

APPENDIX

Source Code

Menu.java

```
package com.example.stockopname;
import ...

public class Menu extends AppCompatActivity {
    ...

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.form_menu_draw);

        session = new SessionManager...
        HashMap...

        if (userid == null || !userid.equals("ADM")) {
            try {
                jobj = new JSONArray(userAccess);
            } catch (Exception e) {
                AlertDialog.Builder alertBuilder = new
AlertDialog.Builder(Menu.this);
                alertBuilder.setMessage("Error " + e)
                    .setCancelable(false)
                    .setPositiveButton("OK", new
DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface
dialog, int id) {
                            }
                        });
                AlertDialog dialog = alertBuilder.create();
                dialog.show();
            }
        }

        final DrawerLayout drawer =
findViewById(R.id.drawer_layout);
        ActionBarDrawerToggle toggle = new
ActionBarDrawerToggle(this, drawer, toolbar,
            R.string.navigation_drawer_open,
            R.string.navigation_drawer_close);
        drawer.addDrawerListener(toggle);
        toggle.syncState();

        // get the listview
        expListView = findViewById(R.id.lvExp);
        // preparing list data
        prepareListData();
    }
}
```

```

        listAdapter = new Expandable_List_Adapter(this,
listDataHeader, listDataChild);
        // setting list adapter
        expListView.setAdapter(listAdapter);

        // Listview Group click listener
        expListView.setOnGroupClickListener(new
OnGroupClickListener() {
            @Override
            public boolean onGroupClick(ExpandableListView parent,
View v,
int groupPosition, long
id) {
                return false;
            }
        });

        // Listview on child click listener
        expListView.setOnChildClickListener(new
OnChildClickListener() {
            @Override
            public boolean onChildClick(ExpandableListView parent,
View v,
int groupPosition, int
childPosition, long id) {
                String str_parent =
listDataHeader.get(groupPosition);
                String str_child =
Objects.requireNonNull(listDataChild.get(listDataHeader.get(groupP
osition)))
                    .get(childPosition);

                switch (str_parent) {
                    case "SO Harian":
                        if (userid.equals("ADM")) {
                            AlertDialog.Builder aBu = new
AlertDialog.Builder(Menu.this);
                            aBu.setMessage("ADMIN hanya bisa
setting webservice")
                                .setCancelable(false)
                                .setPositiveButton("OK", new
DialogInterface.OnClickListener() {
                                    public void
onClick(DialogInterface dialog, int id) {
                                        }
                                    });
                            AlertDialog dia = aBu.create();
                            dia.show();
                            return true;
                        }

                        for (i = 0; i < jobj.length(); i++) {
                            try {

```

```

                                jsonObject =
jobj.getJSONObject(i);

                                if
(jsonObject.getString("ACCESSNAME").equals("SO") &&
jsonObject.getInt("USERLEVEL") <= 2) {
                                switch (str_child) {
                                case "Per
Lokasi" :
                                Intent intent1
= new Intent(getApplicationContext(), SOPerlokasi.class);
                                intent1.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                                intent1.putExtra("EXIT", true);
                                startActivity(intent1);
                                drawer.closeDrawer(GravityCompat.START);
                                return true;
                                case "Per Sub Rak"
:
                                return true;
                                case "Display
Toko" :
                                return true;
                                case "SO+":
                                drawer.closeDrawer(GravityCompat.START);
                                return true;
                                }
                                } catch (Exception e) {
                                ...
                                }
                                ...

                                drawer.closeDrawer(GravityCompat.START);
                                return true;
                                case "Option":
                                switch (str_child) {
                                case "Setting" :
                                session.addFlagSetting("1");

                                Intent intent = new
Intent(getApplicationContext(), Konfigurasi_IP_Server.class);

```

```

intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                    intent.putExtra("EXIT", true);
                    startActivity(intent);

drawer.closeDrawer(GravityCompat.START);
                    return true;
                    case "Logout" :
                        alertBuilder = new
AlertDialog.Builder(Menu.this);
                        alertBuilder.setMessage("Apakah
anda yakin ingin logout?")
                                .setCancelable(false)
                                .setPositiveButton("YES",
new DialogInterface.OnClickListener() {
                                    public void
onClick(DialogInterface dialog, int id) {
                                        if
(userid.equals("ADM")) {
                                            session.logoutUser();
                                                finish();
                                        } else {
                                            ConnectivityManager connectivityManager = (ConnectivityManager)
getSystemService(CONNECTIVITY_SERVICE);
                                                for (Network
network : connectivityManager.getAllNetworks()) {
                                                    NetworkInfo networkInfo =
connectivityManager.getNetworkInfo(network);
                                                        if
(networkInfo.isConnected()) {
                                                            LinkProperties linkProperties =
connectivityManager.getLinkProperties(network);
                                                                addressList = linkProperties.getDnsServers();
                                                                    for
(int i = 0; i < addressList.size(); i++) {
                                                                        strIPAddress = addressList.get(i).toString().replace("/", "");
                                                                            if
(!strIPAddress.contains(":")) {
                                                                                break;
                                                                                    }
                                                                    }
                                                                }
                                                            }
                                                        }
                                                    }
                                                }
                                            }
                                        }
                                    }
                                }
                            }

```

```

= url + "Logout.php?IP=" + strIPAddress;
BackPro().execute(str).get();
(Exception e) {

ret.equals("Connection Failed")) {
session.logoutUser();

try {
String str
ret = new
} catch
...
}
if (!
finish();
}
})
.setNegativeButton("NO",
new DialogInterface.OnClickListener() {
@Override
public void
onClick(DialogInterface dialogInterface, int i) {
});
dialog = alertBuilder.create();
dialog.show();
drawer.closeDrawer(GravityCompat.START);
return true;
}
default:
drawer.closeDrawer(GravityCompat.START);
return false;
}
});
}

// Preparing the list data
private void prepareListData() {
listDataHeader = new ArrayList<>();
listDataChild = new HashMap<>();

// Adding child data
listDataHeader.add("SO Harian");
listDataHeader.add("Option");

// Adding child data
List<String> so_harian = new ArrayList<>();
so_harian.add("Per Lokasi");

```

```

...

List<String> option = new ArrayList<>();
...

listDataChild.put(listDataHeader.get(0), so_harian);
listDataChild.put(listDataHeader.get(1), option);
}

class BackPro extends AsyncTask<String, Void, String> {

    protected String doInBackground(String... params) {
        String url = params[0];

        ret = con.connectToWebService(url);

        if (!ret.equals("Connection Failed")) {
            return ret;
        } else {
            return "Connection Failed";
        }
    }

    protected void onPostExecute(String ret) {
    }
}
}

Login.java
package com.example.stockopname;
import ...

public class Login extends AppCompatActivity {
    ...

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.form_login);

        session = new SessionManager...

        ...

        btn_cancel.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                finishAffinity();
            }
        });

        btn_login.setOnClickListener(new View.OnClickListener() {

```

```

@Override
public void onClick(View v) {
    String userid =
txt_username.getEditableText().toString().toUpperCase();
    String userpassword =
txt_password.getEditableText().toString().toUpperCase();

    if (userid.length() > 0 && userpassword.length() >
0) {
        if (userid.toUpperCase().equals("ADM")) {
            if
(userpassword.toUpperCase().equals("ADMINIGR")) {
                txt_username.setText("");
                txt_password.setText("");
                session.createLoginSession(userid);
                session.createStation("HH");

                Intent intent = new Intent(Login.this,
Menu.class);
            intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                startActivity(intent);
                finish();
            } else {
                txt_username.setText("");
                txt_password.setText("");
                AlertDialog.Builder ...
            }
        } else if (userid.toUpperCase().equals("RST")) {
            if (userpassword.toUpperCase().equals("RST"))
{
                strIPAddress = "";

                ConnectivityManager connectivityManager =
(ConnectivityManager) getSystemService(CONNECTIVITY_SERVICE);
                for (Network network :
connectivityManager.getAllNetworks()) {
                    NetworkInfo networkInfo =
connectivityManager.getNetworkInfo(network);
                    if (networkInfo.isConnected()) {
                        LinkProperties linkProperties =
connectivityManager.getLinkProperties(network);

                            addressList =
linkProperties.getDnsServers();

                                for (int i = 0; i <
addressList.size(); i++) {
                                    strIPAddress =
addressList.get(i).toString().replace("/", "");

                                        if (!
strIPAddress.contains(":")) {
                                            break;

```

```

    }
    }
}

try {
    String str = url + "Logout?IP=" +
strIPAddress;

    ret = new GetReq().execute(str).get();
} catch (Exception e) {
    ...
}

if (!ret.equals("Connection Failed")) {
    if (ret.equals("Sukses")) {
        AlertDialog.Builder ...
    } else {
        AlertDialog.Builder alertBuilder =
new AlertDialog.Builder(Login.this);
        alertBuilder.setMessage("Gagal
membuka user pada HH ini")
            .setCancelable(false)
            .setPositiveButton("OK",
new DialogInterface.OnClickListener() {
                public void
onClick(DialogInterface dialog, int id) {
                }
            });
        AlertDialog dialog =
alertBuilder.create();
        dialog.show();
    }
    txt_username.setText("");
    txt_password.setText("");
} else {
    txt_username.setText("");
    txt_password.setText("");
    AlertDialog.Builder alertBuilder = new
AlertDialog.Builder(Login.this);
    alertBuilder.setMessage("Invalid Username
or Password")
        .setCancelable(false)
        .setPositiveButton("OK", new
DialogInterface.OnClickListener() {
                public void
onClick(DialogInterface dialog, int id) {
                }
            });
    AlertDialog dialog =
alertBuilder.create();
    dialog.show();
}
}

```



```

    } else {
        ConnectivityManager connectivityManager =
        (ConnectivityManager) getSystemService(CONNECTIVITY_SERVICE);
        for (Network network :
connectivityManager.getAllNetworks()) {
            NetworkInfo networkInfo =
connectivityManager.getNetworkInfo(network);
            if (networkInfo.isConnected()) {
                LinkProperties linkProperties =
connectivityManager.getLinkProperties(network);

                addressList =
linkProperties.getDnsServers();

                for (int i = 0; i <
addressList.size(); i++) {
                    strIPAddress =
addressList.get(i).toString().replace("/", "");

                    if (!
strIPAddress.contains(":")) {
                        break;
                    }
                }

                try {
                    String str = url + "Login.php?u=" +
userid + "|" + userpassword +
                    "|" + strIPAddress;
                    ret = new GetReq().execute(str).get();
                } catch (Exception e) {
                    ...
                }

                if (!ret.equals("Connection Failed")) {
                    try {
                        if (ret.substring(0,
1).equals("[")) {
                            jobj = new JSONArray(ret);
                        } else {
                            jsonObject = new
JSONObject(ret);

                            jobj = new JSONArray();
                            jobj.put(jsonObject);
                        }

                        for (int i = 0; i < jobj.length();
i++) {
                            jsonObject =

                            if

```

```

(jsonObject.getString("ACCESSNAME").equals("WRONGPASSWORD")) {
    AlertDialog.Builder
    alertBuilder = new AlertDialog.Builder(Login.this);

    alertBuilder.setMessage("Invalid Username or Password")
        .setCancelable(false)
        .setPositiveButton(
            "OK", new DialogInterface.OnClickListener() {
                public void
                onClick(DialogInterface dialog, int id) {
                    }
                });
    AlertDialog dialog =
    alertBuilder.create();
    dialog.show();
    txt_username.setText("");
    txt_password.setText("");
    return;
    } else if
(jsonObject.getString("ACCESSNAME").equals("OTHERUSER")) {
    AlertDialog.Builder
    alertBuilder = new AlertDialog.Builder(Login.this);

    alertBuilder.setMessage("User sedang login di device lain")
        .setCancelable(false)
        .setPositiveButton(
            "OK", new DialogInterface.OnClickListener() {
                public void
                onClick(DialogInterface dialog, int id) {
                    }
                });
    AlertDialog dialog =
    alertBuilder.create();
    dialog.show();
    txt_username.setText("");
    txt_password.setText("");
    return;
    } else if
(jsonObject.getString("ACCESSNAME").equals("OTHERDEVICE")) {
    AlertDialog.Builder
    alertBuilder = new AlertDialog.Builder(Login.this);

    alertBuilder.setMessage("User sedang login di device lain")
        .setCancelable(false)
        .setPositiveButton(
            "OK", new DialogInterface.OnClickListener() {
                public void
                onClick(DialogInterface dialog, int id) {
                    }
                });
    AlertDialog dialog =

```

```

alertBuilder.create();

dialog.show();
txt_username.setText("");
txt_password.setText("");
return;
} else if
(jsonObject.getString("ACCESSNAME").equals("REGISTERIP")) {
    AlertDialog.Builder
alertBuilder = new AlertDialog.Builder(Login.this);

alertBuilder.setMessage("IP Address tidak terdaftar : " +
ipWebService)

                .setCancelable(false)
                .setPositiveButton
("OK", new DialogInterface.OnClickListener() {
                public void
onClick(DialogInterface dialog, int id) {
                }
                });
AlertDialog dialog =
alertBuilder.create();
dialog.show();
txt_username.setText("");
txt_password.setText("");
return;
} else if
(jsonObject.getString("ACCESSNAME").equals("NOACCESS")) {
    AlertDialog.Builder
alertBuilder = new AlertDialog.Builder(Login.this);

alertBuilder.setMessage("Anda tidak memiliki akses untuk
menggunakan Handheld")

                .setCancelable(false)
                .setPositiveButton
("OK", new DialogInterface.OnClickListener() {
                public void
onClick(DialogInterface dialog, int id) {
                }
                });
AlertDialog dialog =
alertBuilder.create();

dialog.show();
txt_username.setText("");
txt_password.setText("");
return;
} else {
txt_username.setText("");
txt_password.setText("");

session.createLoginSession(userid);

session.createStation(jsonObject.getString("KODESTATION"));

```

```

session.createAccList(jobj.toString());

Intent(Login.this, Menu.class);

Intent intent = new
Intent(Login.this, Menu.class);

intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
startActivity(intent);
finish();
}
}
} catch (Exception e) {
...
}
} else {
AlertDialog.Builder alertBuilder = new
AlertDialog.Builder(Login.this);
alertBuilder.setMessage("Connection
Failed")
.setPositiveButton("OK", new
DialogInterface.OnClickListener() {
public void
onClick(DialogInterface dialog, int id) {
}
});
AlertDialog dialog =
alertBuilder.create();
dialog.show();
}
} else {
AlertDialog.Builder alertBuilder = new
AlertDialog.Builder(Login.this);
alertBuilder.setMessage("Username or Password
cannot Empty")
.setPositiveButton("OK", new
DialogInterface.OnClickListener() {
public void
onClick(DialogInterface dialog, int id) {
}
});
AlertDialog dialog = alertBuilder.create();
dialog.show();
}
}
});
}

class GetReq extends AsyncTask<String, Void, String> {

protected String doInBackground(String... params) {
String url = params[0];

ret = con.connectToWebService(url);
}
}

```

```

        if (!ret.equals("Connection Failed")) {
            return ret;
        } else {
            return "Connection Failed";
        }
    }

    protected void onPostExecute(String ret) {
    }
}
}

```

MdlPublic.java

```

package com.example.stockopname;
import ...

public class MdlPublic {
    private static final int nTambah = 57;
    String res = "";

    public String connectToWebService(String urlString) {
        try {
            URL url = new URL(urlString);
            HttpURLConnection con = (HttpURLConnection)
url.openConnection();

            con.setRequestProperty("Content-Type",
"application/x-www-form-urlencoded");
            con.setConnectTimeout(30000);
            con.setReadTimeout(30000);
            con.setRequestMethod("POST");

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

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

            return res;
        } catch (Exception e) {
            return res = "" + e;
        }
    }

    public String chartoPass(String strData) {
        int nHitung;
        String cOutput = "";
        byte[] ascii;

        strData = strData.trim();
        ascii = strData.getBytes(Charset.forName("Cp1252"));
    }
}

```

```

        for (nHitung = 0; nHitung < strData.length(); nHitung++) {
            cOutput += (char) ((ascii[nHitung] + nTambah) -
nHitung);
        }

        StringBuilder stringBuilder = new StringBuilder();
        stringBuilder.append(cOutput);
        stringBuilder = stringBuilder.reverse();

        return stringBuilder.toString();
    }

    public String passtoChar(String strData) {
        int nHitung;
        String cOutput = "";

        strData = strData.trim();

        StringBuilder stringBuilder = new StringBuilder();
        stringBuilder.append(strData);
        stringBuilder = stringBuilder.reverse();

        byte[] ascii =
stringBuilder.toString().getBytes(Charset.forName("Cp1252"));

        for (nHitung = 0; nHitung < ascii.length; nHitung++) {
            cOutput += (char) ((ascii[nHitung] + nHitung) - 57);
        }

        return cOutput;
    }
}

```

SessionManager.java

```

package com.example.stockopname;
import ...

public class SessionManager {
    private SharedPreferences pref;
    private Editor editor;
    private Context _context;
    MdlPublic con = new MdlPublic();
    private String res;

    // Sharedpref file name
    private static final String PREF_NAME = "AndroidHivePref";
    public static final String strIPAddress = "strIPAddress";
    public static final String ipWebService = "ipWebService";
    public static final String url = "http://" + ipWebService + "/"
WebService/";

    private static final String IS_LOGIN = "IsLoggedIn";
    public static final String UserID = "HH1";
    public static final String Station = "HH";
}

```

```

public static final String userAccess = "userAccess";
public static final String flagSetting = "0";

// Constructor
@SuppressLint("CommitPrefEdits")
public SessionManager(Context context){
    this._context = context;
    int PRIVATE_MODE = 0;
    pref = _context.getSharedPreferences(PREF_NAME,
PRIVATE_MODE);
    editor = pref.edit();
}

public void createLoginSession(String name){
    editor.putBoolean(IS_LOGIN, true);
    editor.putString(UserID, name);

    editor.commit();
}

public void createIPAddress(String ipVal){
    String newIPWebService = con.charToPass(ipVal);
    editor.putString(SessionManager.ipWebService,
newIPWebService);

    editor.commit();
}

public void createStation(String station){
    editor.putString(Station, station);

    editor.commit();
}

public void createAccList(String accList){
    editor.putString(SessionManager.userAccess, accList);

    editor.commit();
}

public void addFlagSetting(String value){
    editor.putString(SessionManager.flagSetting, value);

    editor.commit();
}

public void checkIP(){
    // Check login status
    if(pref.getString(ipWebService, null) == null ||
pref.getString(ipWebService, null).equals("")){
        createIPAddress("192.168.1.2");
    }
}
}

```

```

public void checkLogin(){
    // Check login status
    if(!this.isLoggedIn()){
        // user is not logged in redirect him to Login
Activity
        Intent i = new Intent(_context, Login.class);
        // Closing all the Activities
        i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK);

        // Add new Flag to start new Activity
        i.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);

        // Starting Login Activity
        _context.startActivity(i);
    }
}

private boolean isLoggedIn(){
    return pref.getBoolean(IS_LOGIN, false);
}

public HashMap<String, String> getUserDetails(){
    HashMap<String, String> user = new HashMap<>();
    user.put(UserID, pref.getString(UserID, null));
    user.put(userAccess, pref.getString(userAccess, null));

    return user;
}

public String getIPWebService(){
    res = pref.getString(ipWebService, null);
    res = con.passtoChar(res);
    return res;
}

public String getStation(){
    res = pref.getString(Station, null);
    return res;
}

public String getFlag(){
    return pref.getString(flagSetting, "0");
}

public void logoutUser(){
    // Clearing all data from Shared Preferences
    editor.remove(UserID);
    editor.remove(userAccess);
    editor.remove(IS_LOGIN);
    editor.commit();

    // After logout redirect user to Login Activity
    Intent i = new Intent(_context, Login.class);

```



```

        // Closing all the Activities
        i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);

        // Add new Flag to start new Activity
        i.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);

        // Staring Login Activity
        _context.startActivity(i);
    }
}

```

confi.php

```

/*$username = "username";
$password = "password";
$db = "(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)
(HOST=192.168.1.2) (PORT=8080)))
(CONNECT_DATA=(SID=databasename)))";*/

$dbhost = "localhost";
$dbuser = "root";
$dbpass = "";
$db = "dbstockopname";

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

mysqli_autocommit($con, false);

Login.php
include_once('confi.php');

$u = $_GET['u'];
$arr = explode("|", $u);
$user = strtoupper($arr[0]);
$pass = strtoupper($arr[1]);
$IP = strtoupper($arr[2]);
$accList = array();

//Cek apakah IP terdaftar atau tidak
$stmt = mysqli_query($con, "select * from tbdevice where upper(IP)
= '$IP'");
$dt = mysqli_fetch_assoc($stmt);

if ($dt == false) {
    $con -> close();
    $rows = array('ACCESSNAME' => 'REGISTERIP', 'USERLEVEL' =>
'X');
    $json = json_encode($rows);
    echo $json;
    return $json;
}

```

```

//Cek apakah user sudah login di device lain
$stmt = mysqli_query($con, "select * from tbdevice where
upper(useraktif) = '$user'");
$dt = mysqli_fetch_assoc($stmt);

if ($dt != false) {
    $con -> close();
    $rows = array('ACCESSNAME' => 'OTHERDEVICE', 'USERLEVEL' =>
'X');
    $json = json_encode($rows);
    echo $json;
    return $json;
}

//Cek apakah ada user yang sedang login
$stmt = mysqli_query($con, "select * from tbdevice where upper(IP)
= '$IP' and useraktif is not null");
$dt = mysqli_fetch_assoc($stmt);

if ($dt != false) {
    $con -> close();
    $rows = array('ACCESSNAME' => 'OTHERUSER', 'USERLEVEL' =>
'X');
    $json = json_encode($rows);
    echo $json;
    return $json;
}

//Cek apakah user memiliki akses HandHeld
$stmt = mysqli_query($con, "select a.accessname as ACCESSNAME from
tbuseraccess u, tbaccess a
                                where u.userid = '$user' and
u.accesscode = a.accesscode");
$dt = mysqli_fetch_assoc($stmt);
if ($dt == false) {
    $con -> close();
    $rows = array('ACCESSNAME' => 'NOACCESS', 'USERLEVEL' => 'X');
    $json = json_encode($rows);
    echo $json;
    return $json;
}

//Cek apakah username atau password sesuai
$stmt = mysqli_query($con, "select * from tbuser
                                where upper(userid) = '$user'
and upper(userpassword) = '$pass'");
$dt = mysqli_fetch_assoc($stmt);

if ($dt != false) {
    try {
        mysqli_query($con, "UPDATE tbdevice SET USERAKTIF =
'$user' WHERE IP = '$IP'");
        mysqli_commit($con);
    } catch (Exception $e) {

```

```

    ...
}
$stmt = mysqli_query($con, "select * from tbdevice where ip =
'$IP'");
$dtComp = mysqli_fetch_assoc($stmt);

$stmt = mysqli_query($con, "select a.accessname as ACCESSNAME,
s.userlevel as USERLEVEL
                                from tbuseraccess u,
tbuser s, tbaccess a
                                where u.userid = s.userid
and s.userid = '$user' and u.accesscode = a.accesscode");

while($rows = mysqli_fetch_assoc($stmt)) {
    $accList[] = array('ACCESSNAME' => $rows['ACCESSNAME'],
'USERLEVEL' => $rows['USERLEVEL'],
                                'KODESTATION' => "HH");
}
$con -> close();
$json = json_encode($accList);

echo $json;
return $json;
} else {
    mysqli_rollback($con);
    $con -> close();
    $dt = array('ACCESSNAME' => 'WRONGPASSWORD', 'USERLEVEL' =>
'X');
    $json = json_encode($dt);
    echo $json;
return $json;
}

```

Logout.php

```

include_once('confi.php');

$IP = $_GET['IP'];

try {
    $stmt = mysqli_query($con, "UPDATE tbdevice SET USERAKTIF =
NULL WHERE IP = '$IP'");
    mysqli_commit($con);

    $con -> close();
    echo "Sukses";
    return "Sukses";
} catch (Exception $e) {
    ...
}

```



FAKULTAS ILMU KOMPUTER
PROGRAM STUDI TEKNIK INFORMATIKA
UNIKA SOEGIJAPRANATA
BIMBINGAN PROJECT
SEMESTER GASAL / GENAP TAHUN AJARAN 2019 / 2020

TANGGAL	CATATAN	TANDA TANGAN
11/09/2019	Membuat dan Mengumpulkan Research outline	
12/09/2019	Membuat dan mengumpulkan Literature review	
23/09/2019	Melakukan analisa data set yang akan dipakai selama penelitian	
30/09/2019	Membuat dan menyimpulkan Report dari analisa dataset	
07/10/2019	Membuat alur sistem yang akan dipakai selama penelitian	



FAKULTAS ILMU KOMPUTER
PROGRAM STUDI TEKNIK INFORMATIKA
UNIKA SOEGIJAPRANATA
BIMBINGAN PROJECT
SEMESTER GASAL / GENAP TAHUN AJARAN 2019 / 2020

TANGGAL	CATATAN	TANDA TANGAN
14/10/2019	Membuat dan mengumpulkan laporan Bab 1	
21/10/2019	Melakukan Revisi Bab 1 jarak space dan format tulisan	
28/10/2019	Membuat dan mengumpulkan laporan Bab 2 berdasarkan literature review yang sudah ada	
4/11/2019	Melakukan Revisi terhadap format penulisan kutipan yang ada di Bab 2	
11/11/2019	Membuat dan mengumpulkan laporan Bab 3	



FAKULTAS ILMU KOMPUTER
PROGRAM STUDI TEKNIK INFORMATIKA
UNIKA SOEGIJAPRANATA
BIMBINGAN PROJECT
SEMESTER GASAL / GENAP TAHUN AJARAN 2019 / 2020

TANGGAL	CATATAN	TANDA TANGAN
18/11/2019	Membuat dan mengumpulkan laporan Bab 4 serta mempersiapkan presentasi sidang 2	
27/11/2019	Melakukan Revisi Bab 4 Sub Bab analysis karena belum tepat	
2/12/2019	Menambahkan penjelasan pada Bab 4 Sub Bab Design pada beberapa ilustrasi/gambar	
9/12/2019	Membuat dan mengumpulkan laporan Bab 5	
16/12/2019	Melakukan revisi Bab 5 Sub Bab implementation dengan menambahkan source code dan keterangannya	



FAKULTAS ILMU KOMPUTER
PROGRAM STUDI TEKNIK INFORMATIKA
UNIKA SOEGIJAPRANATA
BIMBINGAN PROJECT
SEMESTER GASAL / GENAP TAHUN AJARAN 2019 / 2020

TANGGAL	CATAIAN	TANDA TANGAN
4/1/2020	Membuat dan mengumpulkan laporan Bab 6	
8/1/2020	Membuat appendix dengan menambahkan source code lainnya serta mempersiapkan sidang akhir	
10/1/2020	Menggabungkan semua laporan dan membuat jurnal	
12/1/2020	Melakukan revisi jurnal sesuai format IEEE	

Submission author:
16k10041 YESAYA BILLY TANDRIANTO

Check ID:
15994690

Check date:
17.01.2020 10:42:00 GMT+0

Check type:
Doc vs Internet + Library

Report date:
17.01.2020 23:29:05 GMT+0

User ID:
29158



File name: 16.K1.0041_Yesaya Billy Tandrianto.docx

File ID: 20296904 Page count: 11 Word count: 4128 Character count: 24630 File size: 39.63 KB

2.81% Matches

Highest match: 0.56% with source <https://stackoverflow.com/questions/12197608/which-android-barcode-scanner-app-is-best-fo>

2.81% Internet Matches	20	Page 13
0.97% Library matches	13	Page 13

1.55% Quotes

Quotes	3	Page 14
No references found		

0% Exclusions

No exclusions found

Replacement

No replaced characters found

