupakan simbol relasi yang menghubungkan antar entitas,biasanya diawali dengan kata kerja.

6.Asosiasi /*association*
merupakan simbol penghubung antara relasi dan entitas dimana di kedua ujung memiliki *multiplicty* kemungkinan jumlah pemakaian.

Riyanto (2005:22) Mendefinisikan "LRS (*logical record structure*) adalah *representasi* dari struktur *record-record* pada tabel-tabel yang terbentuk dari hasil antar himpunan entitas".

Logical record strukture dibentuk dengan nomor dari tipe *record*.*Logical record strukture* terdiri dari *link-link* diantara tipe *record*. Link ini menunjukan arah dari satu tipe record lainnya. Banyak link dari LRS yang diberi tanda field-field yang kelihatan pada kedua link tipe record.

Penggambaran LRS mulai dengan menggunakan model yang dimengerti. Dua metode yang dapat digunakan, dimulai dengan hubungan kedua model yang dapat dikonversikan ke LRS.
Metode yang lain dimulai dengan ER-diagram dan langsung dikonversikan ke LRS.

1. *One to One* (1-1)
Tingkat hubungan ini menunjukkan hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, dan hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya.

2. *One to Many* (1-M)
Tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu, tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua. Sebaliknya, satu kejadian pada entitas yang kedua hanya dapat mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama.

ISSN: 2598-8719 (Online)
ISSN: 2598-8700 (Printed)
Vol. 2 No.4 November 2018

3. *Many to Many* (M-M)

Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya, dilihat dari sisi entitas yang pertama maupun dilihat dari sisi yang kedua.

III. PEMBAHASAN

Dalam pembahasan ini penulis membahas mulai analisa masalah sampai dibuatnya aplikasi warehose berbasis android.

A. Identifikasi Masalah

Tingkat Proses *mobilitas* penerimaan atau pengeluaran cargo dalam gudang sangat tinggi setiap harinya, hampir terdapat puluhan bahkan ratusan barang masuk atau keluar gudang. mulai dari pendataan pemasukan barang, pengecekan, *request*, *approval* barang, keadaan barang, sampai pengeluaran barang dari gudang dilakukan secara manual. Data fisik merupakan sesuatu yang mudah hilang, pengandaan data juga akan memberikan suatu *redundant* bagi seorang pekerja, dimana dimungkinkan adanya *human eror*, salah satu akibatnya adalah dapat menyebabkan suatu perbedaan data, dimana dari beberapa data yang ada dapat berbeda nilai. Oleh karena itu penulis merancang aplikasi *mobile Warehouse Controling* yang diharapkan dapat bermanfaat untuk mengelolah data *outgoing* atau *incoming* cargo di pergudangan, sehingga dapat lebih *efektif* dan *efisien* bagi pekerja.

B. Analisa Kebutuhan

1. Kebutuhan Perangkat Keras

Spesifikasi perangkat keras minimal pada telepon selular (smartphone) sehingga aplikasi MWarehouse ini dapat dijalankan yaitu:

a.Processor : Core I3
b.RAM : 2 GB
c. Internal storage (hardisk): 500 gb

2.Kebutuhan Perangkat Lunak

Spesifikasi perangkat lunak (software) pada telepon selular (smartphone) sehingga aplikasi MWarehouse ini dapat dijalankan yaitu minimal sistem operasi Android versi 4.2 (Jelly been).

1. .Kebutuhan Informasi

Dalam merancang aplikasi MWarehouse, harus berpedoman pada karateristik dan unsur yang terdapat pada aplikasi tersebut, yaitu :

a.Format

Program telah di format .apk sehingga pengguna tinggal melakukan penginstallan aplikasi MWarehouse pada telepon selular (smartphone) berbasis android.

b.Rules

Pada aplikasi MWarehouse ini terdapat tiga list yaitu list outgoing, incoming, history. Pada list outgoing, pengguna dapat menginput data cargo keluar gudang (tanggal,jam,jumlah cargo, berat cargo, comudity dan nama pengambil cargo tersebut). Pada list incoming pengguna dapat menginput data cargo yang masuk gudang (jumlah, berat, jenis, nama agent dan staff duty). Pada history pengguna dapat melihat rekapan data outgoing dan inoming dengan memasukan kode barang yang diinput. Pada list About pengguna dapat melihat informasi tentang aplikasi Mwarehouse dan pada button Keluar pengguna dapat memilih apakah ingin keluar atau tidak dari aplikasi MWarehouse.

c.Scenario

Scenario dalam aplikasi MWarehouse ini yaitu:

1.List outgoing

Pada list ini menampilkan kolum input data mengenai cargo atau barang yang keluar yang terdiri dari kode barang, nama agent, jumlah berat, jumlah koli, jenis barang, nama pengambil serta nama petugas atau staff duty saat itu. berdasarkan tanggal, jam keluar cargo tersebut.

2.List incoming

Pada list ini menampilkan kolum input data mengenai cargo atau barang masuk yang terdiri dari kode barang, nama agent, jumlah berat, jumlah koli, jenis barang, nama pengambil serta nama petugas atau staff duty saat itu. berdasarkan tanggal, jam masuk cargo tersebut.

3.List history

Pada list ini terdapat informasi rekapan data semua cargo atau barang yang keluar atau masuk dari gudang. proses rekapan dapat dilakukan dengan cara memasukan kode cargo atau barang pada kolum yang telah disediakan.

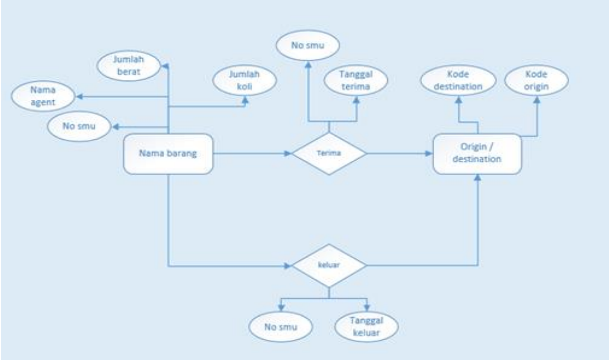
2.Kebutuhan Pengguna (user)

Dilihat dari sistem pengelolahan data keluar dan masuk cargo di gudang. Oleh karena itu penulis membuat aplikasi warehouse controling, alat bantu peninputan berupa data cargo keluar atau masuk yang khusus untuk membantu proses penginputan data, mengakses data lebih cepat berguna untuk mengetahui lokasi barang atau cargo tersebut. kemampuan pengguna dalam pendataan ,penginputan yang dirancang secara mobile dan diharapkan dapat menambah semangat para bekerja, sehingga proses bekerja dapat lebih efektif dan efisien.

C. Desain Database

1. *Entity Relationship Diagram* (*ERD*)

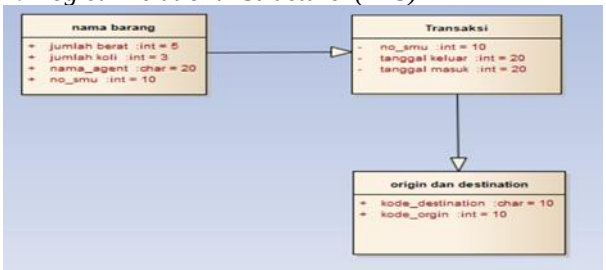
Konseptual database merupakan rincian dari *entity relationship diagram* (*ERD*) dimana terdapat attribute struktur file merupakan interpretasi dari *system* data yang digunakan sebagai media penyimpanan. Berikut ini database mengunakan *ERD* seperti yang dijelaskan dalam table berikut



Sumber: Penelitian (2017)

Gambar 1. Entity Relationship Diagram (ERD)

2. *Logical Relational Structure (LRS)*



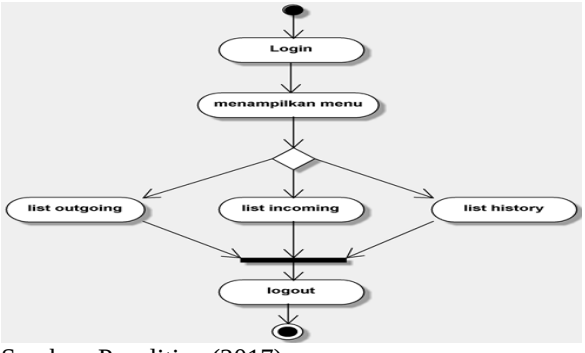
Sumber: Penelitian (2017)

Gambar 2. Logical Relational Structure (LRS)

D. Software Arsitektur

1.*Activity* Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir. *Activity* diagram mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity* diagram bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa.

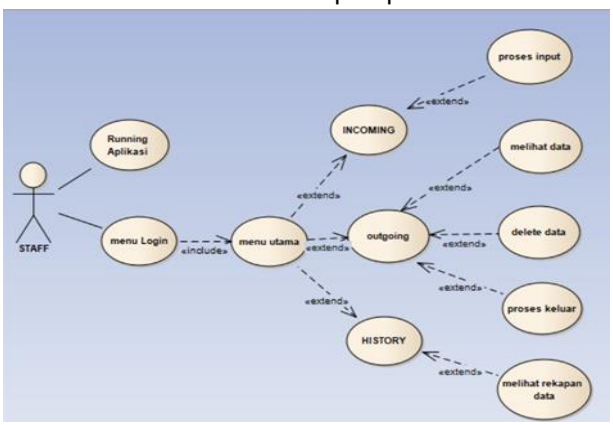


Sumber: Penelitian (2017)

Gambar 3. *Activity Diagram*

2. *Use Case* Diagram

Use case diagram merupakan deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use case* memberikan spesifikasi fungsi-fungsi yang ditawarkan oleh sistem dari perspektif *user*.

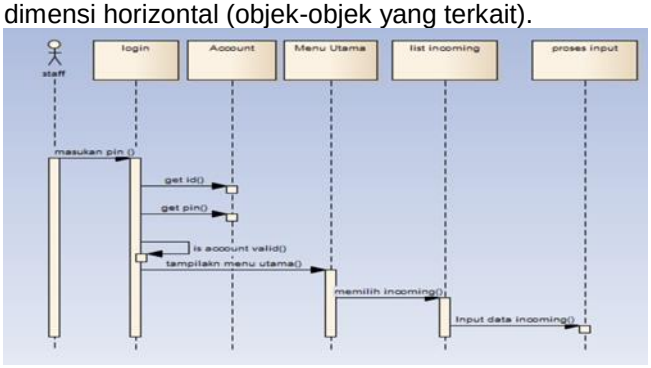


Sumber: Penelitian (2017)

Gambar 4. *Use Case* Diagram Mwarehouse

3. *Sequence* Diagram

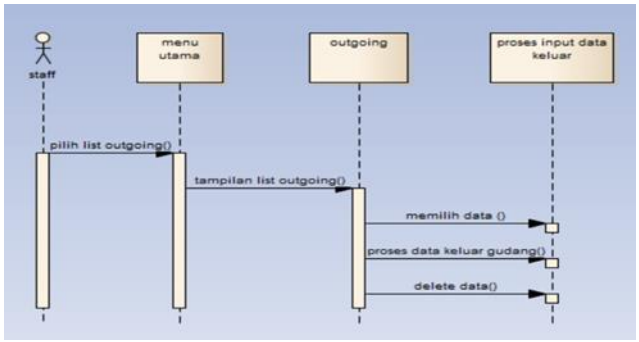
Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah *scenario* (termasuk pengguna, *display* dan sebagainya). Diagram ini menunjukkan sejumlah contoh objek dan *message* (pesan) yang diletakkan diantara obyek-obyek ini di dalam *use case*. *Sequence diagram* terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).



Sumber: Penelitian (2017)

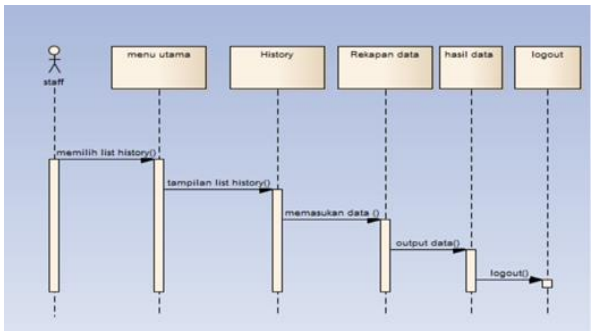
Gambar 5. *Sequence* Diagram *incominng*

Journal of Information System, Applied, Management, Accounting and Research
JISAMAR
ISSN: 2598-8719 (Online)
ISSN: 2598-8700 (Printed)
Vol. 2 No.4 November 2018



Sumber: Penelitian (2017)

Gambar 6. *Sequence* Diagram *outgoing*

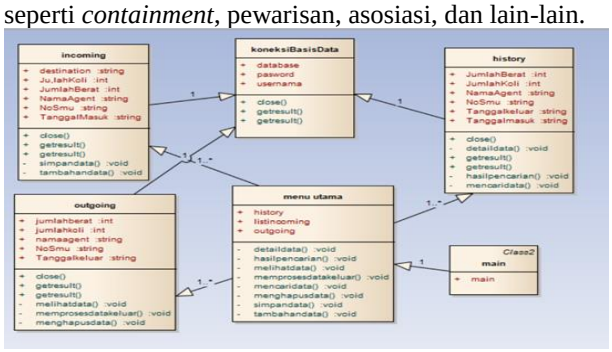


Sumber: Penelitian (2017)

Gambar 7. *Sequence* Diagram History

4. *Class Diagram*

Class menggambarkan keadaan (*atribut/properti*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

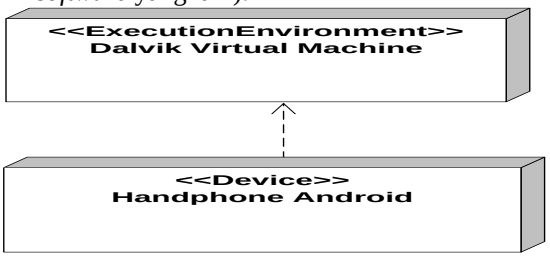


Sumber: Penelitian (2017)

Gambar 8. *Class* Diagram Mwarehouse

5. *Deployment* Diagram

Deployment diagram menyediakan gambaran bagaimana sistem secara fisik akan telihat. Sistem terdiri dari *node-node* dimana setiap *node* diwakili untuk sebuah kubus. Garis yang menghubungkan antara 2 kubus menunjukkan hubungan diantara kedua *node* tersebut. Tipe *node* bisa berupa *device* yang berwujud *hardware* dan bisa juga *processor* (yang mengeksekusi *component*) atau *execution environment* (*software* yang menjadi *host* atau mengandung *software* yang lain).



Sumber: Penelitian (2017)

Gambar 9. *Deployment* Diagram MWarehouse

E. User Interface

User interface merupakan interaksi antara sistem atau aplikasi dan pengguna. *User interface* sebisa mungkin dirancang untuk mudah dipakai agar pengguna dapat menggunakan aplikasi tersebut sekalipun pertama kalinya dalam menggunakan aplikasi. *User interface* yang tidak mudah dipahami akan mengakibatkan pengguna sulit dalam memakai aplikasi dan menyebabkan aplikasi tidak dapat digunakan dengan sempurna dan semestinya. *User interface* hendaknya dirancang semenarik mungkin agar pengguna dapat merasa nyaman dalam menggunakan aplikasi dan tidak mudah merasa bosan sehingga aplikasi dapat bermanfaat secara maksimal.

a. Login Awal Mwarehouse

Gambar di bawah merupakan *layout* (tampilan) login Mwarehouse

Sumber: Penelitian(2017)

Gambar 9. Antarmuka Awal login mwarehouse

b. Antarmuka Awal Mwarehouse

Gambar di bawah merupakan *layout* (tampilan) awal dari aplikasi *Mwarehouse*, yang terdiri dari tiga *list* (outgoing, incoming , dan history)

Journal of Information System, Applied, Management, Accounting and Research
ISSN: 2598-8719 (Online)
ISSN: 2598-8700 (Printed)
Vol. 2 No.4 November 2018

Sumber: Penelitian (2017)

Gambar 10. Antarmuka *List* menu

c. Antarmuka *Layout incoming*

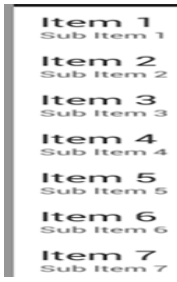
Pada antarmuka *layout incoming* menampilkan input data cargo masuk gudang dan terdapat satu *button* Kembali yang berfungsi untuk kembali ke menu awal.

Sumber: Penelitian (2017)

Gambar 11. Antarmuka *Layout* incoming

d. Antarmuka *List outgoing*

Pada *list* outgoing menampilkan hasil input data masuk yang akan diproses kembali untuk proses keluar cargo terdapat enam *list outgoing* (nama agent, no smu, jumlah berat, jumlah koli, origin tanggal keluar) yang jika diklik list proses akan diproses data keluar gudang dan satu *list* Kembali yang berfungsi untuk keluar dari menu Kalkulator dan kembali ke menu *Mwarehouse*



Sumber: Penelitian (2017)

Gambar 12. Antarmuka *List outgoing*

e. Antarmuka *Layout history*

Pada antarmuka *layout history* terdapat texbox dimana list input data perncarian,klik tombol proses menampilkan data detail rekapan cargo hasil dari pemilihan data tersebut, dipilih dan satu *list* Kembali yang berfungsi kembali menu awal dan *list* keluar berfungsi keluar dari program aplikasi Mwarehouse.

Sumber: Penelitian (2017)

Gambar 13. Antarmuka *Layout history*

F. Pengujian Sistem

Pengujian Whitebox

Pada pengujian *white box* yang penulis uji adalah perintah prosedural dari keseluruhan program secara utuh untuk menjamin operasi-operasi internal sesuai dengan spesifikasi yang telah ditetapkan dengan menggunakan struktur kendali dari prosedur yang dirancang. Pada aplikasi ini terdapat beberapa *sample* yang diuji, yaitu:

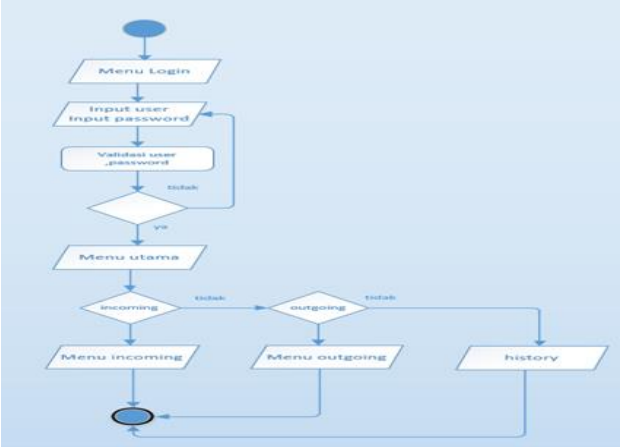
1. Pengujian *White Box* Pada Aplikasi *Mwarehouse (mobile warehouse)*

Secara garis besar, algoritma dari aplikasi *Mwarehouse* yaitu:

1. Memilih list *incoming*
2. Memilih list *outgoing*
3. Memilih list *history*

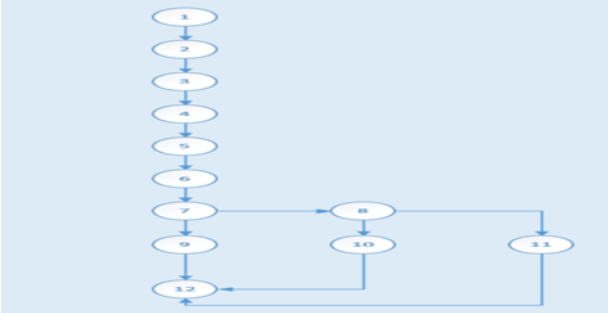
Flowchart dari aplikasi *Mwarehouse List incoming* adalah:

Journal of Information System, Applied, Management, Accounting and Research
ISSN: 2598-8719 (Online)
ISSN: 2598-8700 (Printed)
Vol. 2 No.4 November 2018



Sumber: Penelitian(2017)

Gambar 14. *Flowchart* Aplikasi *Mwarehouse List incoming*



Sumber: Penelitian (2017)

Gambar 14. Grafik Alir Aplikasi*Mwarehouse*

Kompleksitas sisklomatis (*cyclomatic complexity*) adalah metrik *software* yang menyediakan ukuran kuantitatif dari kekompleksan logikal program. Apabila digunakan dalam konteks metode uji coba basis *path*, nilai yang dihitung untuk *cyclomatic complexity* menentukan jumlah jalur *independent* dalam basis *set* suatu program dan memberi batas atas untuk jumlah uji coba yang harus dikerjakan untuk menjamin bahwa seluruh perintah sekurang-kurangnya telah dikerjakan sekali.

Jalur *independent* adalah jalur yang melintasi atau melalui program dimana sekurang-kurangnya terdapat proses perintah yang baru atau kondisi yang baru. *Cyclomatic complexity* digunakan untuk mencari jumlah *path* dalam satu *flowgraph*. Dapat diperoleh dengan perhitungan:

$$V(G) = E - N + 2$$

Penjelasan:

E = Jumlah *edge* grafik alir yang ditandakan dengan gambar panah.

N = Jumlah simpul grafik alir yang ditandakan dengan gambar lingkaran.

Sehingga kompleksitas siklomatisnya adalah:

V(G) = 13 – 12 + 2 = 3

Basis *set* yang dihasilkan dari jalur *independent* secara linier adalah sebagai berikut:

Path 1= 1 – 2 – 3 – 4 – 5– 6–7–9–12

Path 2= 1 – 2 – 3 – 4 – 5–6–7–8–10–12

Path 3= 1 – 2 – 3 – 4–5–6–7–8–11–12

Ketika aplikasi dijalankan, maka terlihat bahwa salah satu basis set path 1 yang dihasilkan adalah 1 – 2 – 3 – 4 – 5 – 7–9 –12 dan terlihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut dari segi kelayakan *software*, sistem telah memenuhi syarat.

Ketika aplikasi dijalankan, maka terlihat bahwa salah satu basis set path 2 yang dihasilkan adalah 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 12 dan terlihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut dari segi kelayakan *software*, sistem telah memenuhi syarat.

Ketika aplikasi dijalankan, maka terlihat bahwa salah satu basis set path 3 yang dihasilkan adalah 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 11 – 12 dan terlihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut dari segi kelayakan *software*, sistem telah memenuhi syarat.

Ketika aplikasi dijalankan, maka terlihat bahwa salah satu basis set path 4 yang dihasilkan adalah 14-2dan terlihat bahwa simpul telah dieksekusi satu kali. Berdasarkan ketentuan tersebut dari segi kelayakan *software*, sistem telah memenuhi syarat.

IV. KESIMPULAN

Berdasarkan hasil dari analisa, implementasi, dan pengujian terhadap aplikasi mobile warehouse (MP) maka dapat diambil kesimpulan sebagai berikut:

1. Penggunaaan aplikasi mobile warehouse (Mwarehouse) sangat praktis dan bermanfaat karena dapat memgelola atau memagement data cargo incoming dan outgoing kapan saja dan dimana saja cukup dengan menggunakan smartphone berbasis sistem operasi android.
2. Terdapat fitur menu warehouse controlling yang terdiri dari tiga pembahasan incoming,outgoing dan history.
3. Terdapat fitur menu history yang berguna untuk membantu pengguna dalam mencari data rekapan cargo.
4. Aplikasi mobile warehouse (MWarehouse) dirancang dengan menggunakan teknologi Java dengan Eclipse Indogo versi Indigo Service Release 2 sebagai perangkat lunak dan hanya dapat diimplementasikan

ISSN: 2598-8719 (Online)
ISSN: 2598-8700 (Printed)
Vol. 2 No.4 November 2018

pada ponsel (smartphone) yang mendukung sistem operasi Android dengan minimal OS versi Jelly Been.

REFERENSI

[1]Athoilah, Muhamad, M. Isa. Irawan. 2013.Perancangan Sistem Informasi *Mobile* Berbasis Android Untuk Kontrol Persedian Barang Di Gudang. Surabaya : Jurnal Sains dan Seni POMITS Vol. 1, No.1, (2013) 1-6.

[2]Hariyanto, Bambang. 2011. Esensi-esensi Bahasa Pemrograman Java. Bandung: Informatika.

[3]Hidayat, Wicak dan Sudarma S. 2011. Buku Pintar Komputer Laptop Netbook dan Tablet iPad dan Android. Jakarta: Mediakita.

[4]Pressman, Roger S. 2010. Rekayasa Perangkat Lunak (Buku Satu). Yogyakarta: Andi.

[5]Satyaputra, Alfa dan Eva Maulina Aritonang. 2012. *Java for Beginners with* Eclipse 4.2 Juno. Jakarta: Elex Media Komputindo.

[6]Sinipar. 2014. Pemograman C++. Bandung: Informatika.

[7]Sukamto, Rosa ariani,M shalahudin. 2015. Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek. Bandung: Informatika.

[8]Supardi, Yuniar. 2010. Semua Bisa Menjadi *Programmer* Java *Basic Programming*. Jakarta: Elex Media Komputindo.

[9]Nugroho, Arif, Andi Sunyoto. 2011. Pembuatan Aplikasi Persedian Barang Pada UD Adi Jaya Berbasis Android. ISSN: 1411 – 3201. Yogyakarta: Jurnal DASI (2011) Vol. 12 No. 3

[10]Triwibowo, Dodi, Rinta Kridalukmana, Kurniawan Teguh Martono. 2015.Pembuatan Aplikasi Terintegrasi, Pendataan Barang di Gudang Berbasis Android ISSN: 2338-0403. Semarang: Jurnal Teknologi dan Sistem Komputer. Vol 3, no 2, april 2015. Semarang.

[11] Verdi Yasin. 2012 Rekayasa Perangkat Lunak Berorientasi Objek, Penerbit; Mitra Wacana Media, Jakarta.

[12] Muhammad Iqbal dan Relita Buaton dan Verdi Yasin.2017. 15 Metode Konsep aplikasi Cerdas, Penerbit; Fakultas Teknik Universitas Pembangunan Panca Budi, Medan, Sumatera Utara.