CHAPTER 5
IMPLEMENTATION AND TESTING

5.1 Implementation

Below in are some function from the source code that has been made:

```
1. public String connectToWebService(String urlString) {
2.     try {
3.         URL url = new URL(urlString);
4.         HttpURLConnection con = (HttpURLConnection)
   url.openConnection();

5.         con.setRequestProperty( "Content-Type", "application/x-
   www-form-urlencoded");
6.         con.setConnectTimeout(300000);
7.         con.setReadTimeout(300000);
8.         con.setRequestMethod("POST");

9.         BufferedReader in = new BufferedReader(new
   InputStreamReader(con.getInputStream()));
10.        res = in.readLine();

11.        in.close();
12.        con.disconnect();

13.        return res;
14.    } catch (Exception e) {
15.        return res = "" + e;
16.    }
17.    }
```

line 1 to 17 program code contains a command to create a function named connectToWebService which requires input a variable named urlString with a String type. Lines 2 to 14 try to make a request. Line 3 creates a url variable with the URL type from the urlString input. Line 4 makes a connection of type HttpURLConnection and opens the connection. Line 5 determines the Request Property with content in the urlencoded form. Lines 6 and 7 determine the connection and read timeout of 5 minutes. Line 8 determines the request post method. Lines 9 and 10 read the response and save it with the variable string type res. Lines 11 and 12 stop reading responses and disconnect. Line 13 returns the result of the res variable. Lines 14 to 16 Catch and return error if an error occurs.

```
18.   $dbhost = "localhost";
19.   $dbuser = "root";
20.   $dbpass = "";
21.   $db = "dbstockopname";

22.   $con = new mysqli($dbhost, $dbuser, $dbpass,$db) or
   die("Connect failed: %s\n". $con -> error);

23.   mysqli_autocommit($con, false);
```

Lines 18 through 21 enter credentials from the existing database. Line 22 makes a connection to the database or returns the message Connect failed: with an error if it fails. Line 23 sets auto commit to false.

```
24.   Try {
25.   $strSQL = "UPDATE TBLOKASI SET LKS_kodebarang = '$KodePLU',
  LKS_EXPDATE = '".DateTime::createFromFormat('d/m/Y', $exp)-
  >format('Y-m-d')." 00:00:00', LKS_QTY = '".(($CTN * $Fraction)
  + $Pcs)."', LKS_MODIFY_BY = '$user', LKS_MODIFY_DT = SYSDATE()
  WHERE LKS_KODERAK = '$KodeRak' AND LKS_KODESUBRAK =
  '$KodeSubRak' AND LKS_TIPERAK = '$TipeRak' AND LKS_SHELVINGRAK
  = '$Shelving' AND LKS_NOURUT = '$NoUrut'";

26.   mysqli_query($con, $strSQL);
27.   mysqli_commit($con);

28.   $con -> close();
29.   echo "Sukses";
30.   return "Sukses";
31.   } catch (Exception $e) {
32.   mysqli_rollback($con);
33.   $con -> close();
34.   echo "Failed";
35.   return "Failed";
36.   }
```

Line 25 creates a strSQL variable of type String that has the value of one of the sample queries to update the database. Line 26 Execute a query from the strSQL variable. Line 27 commits the query. Line 28 closes the connection to the database. Lines 29 and 30 display and return success if the process is successful. Line 31 catches an error if an error occurs. Line 32 rollback all queries that have not been committed. Row 33 closes the connection to the database. Lines 34 and 35 display and return Failed if it fails.

```
37.   try {
38.   $itemList[] = array('KODELOKASI' => $dt['LKS_KODERAK']."".
   $dt['LKS_KODESUBRAK']."".$dt['LKS_TIPERAK'."".
   dt['LKS_SHELVINGRAK']."".$dt['LKS_NOURUT'], 'KODEBARANG' =>
   $dt['LKS_kodebarang'], 'DESKRIPSI' => $dt['brg_DESKRIPSI'],
   'FRAC' => $dt['brg_FRAC'], 'QTY' => $dt['LKS_QTY'], 'EXP' =>
   $exp, 'MODIFY_BY' => $dt['LKS_MODIFY_BY'], 'MODIFY_DT' =>
   $mod);
39.       }
40.   $json = json_encode($itemList);
```

Lines 37 through 39 try to create a json object from an array's key value pair. Line 38 creates an array of key value pairs which will be encoded in line 40.

```
41.   <?php
42.   $awal = microtime(true);
43.   $output = `php SOStorage_GetList.php`;
44.   $akhir = microtime(true);
45.   $lama = $akhir - $awal;
46.   echo "Lama eksekusi script adalah: ".$lama." second";
47.   ?>
```

Line 41 starts the php programming language. Lines 42 and 44 call the microtime function with the parameter true so that the return value of time becomes float second and then stores it in the awal variable. Line 43 Executes the SOStorage_GetList.php file and stores its return value in the output variable. Line 45 to calculate the difference between the akhir variable minus the awal variable and save the difference in the lama variable. Line 46 displays the value of the lama variable with the sentence "The length of execution of the script is: {lama} second".

```
48.   ConnectivityManager connectivityManager =
  (ConnectivityManager) getSystemService(CONNECTIVITY_SERVICE);
49.   for (Network network : connectivityManager.getAllNetworks())
  {
50.   NetworkInfo networkInfo =
  connectivityManager.getNetworkInfo(network);
51.   if (networkInfo.isConnected()) {
52.   LinkProperties linkProperties =
  connectivityManager.getLinkProperties(network);
53.   addressList = linkProperties.getDnsServers();
54.   for (int i = 0; i < addressList.size(); i++) {
55.   strIPAddress = addressList.get(i).toString().replace("/",
  "");
56.   if (!strIPAddress.contains(":")) {
57.   break;
58.   }     }     }     }
```

Line 48 takes system information from the device itself. Line 49 Looping to retrieve all network information that is on the device itself. Line 50 takes information from one of the networks on the device itself. Line 51 checks if there is a network connected to the device itself. Line 52 to retrieve the properties of the connection. Line 53 to take the IP address of the connection and store it in the addressList variable. Line 54 loop as many network connections as possible. Line 55 removes the "/" symbol from addressList and stores it in the strIPAddress variable. Line 56 checks if ":" symbol is found in the strIPAddress variable, it will break looping on line 57.

5.2 Testing

5.2.1 Main Menu Page

The Main Menu Page is the first page that will be displayed when an admin, employee, or anyone in Company X opens the Stock Opname Application. Before this page is ready to be displayed the system will check if there is an account that is still logged in this application or not. If not, the page will be forwarded to the Login page first. But if yes then the page will display as shown in Figure 5.

To display the menu options that can be done in two ways, namely: 1. By pressing the three lines on the top left side of the page title. 2. By swiping or sliding from the left side to the right side of the screen. By doing these two ways, the menu choices will be displayed. Finally, to display sub menus, what needs to be done is to press the menu, then the sub menu will be displayed as shown in Figure 6.

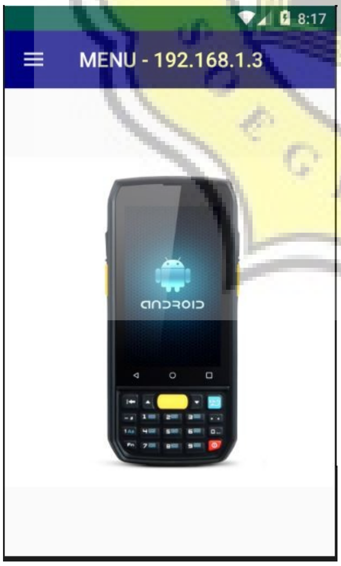


Figure 5: Main Menu (Initial)

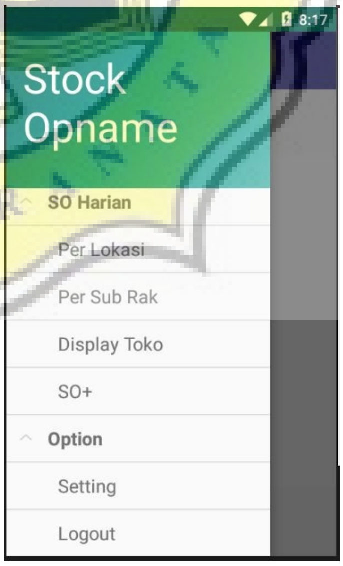


Figure 6: Main Menu (With menu)

5.2.2 Login Page

On the login page the user is asked to fill in the username and password that have been registered previously. When the LOGIN button is pressed, the application checks if the username and password fields have been filled. If not, the application will display a notification dialog to fill in the fields. If the column is filled then the system will make the request and sends the username and password to the Web service to be examined compatibility in Database.

If we want to exit the application then the thing to do is to press the cancel button then the application will close all activities and pages that are still running.

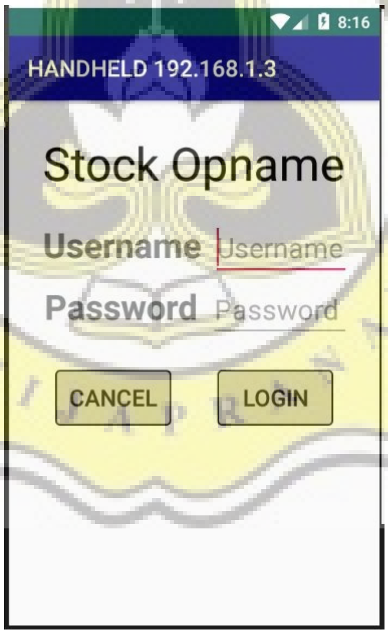
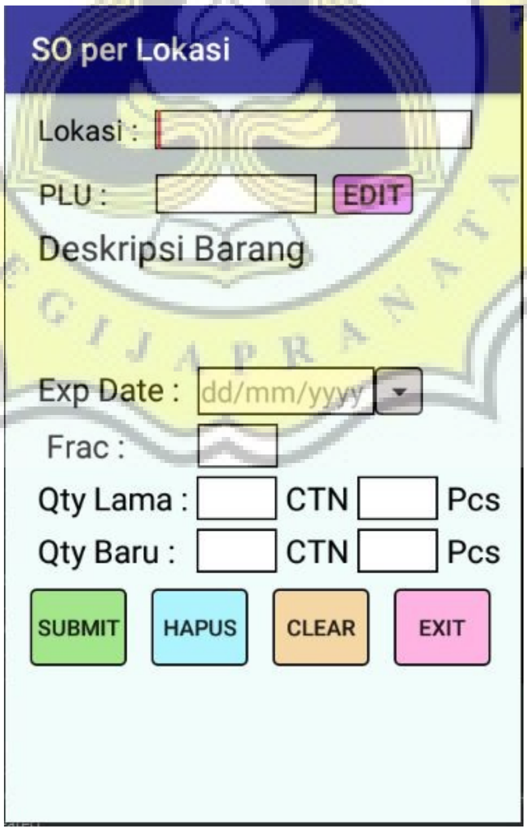


Figure 7: Login Page

5.2.3 SO Per Location Page

On the SO page per Location we can do Stock Opname activities according to the location of the goods in the warehouse. Things we need to know is the complete location of objects that the data need to be update. If we already know the location of the goods then there is some data that can be change, among others: PLU codes, Expired date, and the new quantity by pressing the SUBMIT button. We can also delete all data from a location by pressing the DELETE button. If we have entered the item data and we want to find other items, we can press the CLEAR button to clear all the columns that are there. Finally, when we have finished doing the Stock Menu, we can return to the main menu by pressing the EXIT button.



SO per Lokasi

Lokasi :

PLU :

Deskripsi Barang

Exp Date : dd/mm/yyyy

Frac :

Qty Lama : CTN Pcs

Qty Baru : CTN Pcs

Figure 8: SO Pages per Location

5.2.4 SO Per Sub Shelf Page

The SO page per Sub Shelves has functions that are more or less the same as SO per Location. But the thing that is different from SO Pages per Location is that we can search for item data just by knowing the location of the shelves. After that all data items on the shelf will be displayed on the application screen then we can do Stock Opname according to our needs.

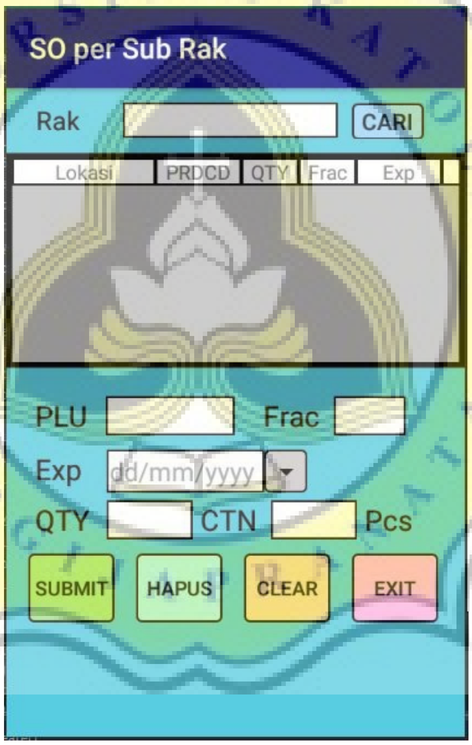


Figure 9: SO Pages per Sub Shelf

5.2.5 SO Display Shop Page

On the SO Display Shop Page we can do Stock Opname activities, what we need to know is the item code that needs to be changed. If we already know the item code we can press the CARI button then the application will make a request to the Web service that contains the item code that we are looking for. Then according to the system flow that has been made the screen will display data items that we will change as needed.

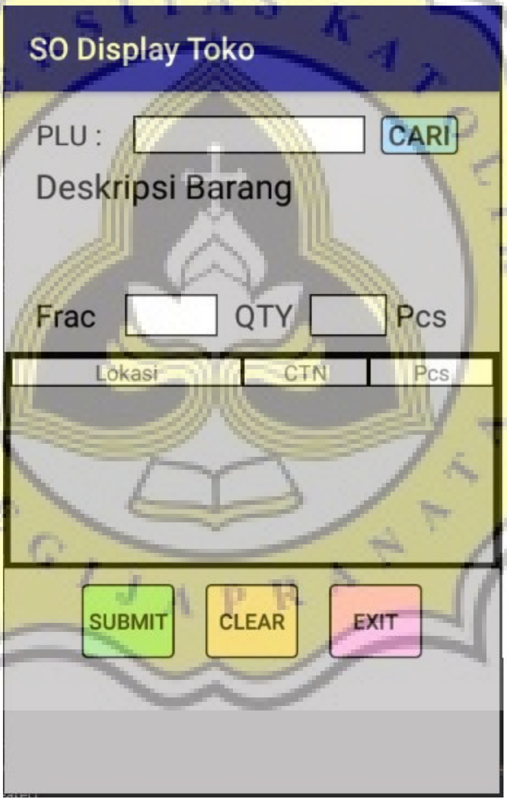


Figure 10: SO Display Shop Page

5.2.6 SO Plus Page

The SO Plus page is almost the same as SO per Location. But what is different from SO Plus is in the programming section when doing SUBMIT SO per Location can do the addition and reduction of the number of items so that it has a more complicated program than SO Plus which is deliberately made to make it easier if you only want to add a certain number of items.

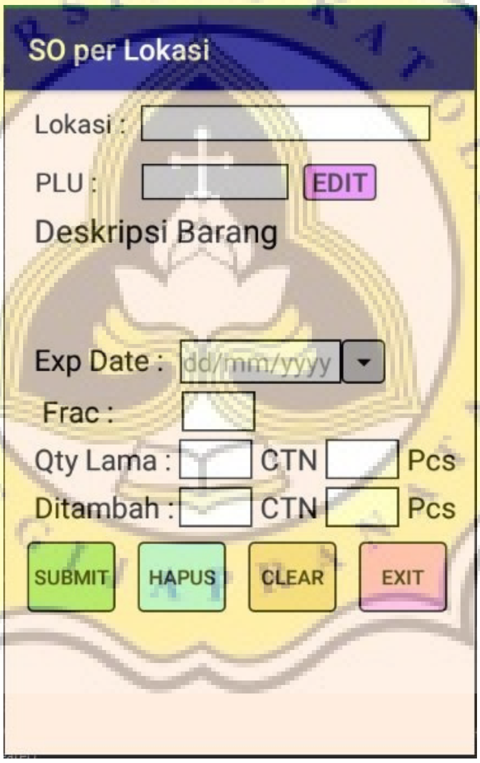
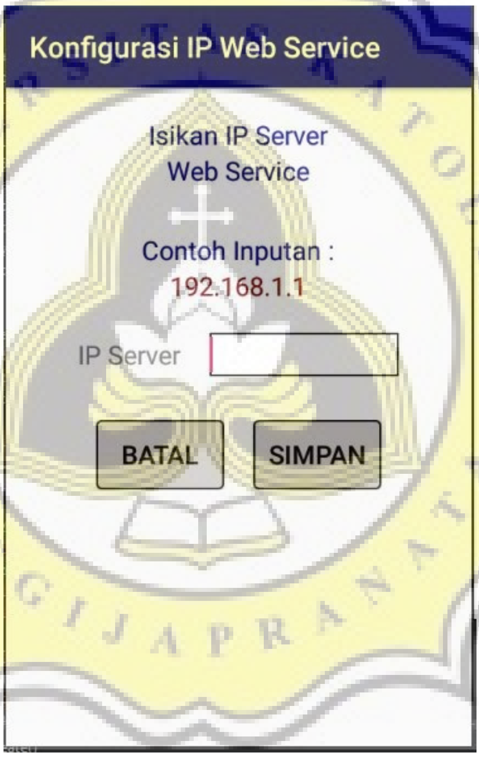


Figure 11: SO Plus Page

5.2.7 IP Configuration Page

IP Address Configuration page is used to change the address IP Web service because in Company X there are many servers having an IP address that is different. Therefore, so that this application can run on any Web service, we only need to change the IP Address without having to change an existing program.


Konfigurasi IP Web Service

Isikan IP Server
Web Service

Contoh Inputan :
192.168.1.1

IP Server

Figure 12: IP Configuration Page

Testing is an important part of the application development cycle. Testing is done to ensure quality and also find out the weaknesses of the application. The purpose of this test is to ensure that the application that is built has good quality. The author tests the application using white box method performed by the author and Support Team in Company X.