

BAB IV

RANCANGAN SISTEM DAN PROGRAM USULAN

4.1 Analisa Kebutuhan Software

A. Tahapan analisis

Dalam proses stock opname yang sudah ada sudah tersistem namun masih tidak efektif dalam perhitungan data dikarenakan proses perhitungan data tersebut harus menyatukan data dari masing-masing laptop user yang dipakai untuk stock opname dan karena menggunakan laptop, proses stock opname menjadi kurang efisien waktu.

Guna mengefektifkan proses stock opname di PT. Arie Muti, dalam proses input data stock opname dan perhitungan data stock opname, maka dibangun sebuah aplikasi untuk menginput dan menghitung data stock opname berbasis mobile, agar lebih fleksibel, efektif dan efisien dalam melakukan stock opname. Sehingga stock opname yang saat ini dilakukan per semester atau 6 (enam) bulan sekali diharapkan dengan adanya aplikasi ini dapat dilakukan minimal seminggu sekali. Berikut ini spesifikasi kebutuhan (*system requitment*) dari sistem perhitungan stock opname :

Halaman *User*:

A1. *User* dapat melakukan *login* dengan data *username* dan *password*.

A2. *User* dapat memasukan data *opname*.

A3. *User* dapat melihat daftar barang.

Halaman *Korektor*:

B1. *Korektor* dapat melakukan *login* dengan data *username* dan *password*.

B2. *Korektor* dapat memasukan data koreksi.

B3. *Korektor* dapat melihat daftar barang

Halaman *Administrator*:

C1. *Admin* dapat melakukan *login* dengan data *username* dan *password*.

C3. *Admin* dapat mengelola data barang.

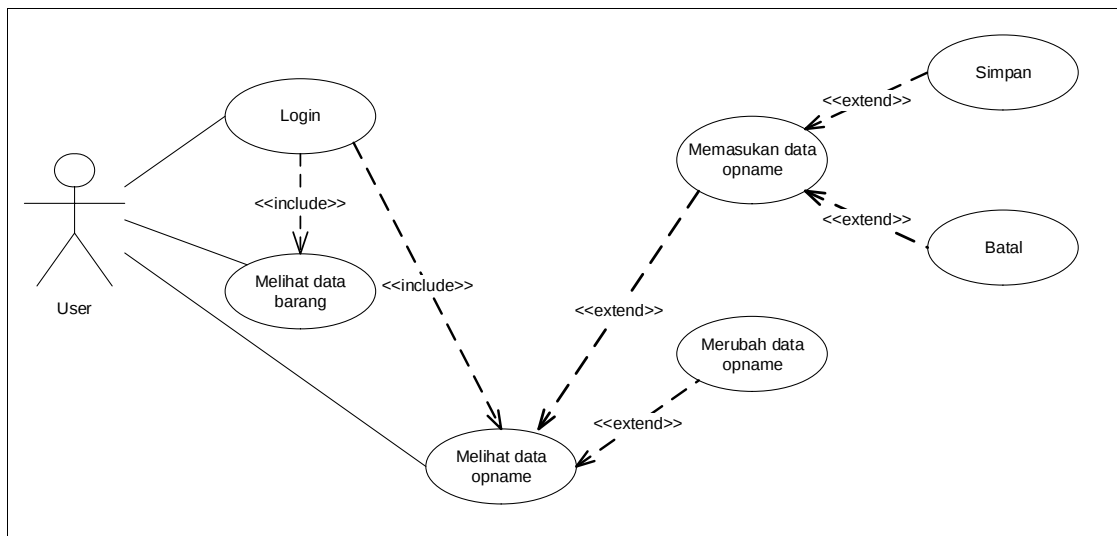
C4. *Admin* dapat mengelola data *opname*.

C5. *Admin* dapat mengelola data koreksi.

C6. *Admin* dapat mengelola laporan *stock opname*.

B. Use Case Diagram

1. Use Case Diagram Pemasukan Data Stock Opname Halaman User



Gambar IV.1

Use case diagram Pemasukan Data Stock Opname halaman User

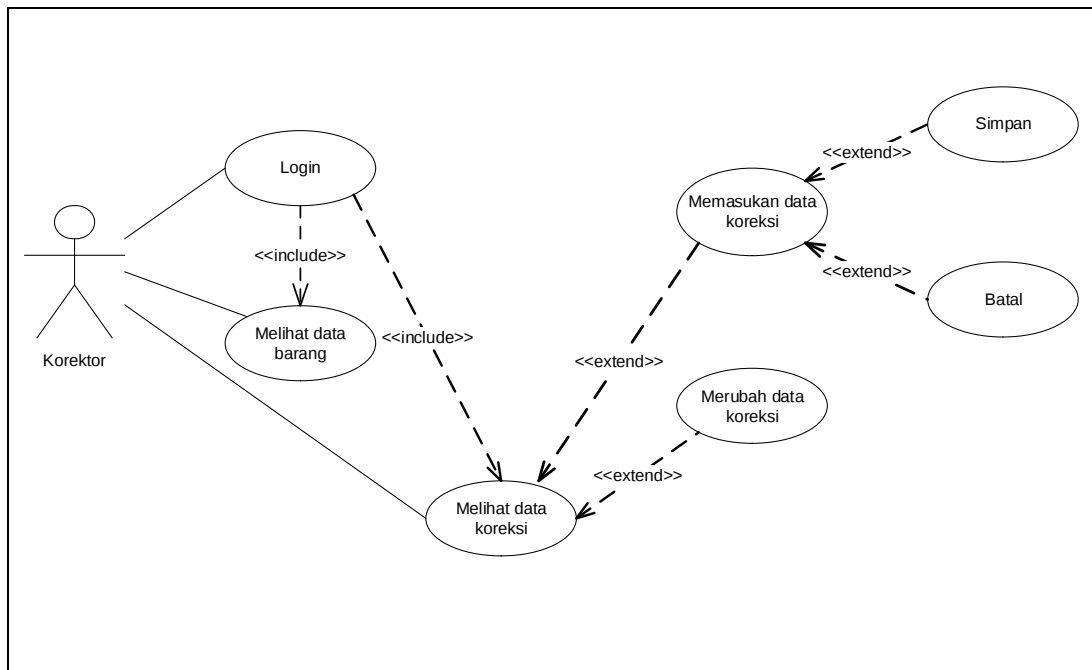
Deskripsi *Use Case Diagram* Pemasukan Data Stock Opname Halaman User.

Tabel IV.1

Deskripsi *Use Case Diagram* Pemasukan Data Stock Opname Halaman User

<i>Use case Name</i>	Penginputan Data Stock Opname
<i>Requiretments</i>	A1 – A3
<i>Goal</i>	User dapat memasukan data Opname melalui aplikasi Mobile
<i>Pre-conditions</i>	User telah login
<i>Post-conditions</i>	User melakukan stock opname
<i>Failed end condition</i>	Kesalahan pemasukan data opname
<i>Primary Actors</i>	Staf kantor
<i>Main Flow / basic path</i>	1. User memasukan Username dan Password 2. User melakukan scan barcode 3. User memasukan data opname 4. User memasukan stock in 5. User memasukan stock out 6. User menyimpan data opname 7. User melihat daftar data opname
<i>Alternate flow / Invariant 1</i>	7a. User mengedit data opname

2. Use Case Diagram Pemasukan Data Koreksi Halaman Korektor



Gambar IV.2

Use Case Diagram Pemasukan Data koreksi Halaman Korektor

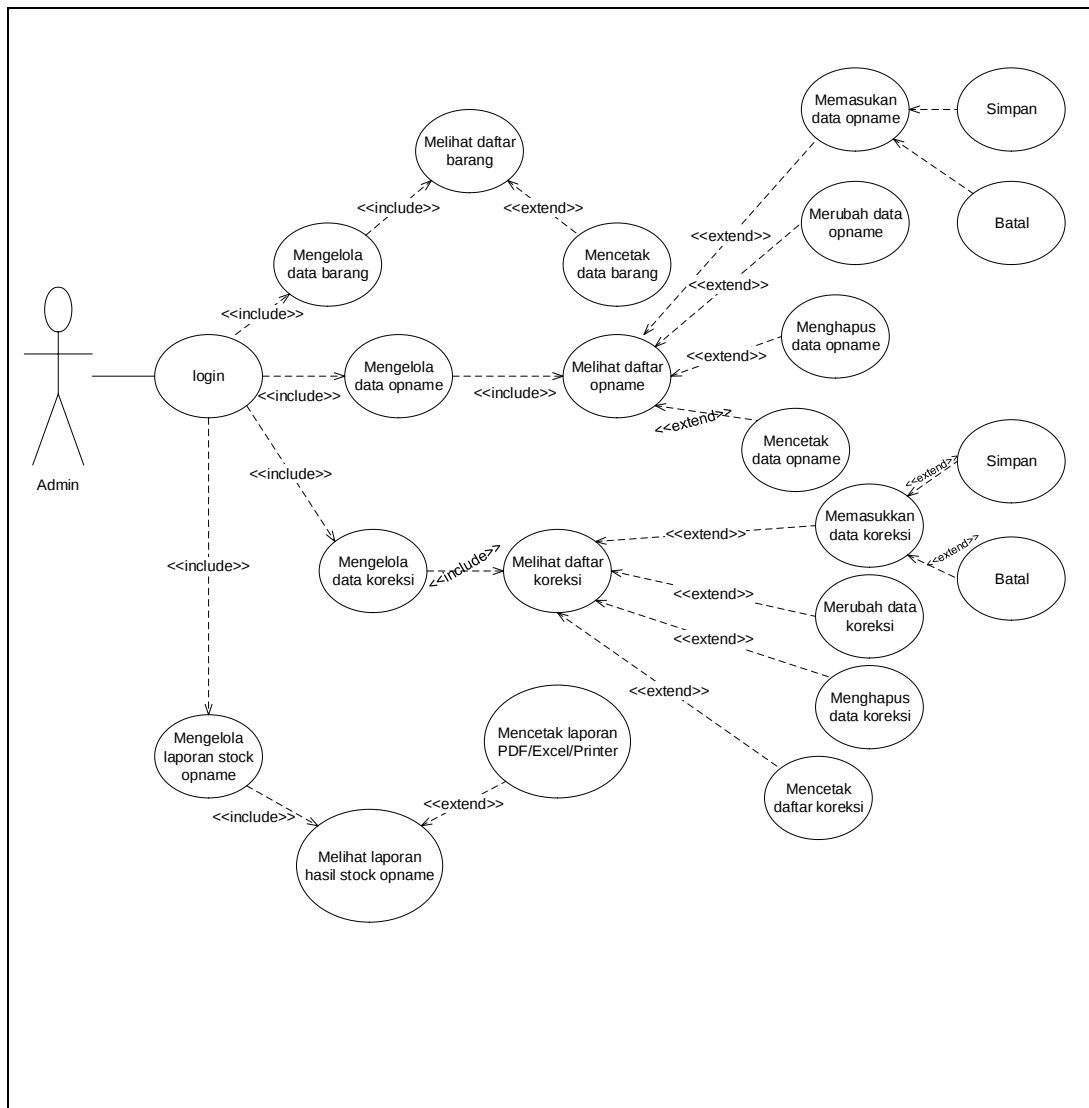
Deskripsi *Use Case Diagram* Pemasukan Data Koreksi Halaman Korektor:

Tabel IV.2

Deskripsi *Use Case Diagram* Pemasukan Data Koreksi Halaman Korektor

<i>Use case Name</i>	Mengelola Data Koreksi
<i>Requiretments</i>	B1 – B3
<i>Goal</i>	Korektor dapat memasukan data koreksi melalui aplikasi <i>mobile</i>
<i>Pre-conditions</i>	Korektor telah <i>login</i>
<i>Post-conditions</i>	Korektor melakukan koreksi
<i>Failed end condition</i>	Kesalahan pemasukan data koreksi
<i>Primary Actors</i>	Staf IT
<i>Main Flow / basic path</i>	1. Korektor melakukan <i>login</i> 2. Korektor memasukan angka <i>barcode</i> 3. Korektor memasukan data koreksi 4. Korektor menyimpan data koreksi 5. Korektor dapat melihat daftar koreksi
<i>Alternate flow / Invariant 1</i>	5a. Korektor mengedit data koreksi

3. Use case diagram Menyimpan Data Akhir Stock Opname Halaman Admin



Gambar IV.3

Use Case Diagram Mengetahui Data Akhir Stock opname Halaman Admin

Deskripsi *Use Case Diagram* Mengetahui Data Akhir *Stock Opname* Halaman Admin .

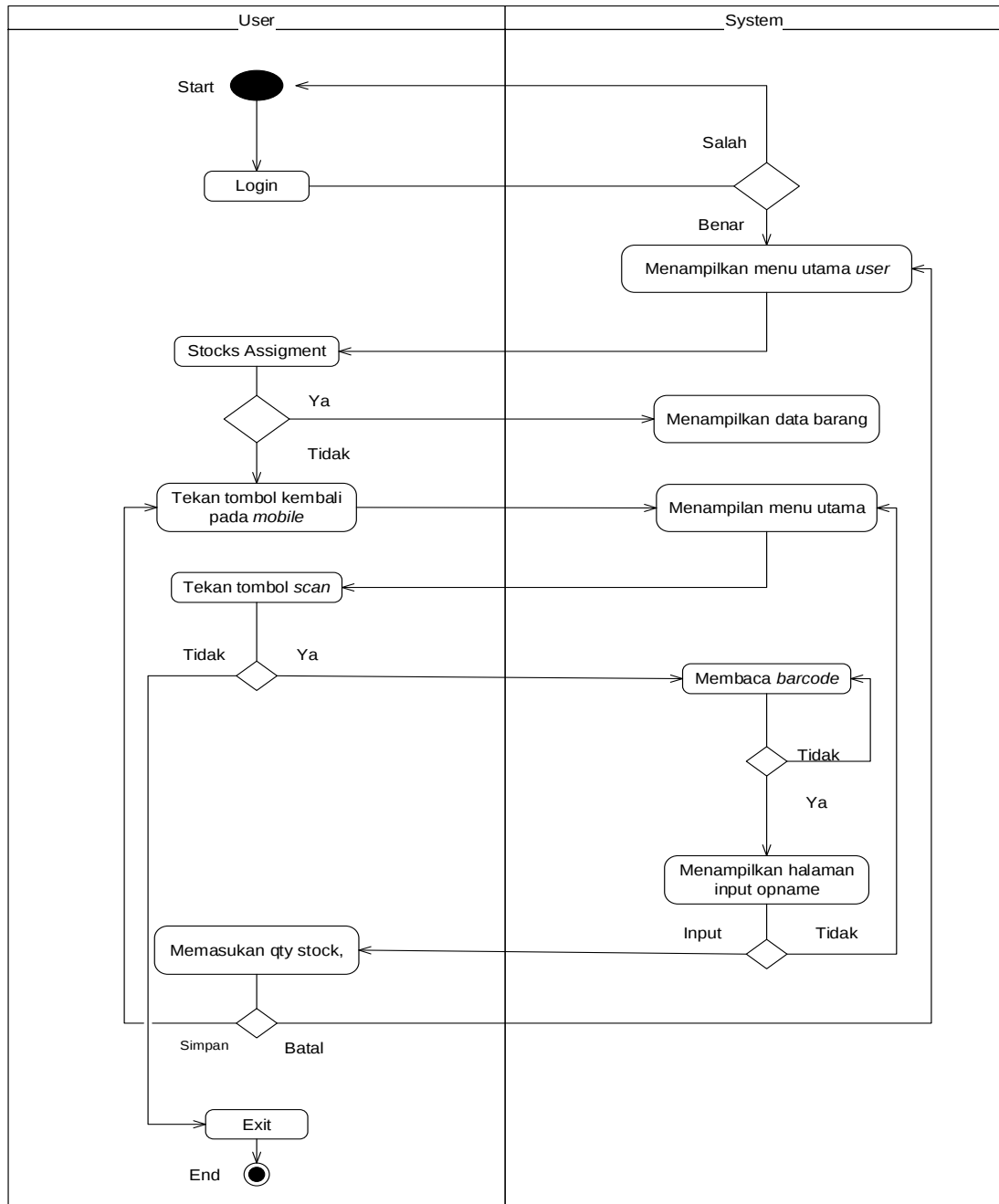
Tabel IV.3

Deskripsi *Use Case Diagram* Mengetahui Data Akhir *Stock Opname* Halaman Admin

<i>Use case Name</i>	Mengelola Data Akhir
<i>Requiretments</i>	C1 – C6
<i>Goal</i>	Admin mengetahui data akhir <i>Stock Opname</i>
<i>Pre-conditions</i>	Admin telah <i>login</i>
<i>Post-conditions</i>	Admin menarik dan mencetak data akhir <i>Stock opname</i>
<i>Failed end condition</i>	Printer Rusak
<i>Primary Actors</i>	Administrator / Operasional Manager
<i>Main Flow / basic path</i>	1. Admin melakukan <i>login</i> 2. Admin menarik data <i>opname</i> 3. Admin menarik data koreksi 4. Admin menarik data akhir <i>stock opname</i>
Alternate flow / Invariant 1	2a. Admin mencetak data <i>opname</i>
Invariant 2	3a. Admin mencetak data koreksi
Invariant 3	4a. Admin mencetak data akhir <i>stock opname</i>

C. Activity Diagram

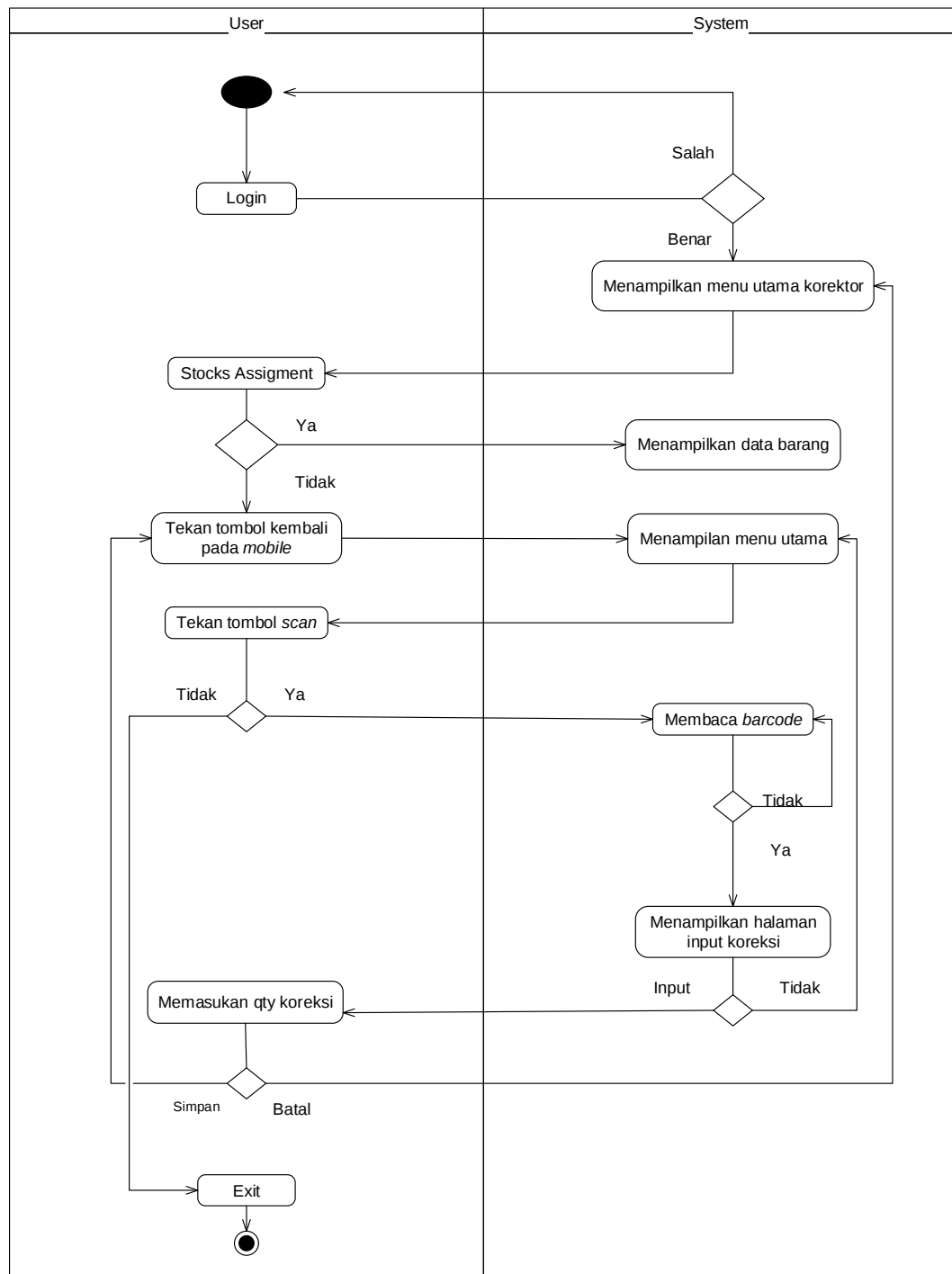
1. Activity Diagram Memasukan Data Stock halaman User



Gambar IV.4

Activity Diagram Memasukan Data Stock Halaman User

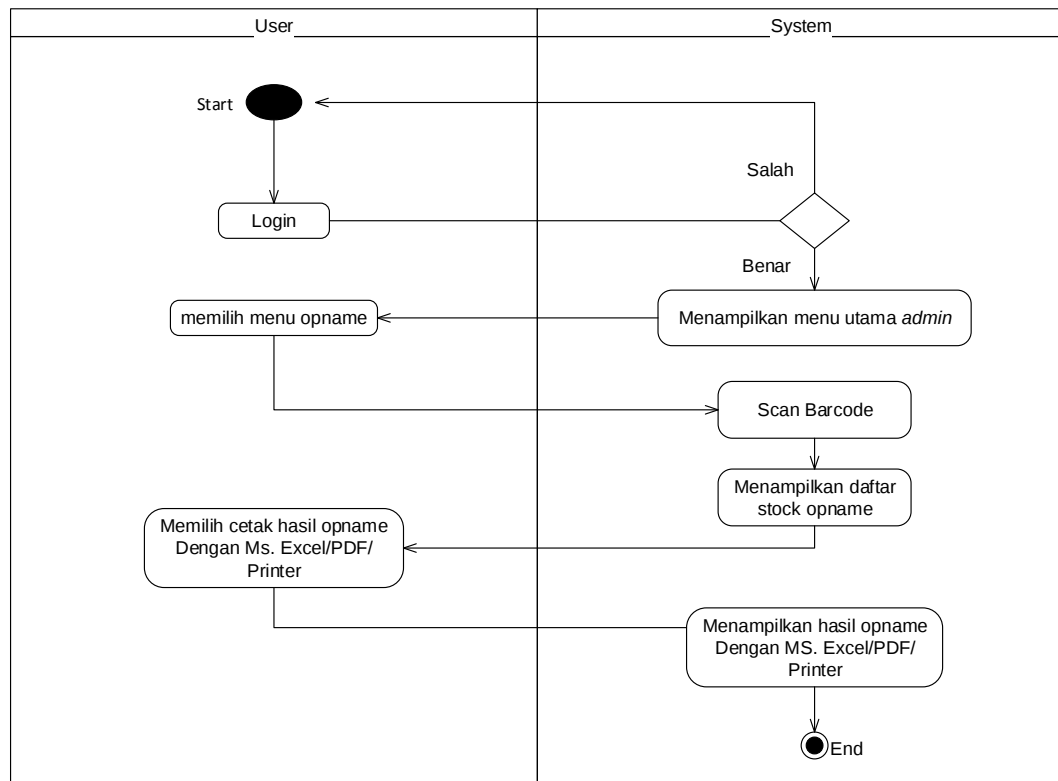
2. Activity Diagram Memasukkan Data Koreksi Halaman Korektor



Gambar IV.5

Activity Diagram Memasukkan Data Koreksi Halaman Korektor

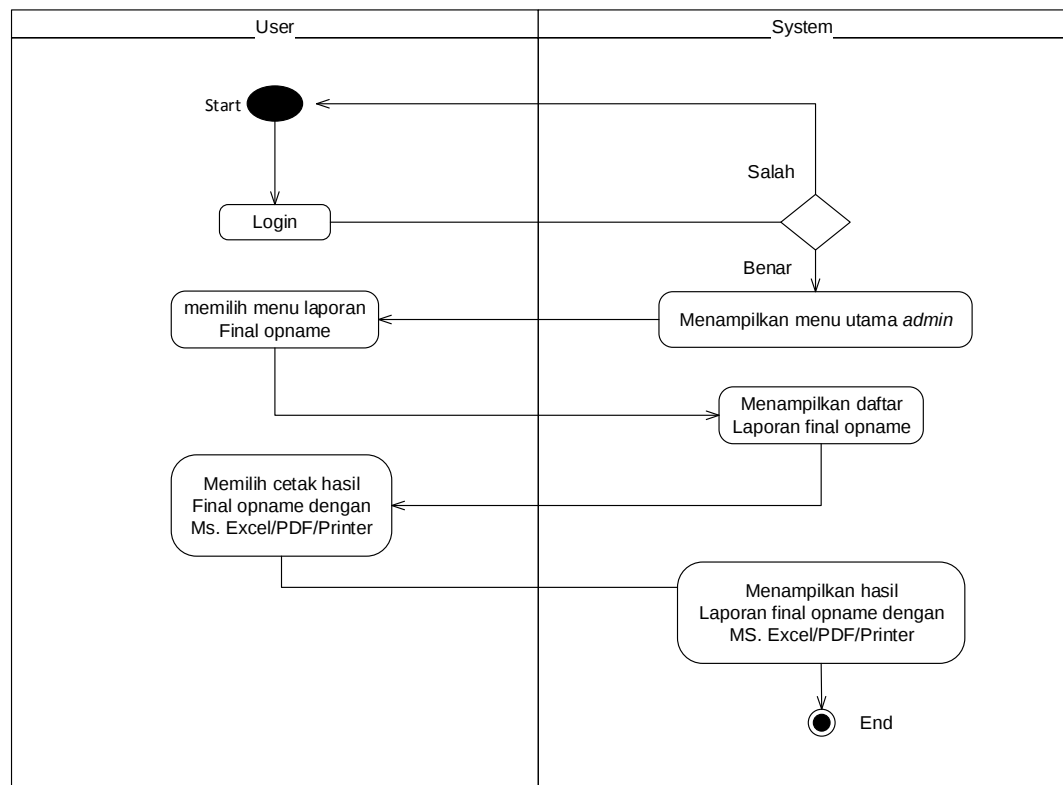
3. Activity Diagram Mengelola Laporan Hasil Stock Opname Halaman Admin



Gambar IV.6

Activity Diagram Mengelola Laporan Hasil Stock Opname Halaman Admin

5. Activity Diagram Mengelola hasil Data Final Halaman Admin

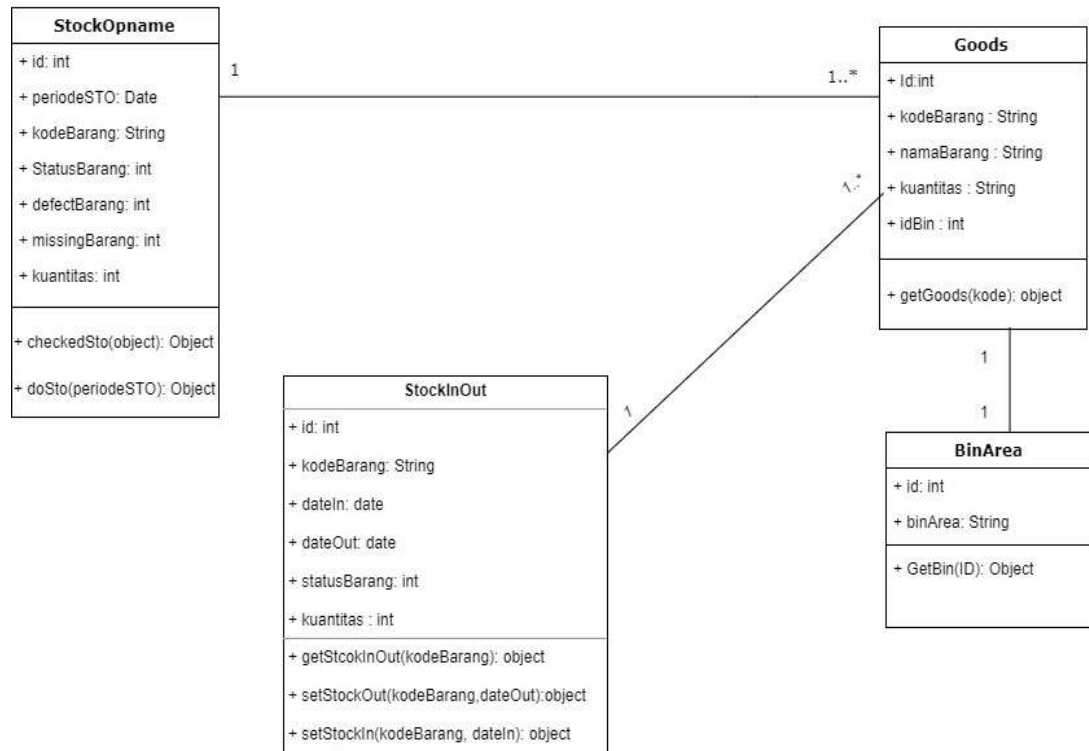


Gambar IV. 8
Activity Diagram Mengelola Hasil Data Final Halaman Admin

4.2 Desain

4.2.1 Database

1. Physical data model

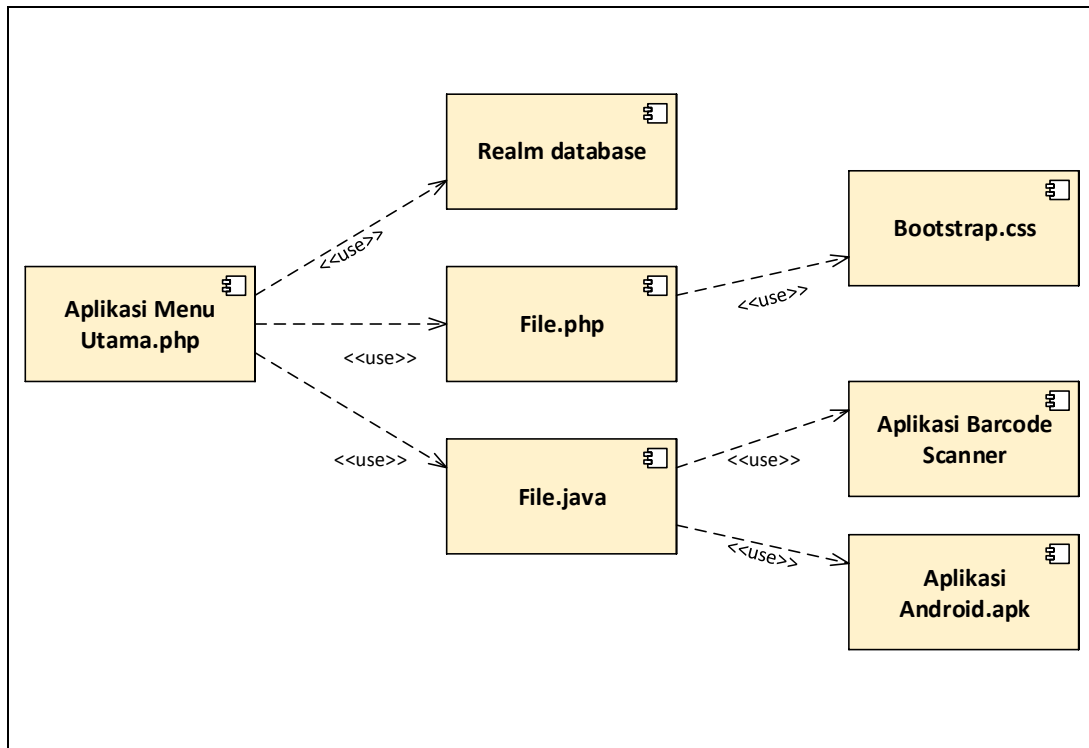


Gambar IV.9

Physical Data Model sistem pengelolaan data Stock Opname

4.2.2 Software Architecture

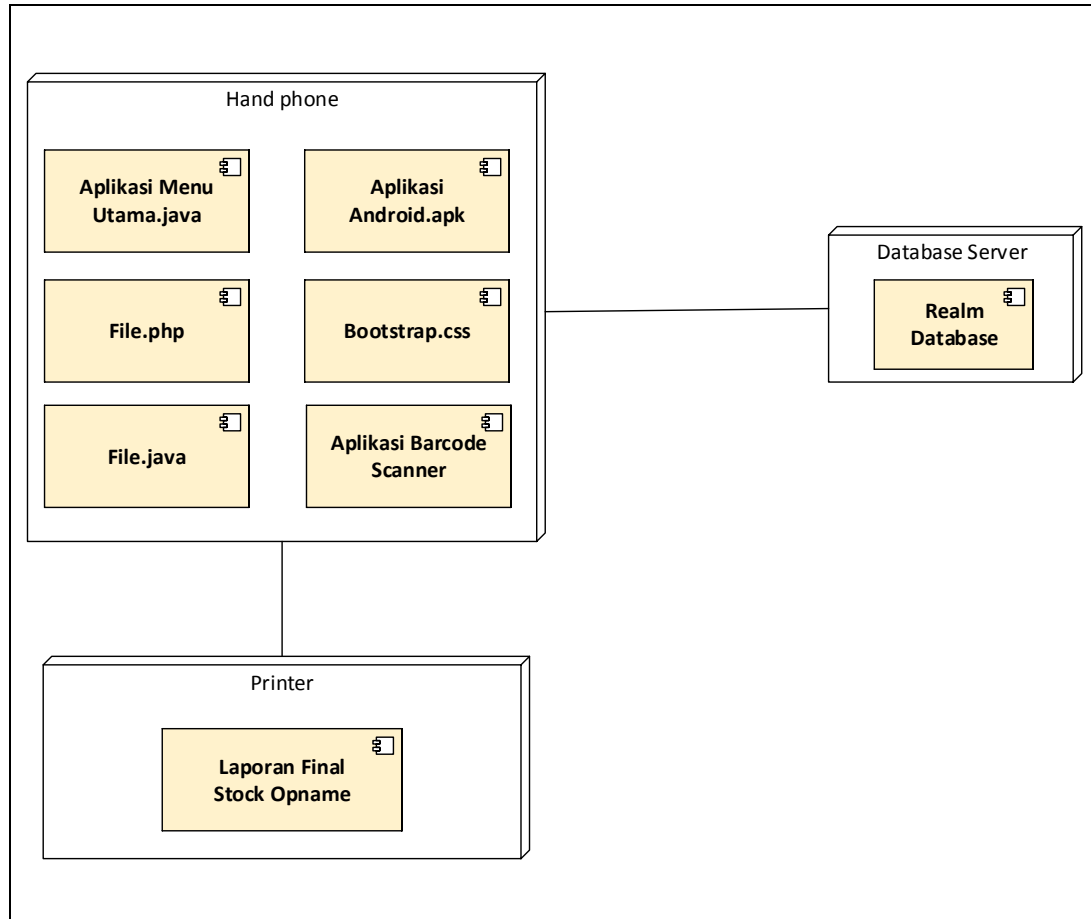
A. Component Diagram



Gambar IV.10

Component Diagram Pengelolaan Data Stock Opname

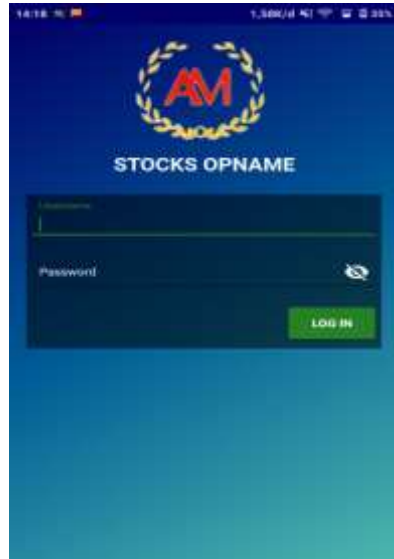
B. Deployment Diagram



Gambar IV.11

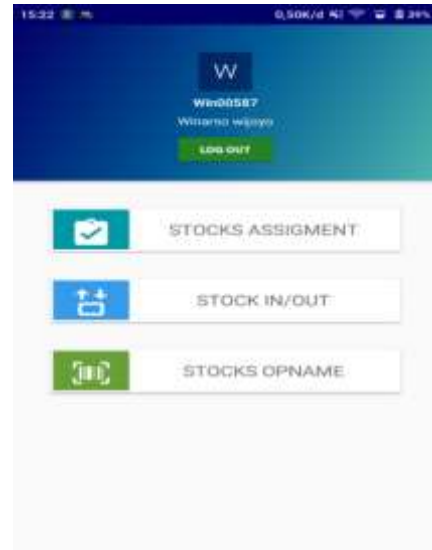
Deployment Diagram Pengelolaan Data Stock Opname

4.2.3 User Interface



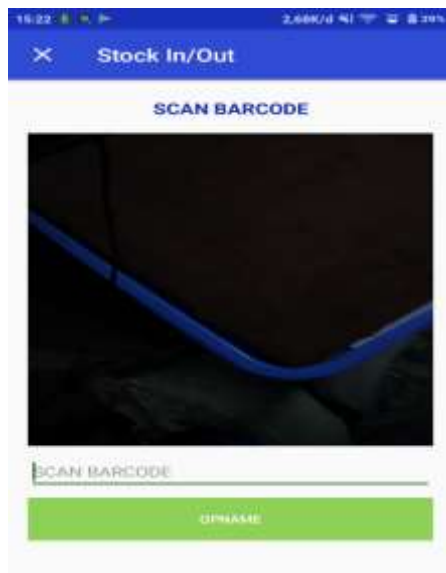
Gambar IV.12

Tampilan menu *login*



Gambar IV.13

Tampilan Menu setelah *login*



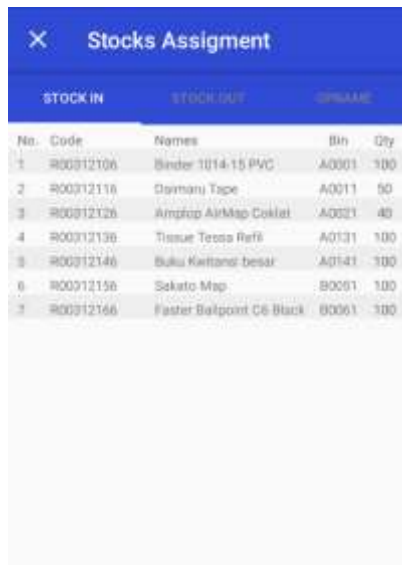
Gambar IV.14

Tampilan Scan barcode *Stock in/out*



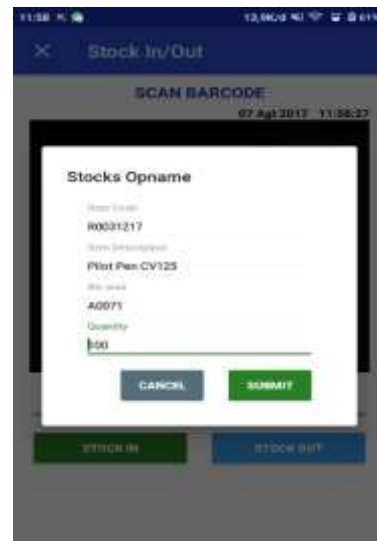
Gambar IV.15

Tampilan scan barocde *stock opname*



No.	Code	Names	Bin	Qty
1	R00312106	Binder 1014-15 PVC	A0001	100
2	R00312116	Garmanj Tape	A0011	50
3	R00312126	Amplop AirMap Coklat	A0021	40
4	R00312136	Tissue Tessa Refil	A0131	100
5	R00312146	Buku Kertanar besar	A0141	100
6	R00312156	Sekato Map	B0061	100
7	R00312166	Faster Ballpoint C6 Black	B0061	100

Gambar IV.16

Tampilan menu *Stock Assignment*


Stocks Opname

Item Code: R0031217

Item Description: Pilot Pen CV125

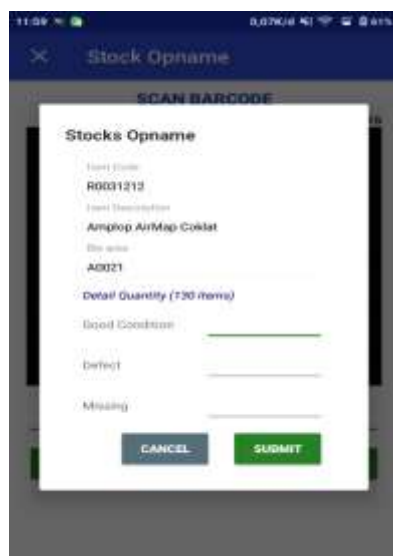
Bin Code: A0071

Quantity: 100

CANCEL SUBMIT

STOCK IN STOCK OUT

Gambar IV.17

Tampilan setelah scan barcode *Stock in/out*


Stocks Opname

Item Code: R0031213

Item Description: Amplop AirMap Coklat

Bin Code: A0021

Detail Quantity (720 items)

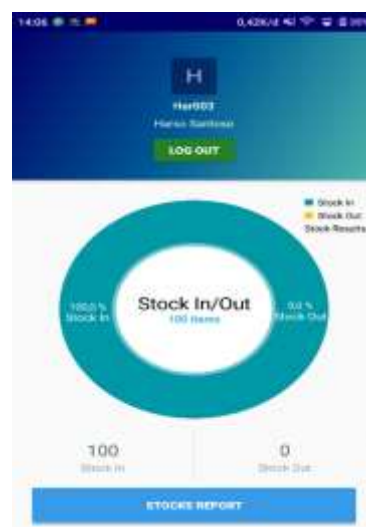
Good Condition

Defect

Missing

CANCEL SUBMIT

Gambar IV.18

Tampilan untuk melihat *stock opname*

Gambar IV.19

Tampilan setelah *login* Admin



No	Code	Names	Bin	Stock	Qty In	Qty Out	Defect	Missing	Total Qty	Update date
1.	R0031210	Binder 1015-15 PVC	A0001	100	100	100	100	0	0	14/08/2017
2.	R00312113	Tissue Tessa Kotak	A00102	80	40	20	0	5	95	14/08/2017
3.	R00312114	Pencil 2B Castle	A00126	1500	0	250	50	0	1800	14/08/2017
4.	R00312115	Stabilo Swan	A00127	2000	0	350	100	0	1550	14/08/2017
5.	R0031211	Daimaru Tape	A0011	500	0	0	0	0	500	01/08/2017
6.	R0031212	Amplop AirMap	A0021	1000	0	200	0	0	800	01/08/2017
7.	R0031213	Tissue Tessa Refil	A00131	1000	0	0	0	0	1000	01/08/2017
8.	R0031214	Buku Kwitansi Besar	A0141	500	50	0	0	0	550	01/08/2017
9.	R0031215	Sakato Map	B0051	500	500	0	0	0	1000	01/08/2017
10.	R0031216	Faster Ballpoint C6	B0061	900	100	0	0	0	1000	01/08/2017
11.	R0031217	Pilot Pen CV125	A0071	900	100	0	0	0	1000	01/08/2017
12.	R0031218	Binder 1014-15 PVC	A0001	500	0	0	0	0	500	01/08/2017

Gambar IV.20
Tampilan *Final Stock Opname* halaman admin *stocks report*

4.3 *Code Generation*

A. **Login**

```
package com.arimurti.sto.ui.activity;

import android.content.Intent;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.text.TextUtils;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;


import com.arimurti.sto.R;

import com.arimurti.sto.modelHelpers.RealmBaseHelper;

import com.arimurti.sto.modelHelpers.UserHelpers;

import com.arimurti.sto.models.UserProfile;

import com.arimurti.sto.utils.Helpers;


import io.realm.Realm;


public class LoginActivity extends AppCompatActivity {
```

```
private EditText etUsernameLogin;
```

```
private EditText etPasswordLogin;
```

```
private Button btnLogin;
```

```
private Realm realm;
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_login);
```

```
    etPasswordLogin = findViewById(R.id.etPasswordLogin);
```

```
    etUsernameLogin = findViewById(R.id.etUsernameLogin);
```

```
    btnLogin = findViewById(R.id.btnLogin);
```

```
    realm = Realm.getDefaultInstance();
```

```
    btnLogin.setOnClickListener(new View.OnClickListener() {
```

```
        @Override
```

```
        public void onClick(View view) {
```

```
            String msg = validation();
```

```
            if (TextUtils.isEmpty(msg)) {
```

```

        doCheckDb();

    } else {

        Helpers.showDialogWithMessage(LoginActivity.this,
getString(R.string.validation),

        msg, getString(R.string.ok), null, true);

    }

}

});

new UserHelpers().setData();

}

private String validation() {

    String msg = "";

    String userId = etUsernameLogin.getText().toString().trim();

    String password = etPasswordLogin.getText().toString().trim();

    if (TextUtils.isEmpty(userId)) {

        msg += "UserName cannot be empty\n";

    }

    if (TextUtils.isEmpty(password)) {

        msg += "Password cannot be empty";

    }
}

```

```

        return msg;
    }

    private void doCheckDb() {

        String password = etPasswordLogin.getText().toString().trim();

        String id= etUsernameLogin.getText().toString().trim();

        UserProfile profile = new
RealmBaseHelper().find(realm,UserProfile.class,"userID",id);

        if (profile!=null){

            if (!profile.getPassword().equalsIgnoreCase(password))

                {

                    Helpers.showDialogWithMessage(LoginActivity.this,getString(R.string.validation
n),

                        getString(R.string.profile_invalid), getString(R.string.ok), null,
true);

                }else{

                    goNext(id);

                }

            }else {

```

```

Helpers.showDialogWithMessage(LoginActivity.this,getString(R.string.validation),
    getString(R.string.profile_notfound), getString(R.string.ok), null,
    true);
    }
}

private void goNext(String id){

    Intent intent =new Intent(this,MainMenuActivity.class);

    intent.putExtra("userID",id);

    startActivityForResult(intent,101);

}

@Override

protected void onDestroy() {

    super.onDestroy();

    if (realm != null) {

        realm.close();

    }

}

}

```

B. UI Layout

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:background="@drawable/side_nav_bar"

    android:orientation="vertical"

    android:padding="@dimen/activity_vertical_margin">


    <android.support.v7.widget.AppCompatImageView

        android:id="@+id/imageViewLogo"

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:padding="@dimen/margin_5dp"

        android:scaleType="fitCenter"

        android:src="@drawable/ic_logo" />


    <android.support.v7.widget.AppCompatTextView

        android:layout_width="match_parent"

        android:layout_height="wrap_content"
```

```

    android:textStyle="bold"

    android:padding="@dimen/margin_5dp"

    android:textAllCaps="true"

    android:text="@string/app_name"

    android:gravity="center"

    android:textColor="@color/window_background"

    android:textSize="@dimen/text_24sp"/>

```

```
<LinearLayout
```

```

    android:id="@+id/linLoginForm"

    android:layout_width="match_parent"

    android:layout_height="wrap_content"

    android:layout_marginTop="@dimen/activity_horizontal_margin"

    android:background="@color/blackTransparent"

    android:orientation="vertical"

    android:paddingBottom="@dimen/margin_5dp"

    android:paddingLeft="@dimen/margin_2dp"

    android:paddingRight="@dimen/margin_2dp"

    android:paddingTop="@dimen/margin_5dp">

```

```
<android.support.design.widget.TextInputLayout
```

```

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:layout_marginTop="@dimen/margin_between_label"

        android:textColorHint="@color/window_background">

```

```

<EditText

```

```

    android:id="@+id/etUsernameLogin"

    style="@style/InputTextMaterialLogin"

    android:layout_width="match_parent"

    android:layout_height="wrap_content"

    android:hint="@string/username"

    android:inputType="text"

    android:maxLength="50"

    android:maxLines="1"

    android:textColor="@color/window_background" />

```

```

</android.support.design.widget.TextInputLayout>

```

```

<android.support.design.widget.TextInputLayout

```

```

    android:layout_width="match_parent"

    android:layout_height="wrap_content"

    android:layout_marginTop="@dimen/margin_between_label"

```

```

        android:textColorHint="@color/window_background"

        app:passwordToggleEnabled="true"

        app:passwordToggleTint="@color/window_background">

```

```

<EditText

```

```

    android:id="@+id/etPasswordLogin"

    style="@style/InputTextMaterialLogin"

    android:layout_width="match_parent"

    android:layout_height="wrap_content"

    android:hint="@string/password"

    android:inputType="textPassword"

    android:maxLength="50"

    android:maxLines="1"

    android:textColor="@color/window_background" />

```

```

</android.support.design.widget.TextInputLayout>

```

```

<RelativeLayout

```

```

    android:layout_width="match_parent"

    android:layout_height="wrap_content"

    android:layout_marginTop="@dimen/margin_10dp"

    android:orientation="horizontal">

```

```

<Button
    android:id="@+id/btnLogin"
    style="@style/MaterialButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:gravity="center"
    android:text="@string/login"
    android:textColor="@color/window_background"
    android:theme="@style/MaterialBtBlack" />

</RelativeLayout>

</LinearLayout>

</LinearLayout>

```

C. Main Menu

```

package com.arimurti.sto.ui.activity;

import android.content.Intent;

import android.graphics.Color;

import android.graphics.Typeface;

import android.os.Bundle;

```

```
import android.support.annotation.NonNull;

import android.support.v7.app.AppCompatActivity;

import android.support.v7.widget.AppCompatButton;

import android.support.v7.widget.AppCompatImageView;

import android.support.v7.widget.AppCompatTextView;

import android.support.v7.widget.GridLayoutManager;

import android.support.v7.widget.RecyclerView;

import android.text.SpannableString;

import android.text.style.ForegroundColorSpan;

import android.text.style.RelativeSizeSpan;

import android.text.style.StyleSpan;

import android.view.View;

import android.widget.RelativeLayout;


import com.afollestad.materialdialogs.DialogAction;

import com.afollestad.materialdialogs.MaterialDialog;

import com.arimurti.sto.R;

import com.arimurti.sto.models.Menus;

import com.arimurti.sto.models.UserProfile;

import com.arimurti.sto.ui.adapter.MainMenuAdapter;

import com.arimurti.sto.utils.Helpers;

import com.github.mikephil.charting.animation.Easing;
```

```

import com.github.mikephil.charting.charts.PieChart;

import com.github.mikephil.charting.components.Legend;

import com.github.mikephil.charting.data.PieData;

import com.github.mikephil.charting.data.PieDataSet;

import com.github.mikephil.charting.data.PieEntry;

import com.github.mikephil.charting.formatter.PercentFormatter;

import com.github.mikephil.charting.utils.ColorTemplate;

import com.github.mikephil.charting.utils.MPPointF;

import
com.mikepenz.community_material_typeface_library.CommunityMaterial;


import java.util.ArrayList;

import java.util.List;


public class MainMenuActivity extends AppCompatActivity {

    private UserProfile user;

    private List<Menus> menusList;

    private RecyclerView rvMenu;

    private final int MAIN_MENU = 101;

    private RelativeLayout reportSTO;

    private PieChart mChart;

```

```
private AppCompatButton btnReportStock;

private AppCompatTextView tvStockIn,tvStockout;

private int mTotalStock;
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_main_menu);

    dummyUser();

    setMenus();

    setHeader();

}
```

```
private void setHeader(){

    AppCompatTextView mName =
    findViewById(R.id.textUsernameMainMenu);

    AppCompatTextView mUserId =
    findViewById(R.id.userIDViewMainMenu);

    AppCompatImageView mPic = findViewById(R.id.imageViewMainMenu);

    AppCompatButton btnLogOut = findViewById(R.id.btn_log_out);
```

```

btnLogOut.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View view) {

        Helpers.showDialogTwoButton(MainMenuActivity.this,
getString(R.string.log_out), getString(R.string.message_logout),

        "Yes", new MaterialDialog.SingleButtonCallback() {

            @Override

            public void onClick(@NonNull MaterialDialog dialog,
@NonNull DialogAction which) {

                finish();

            }

            }, "No", null, true);

        }

    });

    if (user!=null){

        mUserId.setText(user.getUserID());

        mName.setText(user.getName());

        mPic.setImageDrawable(Helpers.getIconName(user.getName()));

    }

}

private void setMenus() {

```

```

boolean isVisible = user.getRoles() == 0;

reportSTO = findViewById(R.id.reportSTO);

mChart = findViewById(R.id.chart1Pie);

tvStockIn = findViewById(R.id.tvStockIn);

tvStockout = findViewById(R.id.tvStockOut);


rvMenu = findViewById(R.id.rvMenu);

rvMenu.setLayoutManager(new
GridLayoutManager(MainMenuActivity.this, 1));

rvMenu.setHasFixedSize(true);

rvMenu.setFocusable(false);


if (isVisible) {

    rvMenu.setVisibility(View.VISIBLE);

    reportSTO.setVisibility(View.GONE);

    menusList = new ArrayList<>();

    menusList.add(new Menu(getString(R.string.stocks),
Helpers.getLightIconActionBar(MainMenuActivity.this,
CommunityMaterial.Icon.cmd_clipboard_check),
getResources().getColor(R.color.colorStock), 0));

    menusList.add(new Menu(getString(R.string.stock_inout),
Helpers.getLightIconActionBar(MainMenuActivity.this,

```

```

        CommunityMaterial.Icon.cmd_elevator),
getResources().getColor(R.color.colorInOut), 1));

        menusList.add(new Menu(getString(R.string.app_name),
Helpers.getLightIconActionBar(MainMenuActivity.this,

        CommunityMaterial.Icon.cmd_barcode_scan),
getResources().getColor(R.color.colorAssignment), 2));

        rvMenu.setAdapter(new MainMenuAdapter(menusList,
MainMenuActivity.this, adapterCallback));

    }

    else {

        rvMenu.setVisibility(View.GONE);

        reportSTO.setVisibility(View.VISIBLE);

        setChart();

    }

}

private void setChart(){

    mChart.setUsePercentValues(true);

    mChart.getDescription().setEnabled(false);

    mChart.setExtraOffsets(5, 10, 5, 5);

```

```
mChart.setDragDecelerationFrictionCoef(0.95f);

mChart.setCenterText(generateCenterSpannableText());

mChart.setDrawHoleEnabled(true);

mChart.setHoleColor(Color.WHITE);

mChart.setTransparentCircleColor(Color.WHITE);

mChart.setTransparentCircleAlpha(110);

mChart.setHoleRadius(58f);

mChart.setTransparentCircleRadius(61f);

mChart.setDrawCenterText(true);

mChart.setRotationAngle(0);

// enable rotation of the chart by touch

mChart.setRotationEnabled(true);

mChart.setHighlightPerTapEnabled(true);

// mChart.setUnit(" €");

// mChart.setDrawUnitsInChart(true);
```

```

setData(200, 100,300);

mChart.animateY(1400, Easing.EasingOption.EaseInOutQuad);

// mChart.spin(2000, 0, 360);

Legend l = mChart.getLegend();

l.setVerticalAlignment(Legend.LegendVerticalAlignment.TOP);

l.setHorizontalAlignment(Legend.LegendHorizontalAlignment.RIGHT);

l.setOrientation(Legend.LegendOrientation.VERTICAL);

l.setDrawInside(false);

l.setXEntrySpace(7f);

l.setYEntrySpace(0f);

l.setYOffset(0f);

// entry label styling

mChart.setEntryLabelColor(Color.WHITE);

mChart.setEntryLabelTextSize(12f);
}

private float calculatedata(int stock, int total){

```

```
        return (float)(stock*100)/total;
    }

    private void setData(int stockIN,int stockOUT, int total) {

        ArrayList<PieEntry> entries = new ArrayList<PieEntry>();

        entries.add(new PieEntry(calculatedata(stockIN,total),"Stock In"));
        entries.add(new PieEntry(calculatedata(stockOUT,total),"Stock Out"));

        PieDataSet dataSet = new PieDataSet(entries, "Election Results");

        dataSet.setDrawIcons(false);

        dataSet.setSliceSpace(3f);
        dataSet.setIconsOffset(new MPPointF(0, 40));
        dataSet.setSelectionShift(5f);

        // add a lot of colors

        ArrayList<Integer> colors = new ArrayList<Integer>();
        colors.add(getResources().getColor(R.color.colorStock));
```

```
colors.add(getResources().getColor(R.color.yellow_color));
```

```
dataSet.setColors(colors);
```

```
//dataSet.setSelectionShift(0f);
```

```
PieData data = new PieData(dataSet);
```

```
data.setValueFormatter(new PercentFormatter());
```

```
data.setValueTextSize(11f);
```

```
data.setValueTextColor(Color.WHITE);
```

```
mChart.setData(data);
```

```
// undo all highlights
```

```
mChart.highlightValues(null);
```

```
mChart.invalidate();
```

```
}
```

```
private SpannableString generateCenterSpannableText() {
```

```
    SpannableString s = new SpannableString("Stock In/Out\n"+mTotalStock+"  
items");
```

```
    s.setSpan(new RelativeSizeSpan(1.7f), 0, 12, 0);
```

```

        s.setSpan(new RelativeSizeSpan(1f), s.length() - 12, s.length() , 0);

        s.setSpan(new StyleSpan(Typeface.BOLD_ITALIC), s.length() - 12,
s.length(), 0);

        s.setSpan(new ForegroundColorSpan(ColorTemplate.getHoloBlue()),
s.length() - 12, s.length(), 0);

        return s;
    }

```

```

private void dummyUser() {

    user = new UserProfile();

    user.setName("Winarso wijoyo");

    user.setUserID("Win");

    user.setRoles(0);

}

```

```

private MainMenuAdapter.Callback adapterCallback = new
MainMenuAdapter.Callback() {

    @Override

    public void onItemClick(Menus homeMenu) {

        if (menusList == null) return;

        if (menusList.isEmpty()) return;

```

```
Intent intent = null;

// using head menu name

switch (homeMenu.getSequence()) {

    case 0:

        intent = new
Intent(MainMenuActivity.this,AssignmentStockActivity.class);

        break;

    case 1:

        intent = new
Intent(MainMenuActivity.this,StockOpnamesActivity.class);

intent.putExtra(getString(R.string.title_reference),getString(R.string.stock_inout))
;

        intent.putExtra(getString(R.string.is_inout),true);

        break;

    case 2:

        intent = new
Intent(MainMenuActivity.this,StockOpnamesActivity.class);

intent.putExtra(getString(R.string.title_reference),getString(R.string.stock_inout))
;

        intent.putExtra(getString(R.string.is_inout),false);

        break;

}
```

```
        if (intent != null) {  
            startActivityForResult(intent, MAIN_MENU);  
        }  
    }  
};  
  
}
```

D. Stock Opname

```
package com.arimurti.sto.ui.activity;  
  
import android.Manifest;  
  
import android.content.Intent;  
  
import android.content.pm.PackageManager;  
  
import android.os.Bundle;  
  
import android.support.annotation.NonNull;  
  
import android.support.annotation.Nullable;  
  
import android.support.v4.app.ActivityCompat;  
  
import android.support.v7.app.ActionBar;  
  
import android.support.v7.app.AppCompatActivity;
```

```
import android.support.v7.widget.AppCompatButton;

import android.support.v7.widget.AppCompatEditText;

import android.text.TextUtils;

import android.util.Log;

import android.util.SparseArray;

import android.view.MenuItem;

import android.view.SurfaceHolder;

import android.view.SurfaceView;

import android.view.View;

import android.widget.LinearLayout;


import com.afollestad.materialdialogs.MaterialDialog;

import com.arimurti.sto.R;

import com.arimurti.sto.utils.Helpers;

import com.fastaccess.permission.base.PermissionHelper;

import com.fastaccess.permission.base.callback.OnPermissionCallback;

import com.google.android.gms.vision.CameraSource;

import com.google.android.gms.vision.Detector;

import com.google.android.gms.vision.barcode.Barcode;

import com.google.android.gms.vision.barcode.BarcodeDetector;


import java.io.IOException;
```

```

public class StockOpnamesActivity extends AppCompatActivity implements
OnPermissionCallback {

    private String mTitle;

    private boolean isInOut;

    private SurfaceView cameraView;

    private CameraSource cameraSource;

    private AppCompatActivity etBarcode;

    private AppCompatActivity btnSubmitBarcode;

    private AppCompatActivity btnStockIn;

    private AppCompatActivity btnStockOut;

    private LinearLayout linBtnStock;

    private BarcodeDetector detector;

    private PermissionHelper permissionHelper;

    private MaterialDialog mProgressDialog;

    private final int REQUEST_STO = 102;

    private int statusStock = 0; // 0 = notINOUT, 1 = IN , 2 = OUT;

    @Override

    protected void onCreate(@Nullable Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

```

```

setContentView(R.layout.layout_scanner);

Bundle bundle = getIntent().getExtras();

if (bundle!=null){

    if (bundle.containsKey(getString(R.string.title_reference))){

        mTitle=bundle.getString(getString(R.string.title_reference));

    }

    if (bundle.containsKey(getString(R.string.is_inout))){

        isInOut = bundle.getBoolean(getString(R.string.is_inout));

    }

}

ActionBar actionBar = getSupportActionBar();

if (actionBar!=null){

Helpers.setMainActionBar(StockOpnamesActivity.this,actionBar,mTitle,true);

}

cameraView = findViewById(R.id.camera_view);

etBarcode = findViewById(R.id.et_barcode);

linBtnStock = findViewById(R.id.lay_in_out);

btnSubmitBarcode = findViewById(R.id.button_opname);

btnStockIn = findViewById(R.id.button_in);

btnStockOut = findViewById(R.id.button_out);

```

```

        btnSubmitBarcode.setVisibility(!isInOut ? View.VISIBLE : View.GONE);

        linBtnStock.setVisibility(isInOut ? View.VISIBLE : View.GONE);

        showCamera();

        permissionHelper
        PermissionHelper.getInstance(StockOpnamesActivity.this);

        setListener();
    }

    private void setListener(){

        btnSubmitBarcode.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View view) {

                loadData();

            }

        });

        btnStockIn.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View view) {

                statusStock = 1;

```

```

        loadData();

    }

});

```

```

btnStockOut.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View view) {

        statusStock = 2;

        loadData();

    }

});

```

```

}

```

```

private void loadData(){

    String values = etBarcode.getText().toString().trim();

    if (!TextUtils.isEmpty(values)) {

        getValues(values);

    }else {

```

```

Helpers.showDialogWithMessage(StockOpnamesActivity.this,getString(R.string.
validation),

```

```

        getString(R.string.stock_notfound),getString(R.string.yes),null,true);

    }

}

public void showCamera() {

    mProgressDialog =
    Helpers.showProgressDialog(StockOpnamesActivity.this, "", true);

    mProgressDialog.show();

    detector = new
    BarcodeDetector.Builder(StockOpnamesActivity.this).setBarcodeFormats(Barcode.QR_CODE)

    .build();

    cameraSource = new CameraSource.Builder(StockOpnamesActivity.this,
    detector)

    .setFacing(CameraSource.CAMERA_FACING_BACK)

    .setAutoFocusEnabled(true)

    .setRequestedPreviewSize(640, 480)

    .build();

    cameraView.getHolder().addCallback(new SurfaceHolder.Callback() {

        @Override

        public void surfaceCreated(SurfaceHolder holder) {

```

```

try {

    setCamera(holder);

} catch (IOException e) {

    Log.e("CAMERA SOURCE", e.getMessage());

}

}

```

@Override

```

public void surfaceChanged(SurfaceHolder holder, int format, int width, int
height) {

```

```

    try {

        setCamera(holder);

    } catch (IOException e) {

        e.printStackTrace();

        Log.e("CAMERA SOURCE", e.getMessage());

    }

}

```

@Override

```

public void surfaceDestroyed(SurfaceHolder holder) {

    cameraSource.stop();

}

```



```

        }

    }

    });

}

}

}

});

}

```

```

private void getValues(String valueBarcode){

    if (doCheckDB(valueBarcode)){

        goesToDetail(valueBarcode);

    }else {

```

```

Helpers.showDialogWithMessage(StockOpnamesActivity.this,getString(R.string.
validation),

```

```

        getString(R.string.stock_notfound),getString(R.string.yes),null,true);

    }

}

```

```

private void goesToDetail(String values){

    boolean wrapInScrollView = true;

    MaterialDialog dialog = new MaterialDialog.Builder(this)

```

```
.title(R.string.app_name)

.customView(R.layout.dialog_detail, wrapInScrollView)

.show();
```

```
View view = dialog.getCustomView();
```

```
switch (statusStock){
```

```
    case 0 :
```

```
        break;
```

```
    case 1:
```

```
        break;
```

```
    case 2:
```

```
        break;
```

```
}
```

```
}
```

```
private boolean doCheckDB(String values){
```

```
    return true;
```

```
}
```

```

private void setCamera(SurfaceHolder holder) throws IOException {

    if (ActivityCompat.checkSelfPermission(StockOpnamesActivity.this,

        Manifest.permission.CAMERA) !=
PackageManager.PERMISSION_GRANTED) {

        // TODO: Consider calling

        //   ActivityCompat#requestPermissions

        // here to request the missing permissions, and then overriding

        //   public void onRequestPermissionsResult(int requestCode, String[]
permissions,

        //                                       int[] grantResults)

        // to handle the case where the user grants the permission. See the
documentation

        // for ActivityCompat#requestPermissions for more details.

        permissionHelper.setForceAccepting(false) // default is false. its here so
you know that it exists.

        .request(new String[]{Manifest.permission.CAMERA,
Manifest.permission.READ_EXTERNAL_STORAGE});

    } else {

        if (mProgressDialog != null) {

            mProgressDialog.dismiss();

        }

        cameraSource.start(holder);

```

```
}
```

```
}
```

```
@Override
```

```
public void onPermissionGranted(@NonNull String[] permissionName) {
```

```
    try {
```

```
        setCamera(cameraView.getHolder());
```

```
    } catch (IOException e) {
```

```
        e.printStackTrace();
```

```
    }
```

```
}
```

```
@Override
```

```
public void onPermissionDeclined(@NonNull String[] permissionName) {
```

```
    if (mProgressDialog != null) {
```

```
        mProgressDialog.dismiss();
```

```
    }
```

```
}
```

```
@Override
```

```
public void onPermissionPreGranted(@NonNull String permissionsName) {
```

```
}
```

```
@Override
```

```
public void onPermissionNeedExplanation(@NonNull String permissionName)
{

}

}
```

```
@Override
```

```
public void onPermissionReallyDeclined(@NonNull String permissionName) {

    if (mProgressDialog != null) {

        mProgressDialog.dismiss();

    }

}
```

```
@Override
```

```
public void onNoPermissionNeeded() {

    try {

        setCamera(cameraView.getHolder());

    } catch (IOException e) {

        e.printStackTrace();

    }

}
```

```

    }
}

```

@Override

```

public void onActivityResult(int requestCode, int resultCode, Intent data) {

    permissionHelper.onActivityResult(requestCode);

    super.onActivityResult(requestCode, resultCode, data);

}

```

@Override

```

public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {

    permissionHelper.onRequestPermissionsResult(requestCode, permissions,
grantResults);

    super.onRequestPermissionsResult(requestCode, permissions, grantResults);

}

```

@Override

```

public boolean onOptionsItemSelected(MenuItem item) {

    switch (item.getItemId()) {

        case android.R.id.home:

            setResult(RESULT_CANCELED);

            finish();


```

```
        return true;

    default:

        return super.onOptionsItemSelected(item);

    }

}

}
```

4.4 Testing

A. Form login pegawai/user

No.	Skenario pengujian	Test case	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1.	Mengosongkan semua field form login	Username : (kosong) Password : (kosong)	Sistem akan menolak dan menampilkan pesan Notif : Username cannot be empty password cannot be empty	Sesuai harapan	<i>valid</i>
2.	Mengosongkan field username form login	Username : (kosong) Password : test@122	Sistem akan menolak dan Menampilkan pesan Notif : Username cannot be empty	Sesuai harapan	<i>valid</i>
3.	Mengosongkan field Password form login	Username : Win Password : (kosong)	Sistem akan menolak dan Menampilkan pesan Notif : password cannot be empty	Sesuai harapan	<i>valid</i>
4.	Mengisi semua field form login dengan memasukan username & password yang salah	Username : win005810 (salah) Password : test@122 (salah)	Sistem akan menolak dan menampilkan pesan Notif : profil not found	Sesuai harapan	<i>valid</i>
5.	Mengisikan semua field form login dengan memasukan password yang salah	Username : win (benar) Password : tes@122	Sistem akan menolak dan menampilkan pesan Notif :password invalid, please check your password	Sesuai harapan	<i>valid</i>
6.	Mengisi semua field form login Dengan memasukan username & password yang benar	Username : win (benar) Password : a	Sistem akan menerima akses login dan masuk ke menu user	Sesuai harapan	<i>valid</i>

Tabel IV. 4

Hasil Pengujian *Black Box Testing Form Login Pegawai/user*

B. Form input/output stock opname

No.	Skenario pengujian	Test case	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1.	Sudah login sebagai user, lalu mengosongkan semua isian data, lalu klik tombol submit	Barcode : (kosong)	Sistem akan menolak menyimpan data dan menampilkan pesan “Please scan/input barcode first”	Sesuai harapan	<i>valid</i>
2.	Sudah login sebagai user, dan memasukan semua isian yang sudah diinput sebelumnya, lalu mengklik tombol submit	Barcode : R0031212 (benar) Item description : Amplop AirMap Coklat (benar) Bin area : A0021 (benar) Quantity : 40 (benar)	Sistem akan menolak menyimpan data dengan Qty yang berbeda di assignment stock dan menampilkan pesan “Quantity is not same with assignment”	Sesuai harapan	<i>valid</i>
3.	Sudah login sebagai user, dan memasukan semua isian dengan barcode yang salah, lalu mengklik tombol submit	Barcode : R00312331 (salah)	Sistem akan menolak menyimpan data dan menampilkan pesan “ the stock data has not been registered in the database”	Sesuai harapan	<i>valid</i>
4.	Sudah login sebagai user, dan memasukan semua isian dengan benar, lalu mengklik tombol submit	Barcode : R0031212 (benar)	Sistem akan menyimpan data dengan Qty yang sama di assignment stock, dan menampilkan “Do you want to proccess this item ?”	Sesuai harapan	<i>valid</i>

Tabel IV. 5

Hasil Pengujian *Black Box Testing Form input/output opname*

C. Form koreksi

No.	Skenario pengujian	Test case	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1.	Sudah login sebagai korektor, lalu mengosongkan semua isian data, lalu klik tombol submit	Barcode : (kosong)	Sistem akan menolak menyimpan data dan menampilkan pesan “Please scan/input barcode first”	Sesuai harapan	<i>valid</i>
2.	Sudah login sebagai korektor, dan memasukan barcode, semua isian yang sudah diinput sebelumnya, dan mengosongkan good condition, defect, missing lalu mengklik tombol submit	Barcode : R0031212 (benar) Good condition : (kosong) Defect : (kosong) Missing : (kosong)	Sistem akan menampilkan data stock opname dan menampilkan pesan “Please input detail qty”	Sesuai harapan	<i>valid</i>
3.	Sudah login sebagai korektor, dan memasukan semua isian dengan benar sesuai quantiy yang sudah tersedia, lalu mengklik tombol submit	Barcode : R0031212 (benar) Good condition : 110 Defect : 10 Missing : 10	Sistem akan menyimpan data dan menampilkan pesan “do you want to proccess this item”	Sesuai harapan	<i>valid</i>

Tabel IV.6

Hasil Pengujian *Black Box Testing Form* koreksi

4.5 Support

4.5.1 Spesifikasi *Hardware* dan *software*

Pengembangan sistem informasi pengelolaan data *stock opname* toko ini menggunakan dukungan perangkat keras (*Hardware*) dan dukungan perangkat lunak (*software*).

Tabel IV.7

Spesifikasi *Hardware* dan *Software*

Ke butuhan	Keterangan
Sistem Operasi	Windows XP atau Android 4.1
Processor	Intel Core 2 Duo, 2,4GHz
RAM	1GB
Hard Disk	160GB
Monitor	LCD 14 “atau layar device 4”
Kamera	8MP
Keyboard	108 key atau qwerty atau touch screen
Printer	Laser jet
Mouse	Standard
Browser	Mozilla Firefox atau Google Chrome
Software	Android Studio, Enterprise Architect, Visio, Realm, google vision, draw io

4.6 Spesifikasi Dokumen Sistem Usulan

Spesifikasi sistem Usulan yang dimaksud adalah rancangan pada sistem informasi yang diusulkan mengenai masukan dan keluaran yang berupa tampilan-tampilan layar atau berupa laporan dalam bentuk kertas. Dokumen yang terdapat dalam sistem usulan PT. Arie Muti, yaitu:

1. Nama dokumen : *Form Login*

Fungsi : Sebagai akses masuk untuk melakukan aktivitas stock opname

Tujuan : User, korektor dan admin

Media : Aplikasi

Frekuensi : Setiap akan memulai kegiatan stock opname

Bentuk Dokumen : Lampiran B-1

2. Nama dokumen : *Form input opname*
Fungsi : Sebagai media penyimpanan data opname
Tujuan : User
Media : Aplikasi
Frekuensi : Setiap melakukan kegiatan stock opname
Bentuk dokumen : Lampiran B-2
3. Nama Dokumen : *Form input koreksi*
Fungsi : Sebagai Media penyimpanan data koreksi
Tujuan : Korektor
Media : Aplikasi
Frekuensi : Setiap melakukan kegiatan koreksi opname
Bentuk Dokumen : Lampiran B-3
4. Nama Dokumen : Laporan Final Opname
Fungsi : Sebagai laporan hasil akhir data opname
Tujuan : Admin
Media : Aplikasi dan Kertas
Frekuensi : Setiap setelah melakukan kegiatan stock opname
Bentuk Dokumen : Lampiran B-4