

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Hasil Pengumpulan Data**

Pengumpulan data merupakan langkah penting suatu penelitian dan pengembangan sistem, dengan teknik pengumpulan data dapat menunjang proses penelitian yang akan diteliti, maka dalam rangka mencapai tujuan penelitian, pengumpulan data harus dengan fakta – fakta yang ada dengan teliti dan benar. Pada analisis kebutuhan sistem teknik pengumpulan data berfokus pada observasi dan wawancara.

##### **1.1.1 Hasil Observasi**

Berdasarkan pengamatan langsung aktifitas apa saja yang dilakukan oleh petugas di Warung Telkom Azzainiyah tentang bagaimana sistem yang ada agar dapat memahami langkah- langkah apa saja yang dilakukan, Sehingga sistem informasi yang akan dirancang berjalan dengan baik dan sesuai keinginan.

Dari hasil observasi yang dilakukan di Warung Telkom Azzainiyah penginputan data pemasukan masih dicatat dengan menggunakan buku pembukuan dan pengajuan pengeluaran harus menggunakan pihak ketiga yaitu koordinator untuk menghubungkan kepada *owner* wartel sehingga petugas wartel masih harus menunggu persetujuan koordinator, permasalahan tersebut memperlama pelayanan wartel serta dapat menyebabkan hilangnya data disebabkan buku-buku tidak tersusun rapi.

##### **4.1.2 Hasil Wawancara**

Wawancara dilakukan secara langsung kepada bagian petugas wartel dan koordinator wartel, yang membahas bagaimana alur sistem pengajuan pengeluaran kebutuhan dan pemasukan yang terjadi. Berikut kesimpulan hasil wawancara dengan pihak –pihak yang terkait:

1. Sistem pemasukan keuangan yang ada di wartel masih dicatat secara pembukuan.

2. Proses pengajuan masih menggunakan kertas seperti list pemasukan perbulan dan pengeluaran yang ada di wartel
3. Pengajuan kekurangan dan pendapatan wartel harus menggunakan pihak ketiga seperti koordinator wartel untuk diajukan kepada *owner* wartel.

### **1.1.2 Studi Literatur**

Hasil Pada tahapan ini membandingkan penelitian dengan penelitian-penelitian sebelumnya guna membuat sistem informasi yang dapat mengatasi kendala pada sistem informasi monitoring wartel. Sebagai referensi literatur sejenis atau penelitian terkait terdapat di awal

## **BAB II.**

### **4.2 Hasil Analisis dan Desain**

#### **4.2.1 Analisis Kebutuhan Fungsional**

Kebutuhan fungsional merupakan jenis kebutuhan yang berisi tentang proses-proses apa saja yang nantinya dilakukan sistem. Kebutuhan fungsional juga berisikan tentang informasi-informasi apa saja yang harus ada dan dihasilkan oleh sistem.

Berikut ini adalah kebutuhan fungsional dari perancangan aplikasi yang akan dibuat adalah sebagai berikut:

1. Aplikasi android akan menampilkan laporan pemasukan dan laporan pengeluarannya.
2. Aplikasi android dapat menampilkan data karyawan.
3. Aplikasi android dapat mencetak laporan hasil laba dan rugi.
4. Aplikasi android dapat melihat grafik pemasukan dan pengeluaran perbulannya.

#### **1.2.2 Analisis Kebutuhan Non- Fungsional**

Kebutuhan Non- Fungsional merupakan kebutuhan yang dapat digunakan berupa perangkat yang dibutuhkan sistem dan terbagi dalam berbagai hal untuk pengembangan atau penggunaannya.

##### **a. Kebutuhan Perangkat Keras(*Hardware*)**

Untuk mengetahui secara jelas perangkat yang dibutuhkan serta

mendukung proses pengembangan dan penggunaan dari sistem yang akan dibuat.

**Tabel 4. 1** Analisis Kebutuhan Non- Fungsional

No.	Kebutuhan Perangkat Keras Pengembangan	Kebutuhan Perangkat Keras Penggunaan
1.	Processor (R) Core (TM) i3-6006U CPU @2.00GHz 2.00GHz	Samsung A20S Versi 10
2.	Memory (RAM) 4GB	Memory (RAM) 4GB

b. Kebutuhan Perangkat Lunak (*Software*)

Adalah suatu program yang nantinya akan diperlukan untuk melakukan proses intruksi atau menjalankan perangkat keras. Jadi memerlukan suatu perangkat lunak agar aplikasi yang dirancang dan diimplementasikan sesuai dengan apa yang direncanakan. Berikut adalah tabel yang dibutuhkan:

**Tabel 4. 2** Kebutuhan Perangkat Lunak (*Software*)

No.	Kebutuhan Perangkat Keras Pengembangan	Kebutuhan Perangkat Lunak Penggunaan
1.	Visual Studio Code	<i>Google Chrome</i>
2.	Android Studio	<i>Android 4.4 (Kitkat)</i>
3.	MySQL Database Server Local	<i>Browser</i>

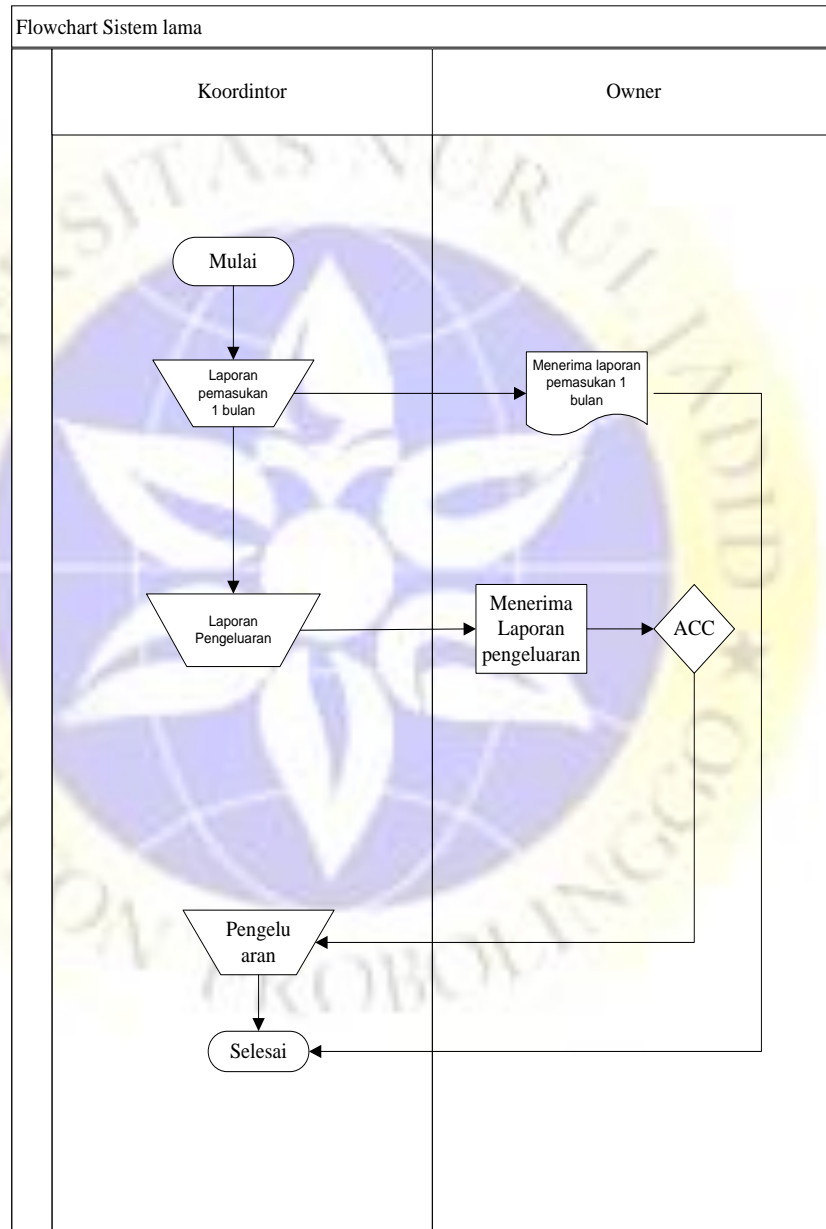
### 1.2.3 Desain Sistem

Setelah dilakukannya analisis kebutuhan sistem, selanjutnya yaitu desain sistem yang dilakukan untuk mengoptimisasi sistem informasi monitoring warung Telkom wilayah Azzainiyah dengan Android studio. Berdasarkan analisa diatas diperlukan suatu perancangan aplikasi yang bisa diberikan kemudahan dalam proses pelaporan dan pengajuan wartel . ada beberapa perangkat dalam perancangan sistem yang digunakan pada penelitian ini yaitu: *flowchart*, *Data flow Diagram (DFD)*, dan *Entity Relationship Diagram (ERD)*.

**a. Flowchart**

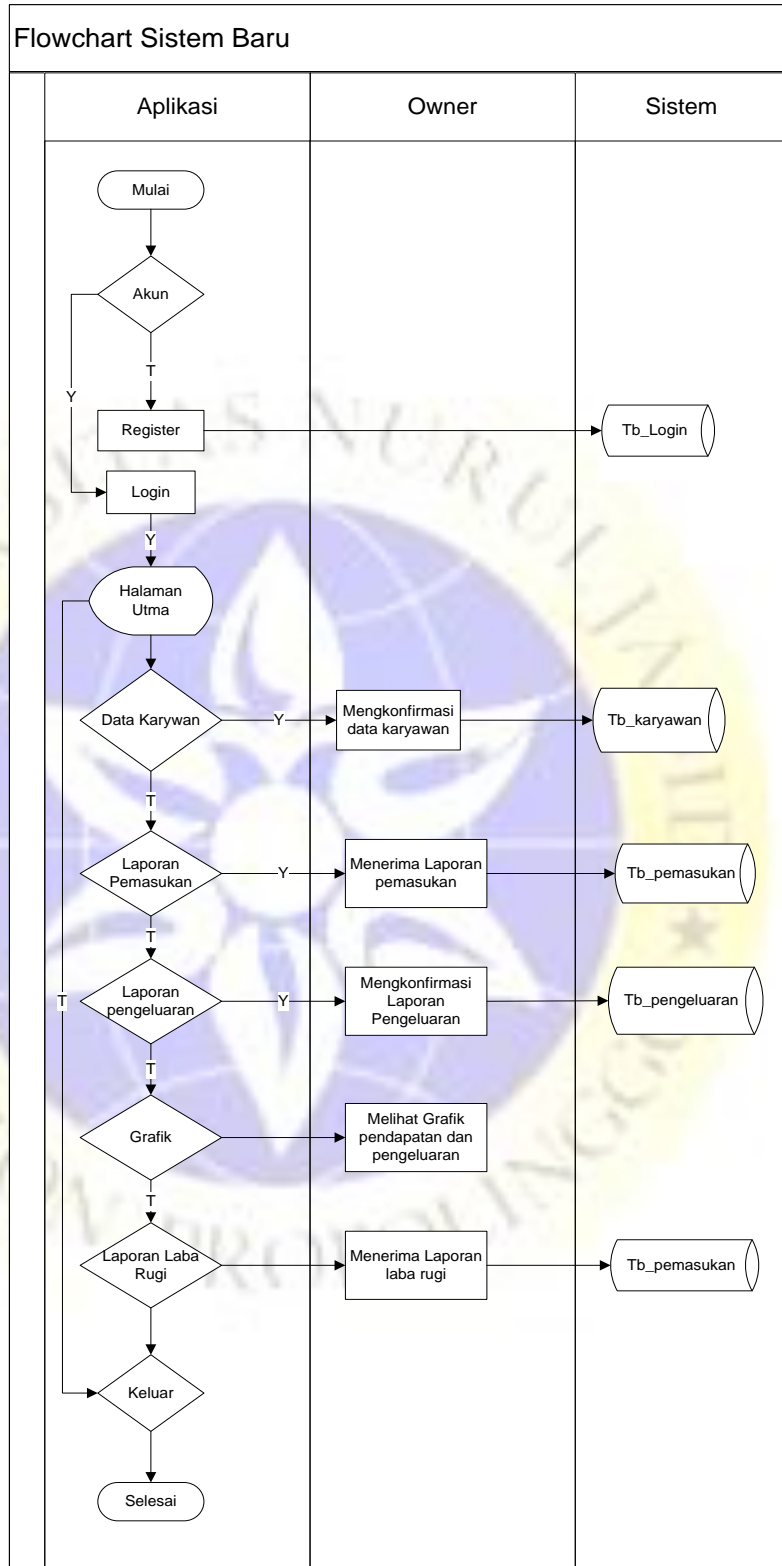
Flowchart ini berfungsi untuk menggambarkan atau mendefinisikan alur kerja dari suatu sistem yang berjalan dari sistem lama, sistem baru dan bagaimana cara kerja dari aplikasi yang telah direncanakan.

**1. Flowchart Sistem Lama**



**Gambar 4. 1**Flowchart sistem lama koordinator ke *owner*

## 2. Flowchart Sistem Baru

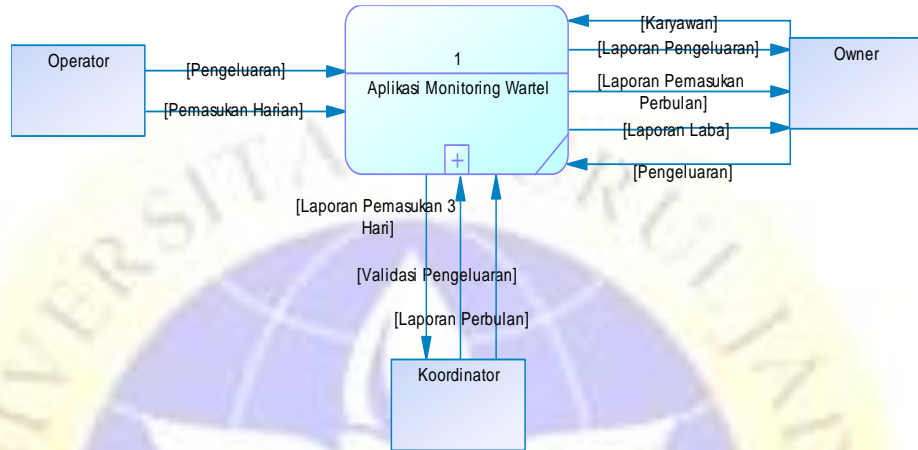


**Gambar 4. 2**Flowchart sistem baru

## b. Data Flow Diagram

DFD suatu diagram atau alur sebuah sistem, untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan dengan database, terdapat beberapa level dalam perancangan sebagai berikut:

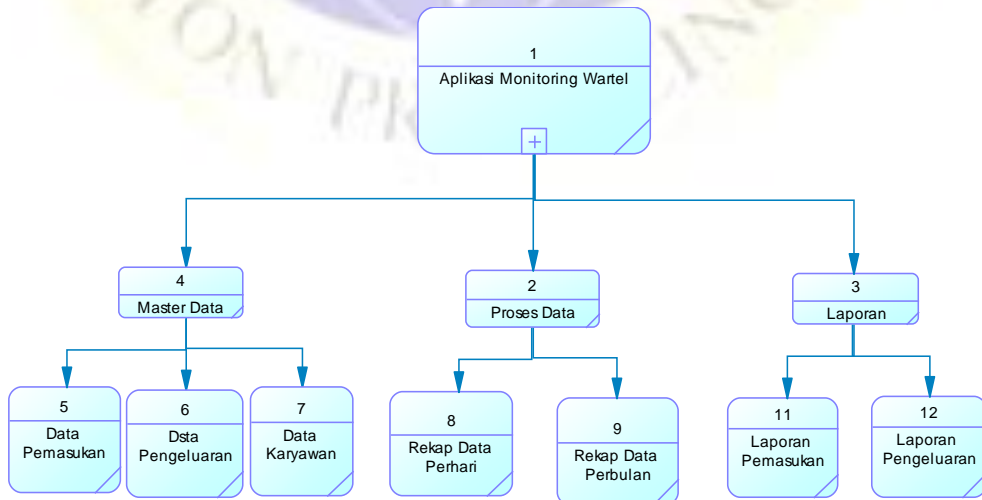
### 1. Context Diagram



**Gambar 4.3** Context Diagram

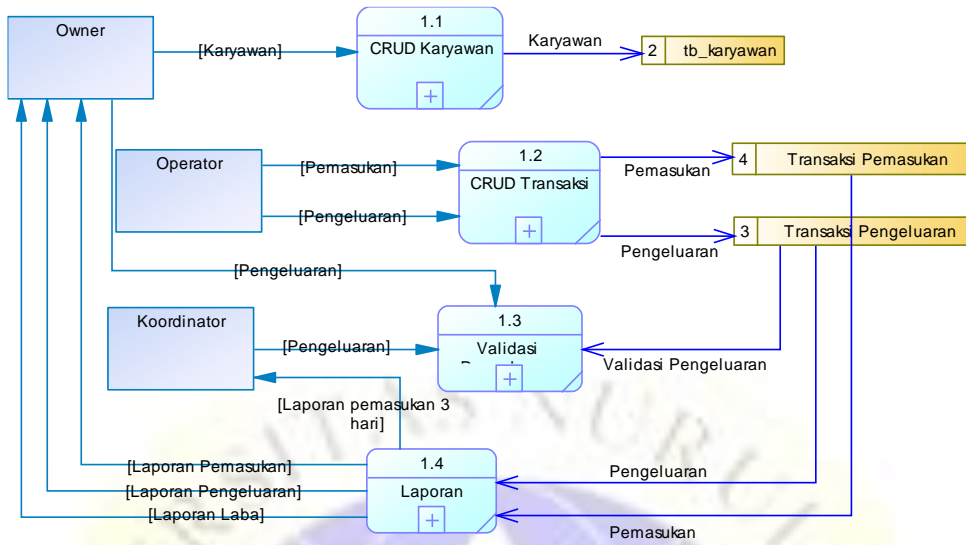
### 2. Bagan Berjenjang

Bagan berjenjang merupakan bagian dari suatu DFD yang menjelaskan menjelaskan semua proses yang ada dalam sebuah sistem, lebih tepatnya mempermudah penggambaran data flow diagram level-level yang lebih bawah lagi. Dibawah ini desain bagan berjenjang aplikasi monitoring sebagai berikut :



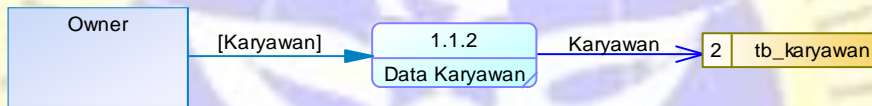
**Gambar 4.4** Bagan berjenjang

**c. DFD Level 1**



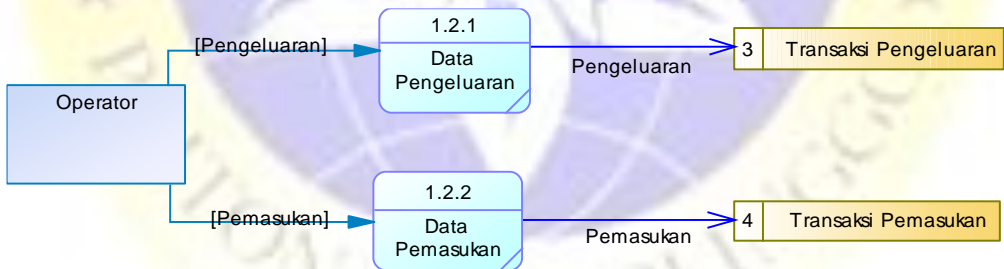
**Gambar 4. 5DFD Level 1**

**d. DFD level 1 Proses 1**



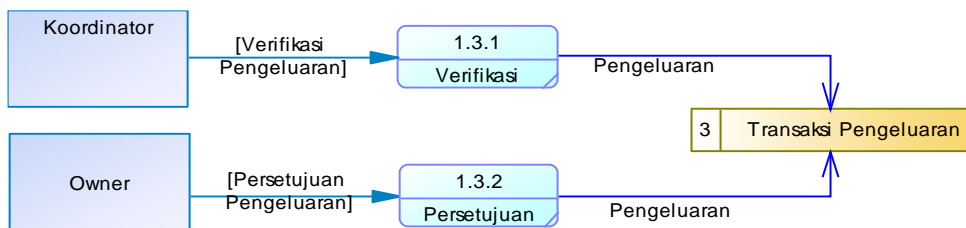
**Gambar 4. 6DFD level 1 proses 1**

**e. DFD level 1 proses 2**



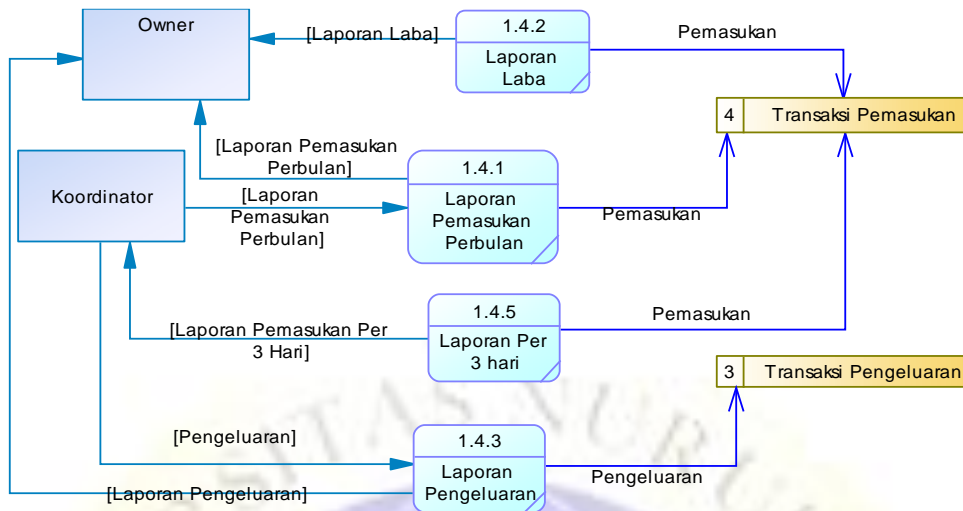
**Gambar 4. 7DFD level 1 proses 2**

**f. DFD level 1 proses 3**



**Gambar 4. 8DFD level 1 proses 3**

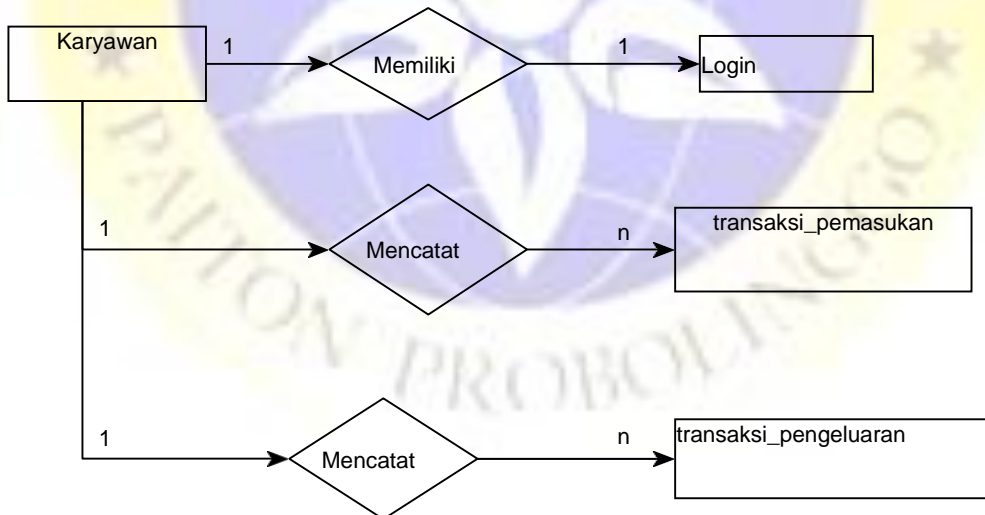
**g. DFD level 1 proses 4**



**Gambar 4. 9**DFD level 1 proses 4

**h. Entity Relationship Diagram**

ERD menjelaskan hubungan antar objek-objek data yang mempunyai hubungan antar relasi, supaya lebih mudah dalam menggambarkan data yang memiliki hubungan atau relasi dalam bentuk sebuah desain.



**Gambar 4. 10**Entity Relationship Diagram



## Kamus Data

tb\_karyawan : id\_karyawan, nama\_karyawan,  
tgl\_lahir, alamat, no\_hp

laporan harian : id\_day, tanggal\_harian, hasil, status.

Login : id\_login, username, password, level,  
id\_karyawan

transaksi\_pemasukan : id\_transaksi, tgl\_transaksi, biaya, no\_hp.

transaksi\_pengeluaran : id\_transaksi\_pengeluaran, tgl\_pengeluaran,  
bahan\_pengeluaran,  
status, : keterangan, biaya\_pengeluaran.

### 1.2.4 Desain database

Database kumpulan data yang saling berhubungan

#### 1. Tabel Karyawan

**Tabel 4.3** Tabel Karyawan

Nama Field	Tipe data	Key
Id_karyawan	Int(5)	Primary Key
Nama_karyawan	Varchar(50)	
tgl_lahir	Date	
Alamat	Varchar(50)	
No hp	Varchar(50)	

#### 2. Tabel Laporan Harian

**Tabel 4.4** Tabel Laporan Harian

Nama Field	Tipe data	Key
Id_day	Int(5)	Primary Key
tanggal_harian	Date	
Hasil	Int(10)	
Status	Enum('TERKIRIM' 'BELUM' 'BELUM')	

### 3. Tabel Login

**Tabel 4. 5**Tabel Login

Nama <i>Field</i>	Tipe data	Key
id_login	Int(5)	Primary Key
Username	Varchar(50)	
Password	Varchar(50)	
level	Varchar(50)	
id_karyawan	Int(5)	

### 4. Tabel Transaksi Pemasukan

**Tabel 4. 6**Tabel Transaksi Pemasukan

Nama <i>Field</i>	Tipe data	Key
id_transaksi	Int(5)	Primary Key
tgl_transaksi	Date	
Biaya	Int(20)	
no_hp	Varchar(20)	

### 5. Tabel Transaksi Pengeluaran

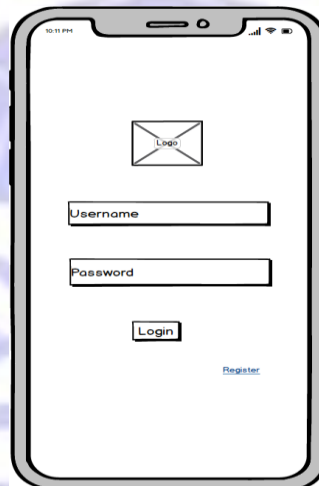
**Tabel 4. 7** Tabel Transaksi Pengeluaran

Nama <i>Field</i>	Tipe data	Key
id_transaksi_pengeluaran	Int(5)	Primary Key
tgl_pengeluaran	Date	
bahan_pengeluaran	Varchar(50)	
status	Varchar(50)	
keterangan	Text	
biaya_pengeluaran	Int(20)	

### 1.2.5 Desain interface

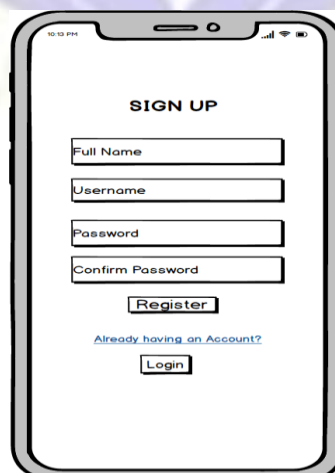
Desain Interface merupakan tahapan yang menggambarkan antarmuka aplikasi dimana pada tahap ini desain yang akan ditampilkan akan sedikit berbeda dengan program yang ada atau yang sebenarnya karena masih mempertimbangkan adanya fitur tambahan dan pengurangan fitur tetapi perbedaan antara aplikasi dan desain interface tidak terlalu berbeda dari segi pokok penggunaan. Berikut adalah rancangan tampilan pada aplikasi yang akan dibuat:

#### 1. Form Login

A mobile application login form displayed on a smartphone screen. At the top center is an envelope icon with the word "Logo" written below it. Below the logo are two input fields: "Username" and "Password". Underneath the "Password" field is a "Login" button. At the bottom right of the form is a "Register" link.

**Gambar 4. 11**Form Login

#### 2. Form Register

A mobile application registration form displayed on a smartphone screen. The title "SIGN UP" is centered at the top. Below the title are four input fields: "Full Name", "Username", "Password", and "Confirm Password". Underneath the "Confirm Password" field is a "Register" button. At the bottom of the form is a "Login" button, with a link "Already having an Account?" positioned above it.

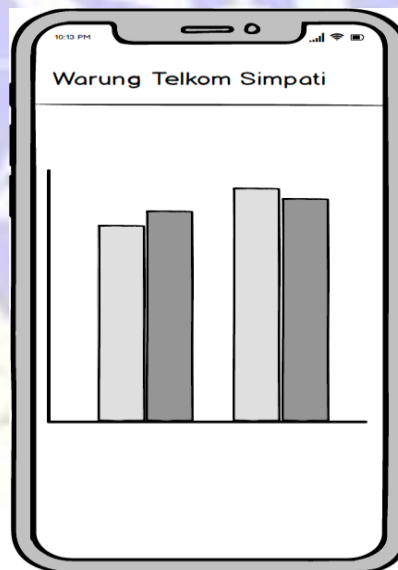
**Gambar 4. 12**Form register

### 3. Menu Utama



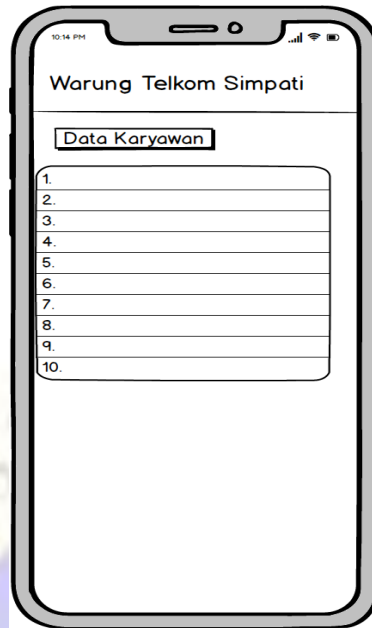
**Gambar 4. 13** Menu Utama

### 4. Menu Grafik



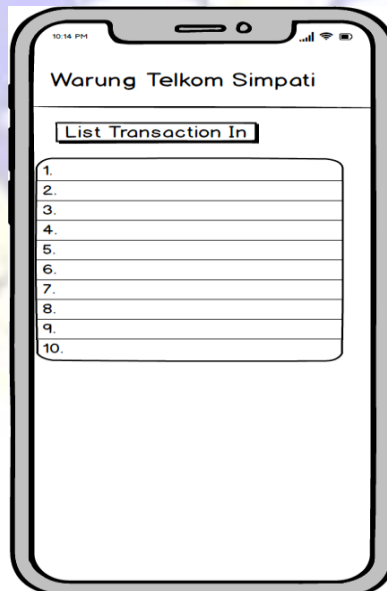
**Gambar 4. 14** Menu Grafik

5. Menu Data Karyawan



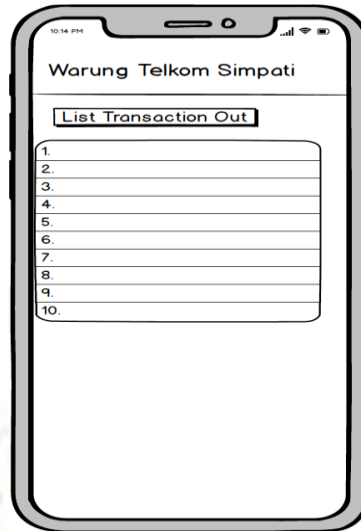
**Gambar 4. 15** Menu Data Karyawan

6. Menu Transaksi Pemasukan



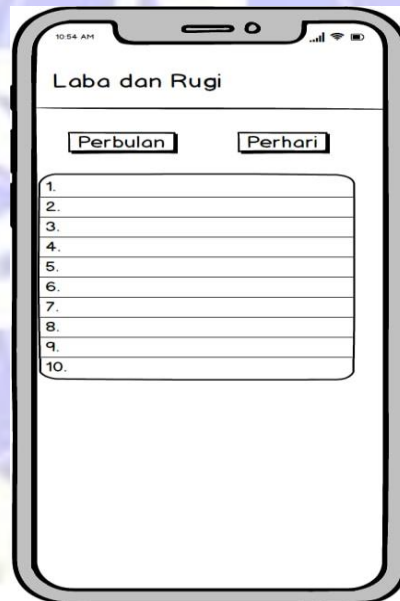
**Gambar 4. 16** Menu Data Pemasukan

7. Menu Transaksi Pengeluaran



**Gambar 4. 17** Menu Data Pengeluaran

8. Menu Laporan Laba dan Rugi



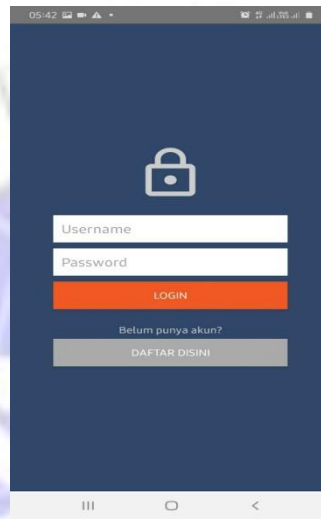
**Gambar 4. 18** Laporan Laba dan Rugi

### 1.3 Implementation

Implementasi merupakan hasil tampilan dari sistem baru. Berikut ini adalah tampilan hasil dari sistem

#### a. Halaman Login

Pada halaman ini *owner* menggunakan *user* yang telah disediakan agar bisa melakukan login yang nantinya harus mengisi *username* dan *password*.



**Gambar 4. 19**Halaman Login

```
package com.androidigniter.loginandregistration;

import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.androidigniter.loginandregistration.Api.BaseApiService;
import com.androidigniter.loginandregistration.Api.URL;
import com.androidigniter.loginandregistration.Utills.SessionHandler;

import org.json.JSONObject;
```

```

import okhttp3.ResponseBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class Login extends AppCompatActivity {

    private static final String KEY_STATUS = "status";
    private static final String KEY_FULL_NAME = "name";
    private static final String KEY_EMPTY = "";

    private EditText etUsername;
    private EditText etPassword;
    private String username;
    private String password;
    private ProgressDialog pDialog;
    private SessionHandler session;

    BaseApiService apiService;
    Boolean backDouble = false;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        session = new
SessionHandler(getApplicationContext());
        apiService = URL.getAPIService();

        if (session.isLoggedIn()) {
            loadDashboard();
        }
        setContentView(R.layout.activity_login);

        etUsername = findViewById(R.id.etLoginUsername);
        etPassword = findViewById(R.id.etLoginPassword);

        Button register =
findViewById(R.id.btnLoginRegister);
        Button login = findViewById(R.id.btnLogin);

        //Launch Registration screen when Register
Button is clicked
        register.setOnClickListener(v -> {
            Intent i = new Intent(Login.this,
Daftar.class);
            startActivity(i);
        });

        login.setOnClickListener(v -> {
            //Retrieve the data entered in the edit
texts

```



```

        username =
etUsername.getText().toString().toLowerCase().trim();
        password =
etPassword.getText().toString().trim();
        if (validateInputs()) {
            login();
        }
    });
}

@Override
public void onBackPressed() {
    if (backDouble) {
        Intent home = new
Intent(Intent.ACTION_MAIN);
        home.addCategory(Intent.CATEGORY_HOME);

home.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(home);
        return;
    }
    this.backDouble = true;
    Toast.makeText(getApplicationContext(), "Tekan
sekali lagi untuk kembali ke layar awal!",
Toast.LENGTH_SHORT).show();

        new Handler().postDelayed(() -> backDouble =
false, 2000);
    }
}

/**
 * Launch Dashboard Activity on Successful Login
 */
private void loadDashboard() {
    Intent i = new Intent(getApplicationContext(),
DashboardActivity.class);
    startActivity(i);
    finish();
}

/**
 * Display Progress bar while Logging in
 */
private void displayLoader() {
    pDialog = new ProgressDialog(Login.this);
    pDialog.setMessage("Sedang masuk.. Silahkan
tunggu...");
    pDialog.setIndeterminate(false);
    pDialog.setCancelable(false);
    pDialog.show();
}
}

```

```

private void login() {
    displayLoader();

    apiService.login(username, password).enqueue(new
    Callback<ResponseBody>() {
        @Override
        public void onResponse(Call<ResponseBody>
        call, Response<ResponseBody> response) {
            pDialog.dismiss();
            try {
                JSONObject result = new
                JSONObject(response.body().string());
                if
                (result.getString(KEY_STATUS).equals("1")) {
                    session.loginUser(username,
                    result.getJSONObject("user").getString(KEY_FULL_NAME));
                    loadDashboard();
                } else {

                Toast.makeText(getApplicationContext(), "Username dan
                password tidak dikenali!", Toast.LENGTH_SHORT).show();
                }
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        @Override
        public void onFailure(Call<ResponseBody>
        call, Throwable t) {
            pDialog.dismiss();
            Toast.makeText(getApplicationContext(),
            t.getMessage(), Toast.LENGTH_SHORT).show();
        }
    });
}

/**
 * Validates inputs and shows error if any
 */
private boolean validateInputs() {
    if (KEY_EMPTY.equals(username)) {
        etUsername.setError("Username cannot be
        empty");
        etUsername.requestFocus();
        return false;
    }
    if (KEY_EMPTY.equals(password)) {
        etPassword.setError("Password cannot be
        empty");
        etPassword.requestFocus();
        return false;
    }
}

```

```

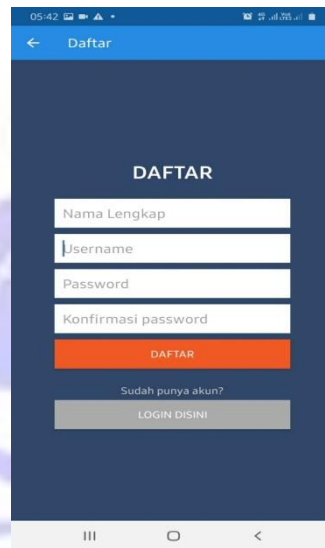
    }
    return true;
}
}

```

### Segmen Program 4.1 Login Admin

#### b. Halaman register

Pada halaman ini owner bisa membuat akun baru jika tidak memiliki akun.



**Gambar 4. 20** Halaman Register

```

package com.androidigniter.loginandregistration;

import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.androidigniter.loginandregistration.Api.BaseApiService;
import com.androidigniter.loginandregistration.Api.URL;
import com.androidigniter.loginandregistration.Utills.SessionHandler;

import org.json.JSONObject;

```

```

import okhttp3.ResponseBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class Daftar extends AppCompatActivity {

    private static final String KEY_STATUS =
"status";
    private static final String KEY_EMPTY = "";

    private EditText etUsername;
    private EditText etPassword;
    private EditText etConfirmPassword;
    private EditText etFullName;
    private String username;
    private String password;
    private String confirmPassword;
    private String fullName;
    private ProgressDialog pDialog;
    private SessionHandler session;

    BaseApiService apiService;

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

getSupportActionBar().setDisplayHomeAsUpEnabled(true)
;

getSupportActionBar().setDisplayShowHomeEnabled(true)
;
        getSupportActionBar().setTitle("Daftar");

        session = new
SessionHandler(getApplicationContext());
        apiService = URL.getAPIService();

        etUsername = findViewById(R.id.etUsername);
        etPassword = findViewById(R.id.etPassword);
        etConfirmPassword =
findViewById(R.id.etConfirmPassword);
        etFullName = findViewById(R.id.etFullName);

        Button login =
findViewById(R.id.btnRegisterLogin);
        Button register =
findViewById(R.id.btnRegister);

```

```

        //Launch Login screen when Login Button is
        clicked
        login.setOnClickListener(v -> finish());

        register.setOnClickListener(v -> {
            //Retrieve the data entered in the edit
            texts
                username =
            etUsername.getText().toString().toLowerCase().trim();
                password =
            etPassword.getText().toString().trim();
                confirmPassword =
            etConfirmPassword.getText().toString().trim();
                fullName =
            etFullName.getText().toString().trim();
            if (validateInputs()) {
                registerUser();
            }
        });
    }

    @Override
    public boolean onSupportNavigateUp() {
        onBackPressed();
        return true;
    }

    /**
     * Display Progress bar while registering
     */
    private void displayLoader() {
        pDialog = new ProgressDialog(Daftar.this);
        pDialog.setMessage("Sedang mendaftar..
        Silahkan tunggu...");
        pDialog.setIndeterminate(false);
        pDialog.setCancelable(false);
        pDialog.show();
    }

    /**
     * Launch Dashboard Activity on Successful Sign
        Up
     */
    private void loadDashboard() {
        Intent i = new
        Intent(getApplicationContext(),
        DashboardActivity.class);
        startActivity(i);
        finish();
    }

    private void registerUser() {

```

```

        displayLoader();

        apiService.daftar(username, password,
        fullName).enqueue(new Callback<ResponseBody>() {
            @Override
            public void onResponse(Call<ResponseBody>
            call, Response<ResponseBody> response) {
                pDialog.dismiss();
                try {
                    JSONObject result = new
                    JSONObject(response.body().string());
                    if
                    (result.getString(KEY_STATUS).equals("1")) {
                        session.loginUser(username,
                        fullName);
                        loadDashboard();
                    }
                    else if
                    (result.getString(KEY_STATUS).equals("2")) {
                        etUsername.setError("Username
                        telah digunakan, gunakan yang lain");
                        etUsername.requestFocus();
                    }
                    else {

                        Toast.makeText(getApplicationContext(), "Terjadi
                        kesalahan ketika mendaftar, silahkan coba lagi",
                        Toast.LENGTH_SHORT).show();
                    }
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }

            @Override
            public void onFailure(Call<ResponseBody>
            call, Throwable t) {
                pDialog.dismiss();

                Toast.makeText(getApplicationContext(),
                t.getMessage(), Toast.LENGTH_SHORT).show();
            }
        });
    }

    /**
     * Validates inputs and shows error if any
     */
    private boolean validateInputs() {
        if (KEY_EMPTY.equals(fullName)) {
            etFullName.setError("Nama lengkap tidak
            boleh kosong");
        }
    }
}

```

```

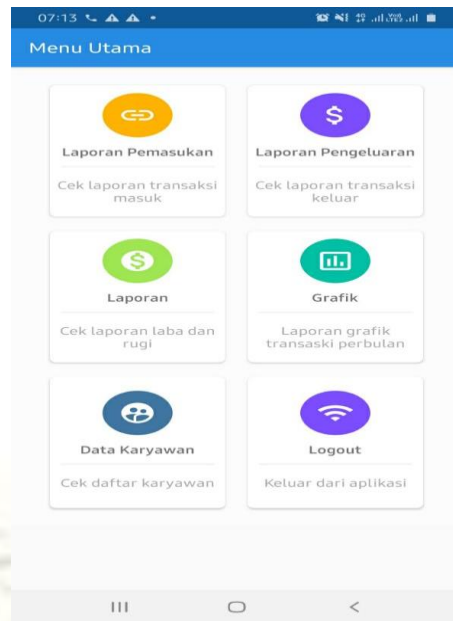
        etFullName.requestFocus();
        return false;
    }
    if (KEY_EMPTY.equals(username)) {
        etUsername.setError("Username tidak boleh
kosong");
        etUsername.requestFocus();
        return false;
    }
    if (KEY_EMPTY.equals(password)) {
        etPassword.setError("Password tidak boleh
kosong");
        etPassword.requestFocus();
        return false;
    }
    if (KEY_EMPTY.equals(confirmPassword)) {
        etConfirmPassword.setError("Konfirmasi
password anda");
        etConfirmPassword.requestFocus();
        return false;
    }
    if (!password.equals(confirmPassword)) {
        etConfirmPassword.setError("Password
tidak cocok");
        etConfirmPassword.requestFocus();
        return false;
    }
    return true;
}
}

```

#### Segmen Program 4.2 Register

##### c. Halaman Menu

Halaman ini akan tampil ketika Owner melakukan login dan akan ditampilkan beberapa menu diantaranya laporan pemasukan, laporan pengeluaran, Grafik, dan laporan laba dan rugi



**Gambar 4. 21**Halaman Menu

```

package com.androidigniter.loginandregistration;

import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import androidx.cardview.widget.CardView;

import com.androidigniter.loginandregistration.Model.User;
import com.androidigniter.loginandregistration.Utills.SessionHandler;
import com.google.firebase.iid.FirebaseInstanceId;
import com.google.firebase.messaging.FirebaseMessaging;

public class DashboardActivity extends AppCompatActivity
{

    private SessionHandler session;

    Boolean backDouble = false;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_dashboard);
        getSupportActionBar().setTitle("Menu Utama");
    }

```



```

        session = new
        SessionHandler(getApplicationContext());
        User user = session.getUserDetails();

        FirebaseMessaging.getInstance().subscribeToTopic("all");

        FirebaseInstanceId.getInstance().getInstanceId().addOnSu
        ccessListener(command ->
        System.out.println(command.getToken()));

        CardView logoutBtn =
        findViewById(R.id.btnLogout);
        CardView BulanBtn =
        findViewById(R.id.btn_monthly);
        CardView Btnlaba = findViewById(R.id.btn_Laba);
        CardView Btngrafik =
        findViewById(R.id.btn_grafik);
        CardView btn_karyawan =
        findViewById(R.id.btn_karyawan);
        CardView btn_transaction =
        findViewById(R.id.btn_transaction_out);

        logoutBtn.setOnClickListener(v -> {
            session.logoutUser();
            Intent i = new
            Intent(DashboardActivity.this, Login.class);
            startActivity(i);
            finish();
        });

        BulanBtn.setOnClickListener(v -> {
            Intent i = new
            Intent(DashboardActivity.this, TransactionIn.class);
            startActivity(i);
        });

        btn_transaction.setOnClickListener(v -> {
            Intent i = new
            Intent(DashboardActivity.this, TransactionOut.class);
            startActivity(i);
        });

        btn_karyawan.setOnClickListener(v -> {
            Intent i = new
            Intent(DashboardActivity.this, DataKaryawan.class);
            startActivity(i);
        });

        Btngrafik.setOnClickListener(v -> {
            Intent i = new
            Intent(DashboardActivity.this, Chart.class);
            startActivity(i);
        });

```

```

});

Btnlaba.setOnClickListener(v -> {
    Intent i = new
Intent(DashboardActivity.this, Laba.class);
    startActivity(i);
});
}

@Override
public void onBackPressed() {
    if (backDouble) {
        Intent home = new
Intent(Intent.ACTION_MAIN);
        home.addCategory(Intent.CATEGORY_HOME);

home.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(home);
        return;
    }
    this.backDouble = true;
    Toast.makeText(getApplicationContext(), "Tekan
sekali lagi untuk kembali ke layar awal!",
Toast.LENGTH_SHORT).show();

    new Handler().postDelayed(() -> backDouble =
false, 2000);
}
}

```

### Segmen Program 4.3 Halaman Menu

#### d. Form menu Pemasukan

Menu akan menampilkan data laporan pemasukan perbulannya



**Gambar 4. 22**Menu Transaction In

```
package com.androidigniter.loginandregistration;

import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.DefaultItemAnimator;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.androidigniter.loginandregistration.Adapter.AdapterTransactionIn;
import com.androidigniter.loginandregistration.Api.BaseApiService;
import com.androidigniter.loginandregistration.Api.URL;
import com.androidigniter.loginandregistration.Model.ItemBulanan;

import org.json.JSONArray;
import org.json.JSONObject;

import java.util.ArrayList;

import okhttp3.ResponseBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class TransactionIn extends AppCompatActivity
{
    RecyclerView lvhape;

    ArrayList<ItemBulanan> listItem = new
ArrayList<>();

    Context context;
    ProgressDialog progressDialog;
    BaseApiService apiService;
```

```

        AdapterTransactionIn adapter;

        @Override
        protected void onCreate(Bundle
savedInstanceState) {
            super.onCreate(savedInstanceState);

            setContentView(R.layout.activity_transaction_in);

            getSupportActionBar().setDisplayHomeAsUpEnabled(true)
;

            getSupportActionBar().setDisplayShowHomeEnabled(true)
;
            getSupportActionBar().setTitle("Transaction
In");

            apiService = URL.getAPIService();
            context = this;

            adapter = new AdapterTransactionIn(context,
listItem);
            lvhape = findViewById(R.id.Lvhape);
            lvhape.setLayoutManager(new
LinearLayoutManager(context));
            lvhape.setItemAnimator(new
DefaultItemAnimator());
            lvhape.setAdapter(adapter);
        }

        @Override
        protected void onResume() {
            super.onResume();
            getData();
        }

        private void getData() {
            listItem.clear();
            displayLoader();

            new Handler().postDelayed(() ->
apiService.statsBulanan().enqueue(new
Callback<ResponseBody>() {
                @Override
                public void onResponse(Call<ResponseBody>
call, Response<ResponseBody> response) {
                    pDialog.dismiss();
                    try {
                        JSONObject object = new
JSONObject(response.body().string());

                        JSONArray array =

```

```

object.getJSONArray("result");
    for (int i = 0; i <
array.length(); i++) {
        JSONObject list =
(JSONObject) array.get(i);

        ItemBulanan item = new
ItemBulanan();

item.setTime(list.getString("time"));

item.setBiaya(list.getString("biaya"));

item.setBulan(list.getString("bulan"));
        listItem.add(item);
    }
    adapter.notifyDataSetChanged();
} catch (Exception e) {
    e.printStackTrace();
}
}

@Override
public void onFailure(Call<ResponseBody>
call, Throwable t) {
    pDialog.dismiss();
    Toast.makeText(context,
t.getMessage(), Toast.LENGTH_SHORT).show();
}
}), 300);
}

public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the
action bar if it is present.
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.optionmenu, menu);
    return true;
}

public boolean onOptionsItemSelected(MenuItem
item) {
    if (item.getItemId() == R.id.t_in) {
        startActivity(new Intent(this,
TransactionIn.class));
        finish();
    } else if (item.getItemId() == R.id.t_out) {
        startActivity(new Intent(this,
TransactionOut.class));
        finish();
    } else if (item.getItemId() == R.id.employee)

```

```

{
    startActivity(new Intent(this,
DataKaryawan.class));
    finish();
    } else if (item.getItemId() == R.id.Logout) {
startActivity(new Intent(this,
Login.class));
    finish();
    } else if (item.getItemId() ==
android.R.id.home) {
    finish();
    return true;
    }
    return true;
}

private void displayLoader() {
    pDialog = new ProgressDialog(context);
    pDialog.setMessage("Sedang memuat
daftar...");
    pDialog.setIndeterminate(false);
    pDialog.setCancelable(false);
    pDialog.show();
}
}

```

#### Segmen Program 4.4 Transaksi Pemasukan

```

package com.androidigniter.loginandregistration;

import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.view.MenuItem;
import android.widget.Toast;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.DefaultItemAnimator;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.androidigniter.loginandregistration.Adapter.AdapterT
ransactionInDetail;
import
com.androidigniter.loginandregistration.Api.BaseApiServi
ce;
import com.androidigniter.loginandregistration.Api.URL;
import

```

```
com.androidigniter.loginandregistration.Model.ItemBulana
nList;

import org.json.JSONArray;
import org.json.JSONObject;

import java.util.ArrayList;

import okhttp3.ResponseBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class TransactionInDetail extends
AppCompatActivity {

    RecyclerView lvhape;

    ArrayList<ItemBulananList> listItem = new
ArrayList<>();

    Context context;
    ProgressDialog pDialog;
    BaseApiService apiService;
    AdapterTransactionInDetail adapter;

    @Override
    protected void onCreate(@Nullable Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

setContentView(R.layout.activity_transaction_in_detail);

getSupportActionBar().setDisplayHomeAsUpEnabled(true);

getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setTitle("Transaction
Detail");

        apiService = URL.getAPIService();
        context = this;

        adapter = new
AdapterTransactionInDetail(context, listItem);
        lvhape = findViewById(R.id.lvhape);
        lvhape.setLayoutManager(new
LinearLayoutManager(context));
        lvhape.setItemAnimator(new
DefaultItemAnimator());
        lvhape.setAdapter(adapter);
    }
}
```

```

@Override
protected void onResume() {
    super.onResume();
    getData();
}

private void getData() {
    Intent detail = getIntent();
    String bulan = detail.getStringExtra("bulan");

    listItem.clear();
    displayLoader();

    new Handler().postDelayed(() ->
    apiService.getBulananList(bulan).enqueue(new
    Callback<ResponseBody>() {
        @Override
        public void onResponse(Call<ResponseBody>
    call, Response<ResponseBody> response) {
            pDialog.dismiss();
            try {
                JSONObject object = new
    JSONObject(response.body().string());
                JSONArray array =
    object.getJSONArray("data");
                for (int i = 0; i < array.length();
    i++) {
                    JSONObject list = (JSONObject)
    array.get(i);
                    ItemBulananList item = new
    ItemBulananList();
                    item.setTgl_transaksi(list.getString("tgl_transaksi"));
                    item.setNo_hp(list.getString("no_hp"));
                    item.setBiaya(list.getString("biaya"));
                    listItem.add(item);
                }
                adapter.notifyDataSetChanged();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    })

    @Override
    public void onFailure(Call<ResponseBody>
    call, Throwable t) {
        pDialog.dismiss();
        Toast.makeText(context, t.getMessage(),

```



```

Toast.LENGTH_SHORT).show();
    }
    }, 300);
}

public boolean onOptionsItemSelected(MenuItem item)
{
    if (item.getItemId() == android.R.id.home) {
        finish();
        return true;
    }

    return true;
}

private void displayLoader() {
    pDialog = new ProgressDialog(context);
    pDialog.setMessage("Sedang memuat daftar...");
    pDialog.setIndeterminate(false);
    pDialog.setCancelable(false);
    pDialog.show();
}
}

```

**Segmen Program 4.5** Detail Transaksi Pemasukan Perbulan dan Perhari  
e. form menu pengeluaran

Menu akan menampilkan data laporan pengeluaran yang nantinya *owner* yang bisa mengkonfirmasi data yang harus di tolak ataupun diterima oleh *owner*



**Gambar 4. 23**Menu *Transaction Out*

```
package com.androidigniter.loginandregistration;

import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;
import android.os.Bundle;
import android.os.Handler;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.DefaultItemAnimator;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.androidigniter.loginandregistration.Adapter.AdapterTransactionOut;
import com.androidigniter.loginandregistration.Api.BaseApiService;
import com.androidigniter.loginandregistration.Api.URL;
import com.androidigniter.loginandregistration.Model.ItemTransaksiKeluar;

import org.json.JSONArray;
import org.json.JSONObject;

import java.util.ArrayList;

import okhttp3.ResponseBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class TransactionOut extends AppCompatActivity {

    private RecyclerView lvti;
```

```

        ArrayList<ItemTransaksiKeluar> listItem = new
        ArrayList<>();

        Context context;
        AlertDialog aDialog;
        ProgressDialog pDialog;
        BaseApiService apiService;
        AdapterTransactionOut adapter;

        @Override
        protected void onCreate(Bundle
        savedInstanceState) {
            super.onCreate(savedInstanceState);

            setContentView(R.layout.activity_transaction_out);

            getSupportActionBar().setDisplayHomeAsUpEnabled(true);

            getSupportActionBar().setDisplayShowHomeEnabled(true);

            getSupportActionBar().setTitle("Transaction Out");

            apiService = URL.getAPIService();
            context = this;

            adapter = new
            AdapterTransactionOut(context, listItem);
            lvti = findViewById(R.id.lvti);
            lvti.setLayoutManager(new
            LinearLayoutManager(context));
            lvti.setItemAnimator(new
            DefaultItemAnimator());
            lvti.setAdapter(adapter);
        }

        @Override
        protected void onResume() {
            super.onResume();
            getData();
        }

        private void getData() {
            listItem.clear();
            displayLoader();

            new Handler().postDelayed(() ->

            apiService.statsTransaksi().enqueue(new
            Callback<ResponseBody>() {
                @Override

```

```

        public void
onResponse(Call<ResponseBody> call,
Response<ResponseBody> response) {
            pDialog.dismiss();
            try {
                JSONObject object = new
JSONObject(response.body().string());

                JSONArray array =
object.getJSONArray("result");
                for (int i = 0; i <
array.length(); i++) {
                    JSONObject list =
(JSONObject) array.get(i);

                    ItemTransaksiKeluar
item = new ItemTransaksiKeluar();
                    item.setId(list.getString("id"));
                    item.setStatus(list.getString("status"));
                    item.setTanggal(list.getString("tanggal"));
                    item.setTentang(list.getString("tentang"));
                    item.setKeterangan(list.getString("keterangan"));
                    item.setBiaya_pengeluaran(list.getString("biaya_pe
ngeluaran"));

                    listItem.add(item);
                }

                adapter.notifyDataSetChanged();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }

        @Override
        public void
onFailure(Call<ResponseBody> call, Throwable t) {
            pDialog.dismiss();
            Toast.makeText(context,
t.getMessage(), Toast.LENGTH_SHORT).show();
        }
    }

    , 300);
}

public void onConfirm(String id) {
    final AlertDialog.Builder builder = new

```

```

AlertDialog.Builder(context);
    final View layout =
LayoutInflater.from(context).inflate(R.layout.dial
og_confirm, null);
    builder.setView(layout);

    aDialog = builder.create();

aDialog.getWindow().setBackgroundDrawable(new
ColorDrawable(Color.TRANSPARENT));
    aDialog.show();

    Button btnKonfir =
layout.findViewById(R.id.btnKonfirmasi);
    Button btnTolak =
layout.findViewById(R.id.btnTolak);

    btnKonfir.setOnClickListener(v ->
proceedData("konfirmasi", id));
    btnTolak.setOnClickListener(v ->
proceedData("tolak", id));
    }

    public void proceedData(String action, String
id) {
        aDialog.dismiss();
        displayLoader();

        new Handler().postDelayed(() ->
apiService.prosesLaporan(action, id).enqueue(new
Callback<ResponseBody>() {
            @Override
            public void
onResponse(Call<ResponseBody> call,
Response<ResponseBody> response) {
                pDialog.dismiss();
                Toast.makeText(context, "Status
laporan telah di" + action.toLowerCase(),
Toast.LENGTH_SHORT).show();
                new Handler().postDelayed(() ->
getData(), 1200);
            }

            @Override
            public void
onFailure(Call<ResponseBody> call, Throwable t) {
                pDialog.dismiss();
                Toast.makeText(context,
t.getMessage(), Toast.LENGTH_SHORT).show();
            }
        })), 1800);
    }
}

```

```

        public boolean onCreateOptionsMenu(Menu menu)
        {
            // Inflate the menu; this adds items to
            the action bar if it is present.
            MenuInflater inflater = getMenuInflater();
            inflater.inflate(R.menu.optionmenu, menu);
            return true;
        }

        public boolean onOptionsItemSelected(MenuItem
        item) {
            if (item.getItemId() == R.id.t_in) {
                startActivity(new Intent(this,
                TransactionIn.class));
                finish();
            } else if (item.getItemId() == R.id.t_out)
            {
                startActivity(new Intent(this,
                TransactionOut.class));
                finish();
            } else if (item.getItemId() ==
            R.id.employee) {
                startActivity(new Intent(this,
                DataKaryawan.class));
                finish();
            } else if (item.getItemId() ==
            R.id.logout) {
                startActivity(new Intent(this,
                Login.class));
                finish();
            } else if (item.getItemId() ==
            android.R.id.home) {
                finish();
                return true;
            }
            return true;
        }

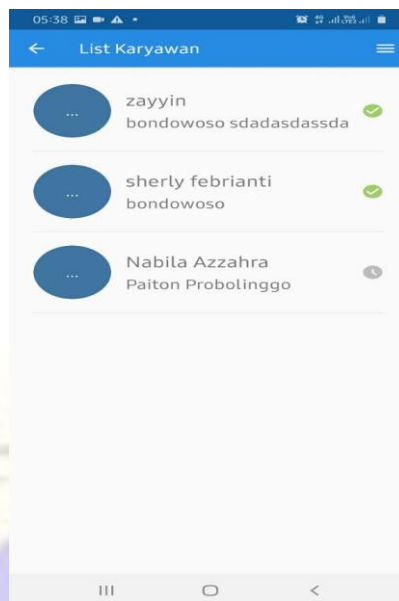
        private void displayLoader() {
            pDialog = new ProgressDialog(context);
            pDialog.setMessage("Sedang memuat
            daftar...");
            pDialog.setIndeterminate(false);
            pDialog.setCancelable(false);
            pDialog.show();
        }
    }
}

```

#### Segmen Program 4.6 Transaksi Pengeluaran

f. Form menu data karyawan

Menu ini akan menampilkan data karyawan.



**Gambar 4. 24** Menu Data Karyawan

```
package com.androidigniter.loginandregistration;

import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.DefaultItemAnimator;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.androidigniter.loginandregistration.Adapter.AdapterKaryawan;
import com.androidigniter.loginandregistration.Api.BaseApiService;
import com.androidigniter.loginandregistration.Api.URL;
import com.androidigniter.loginandregistration.Model.ItemKaryawan;

import org.json.JSONArray;
import org.json.JSONObject;

import java.util.ArrayList;
```

```

import okhttp3.ResponseBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class DataKaryawan extends AppCompatActivity {

    RecyclerView list_data_karyawan;

    Context context;
    ProgressDialog pDialog;
    BaseApiService apiService;
    AdapterKaryawan adapter;

    ArrayList<ItemKaryawan> listItem = new ArrayList<>();

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

        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        getSupportActionBar().setDisplayShowHomeEnabled(true);
        getSupportActionBar().setTitle("List Karyawan");

        apiService = URL.getAPIService();
        context = this;

        adapter = new AdapterKaryawan(context, listItem);
        list_data_karyawan =
        findViewById(R.id.list_data_karyawan);
        list_data_karyawan.setLayoutManager(new
        LinearLayoutManager(context));
        list_data_karyawan.setItemAnimator(new
        DefaultItemAnimator());
        list_data_karyawan.setAdapter(adapter);
    }

    @Override
    protected void onResume() {
        super.onResume();
        getData();
    }

    private void getData() {

        new Handler().postDelayed(() ->

        apiService.dataKaryawan().enqueue(new
        Callback<ResponseBody>() {
            @Override
            public void
            onResponse(Call<ResponseBody> call, Response<ResponseBody>
            response) {

```





```

        startActivity(new Intent(this,
TransactionIn.class));
        finish();
    } else if (item.getItemId() == R.id.t_out) {
        startActivity(new Intent(this,
TransactionOut.class));
        finish();
    } else if (item.getItemId() == R.id.employee) {
        startActivity(new Intent(this,
DataKaryawan.class));
        finish();
    } else if (item.getItemId() == R.id.logout) {
        startActivity(new Intent(this, Login.class));
        finish();
    } else if (item.getItemId() == android.R.id.home) {
        finish();
        return true;
    }
}

return true;
}

// Dialog
private void displayLoader() {
    pDialog = new ProgressDialog(context);
    pDialog.setMessage("Sedang memuat daftar...");
    pDialog.setIndeterminate(false);
    pDialog.setCancelable(false);
    pDialog.show();
}
}

```

#### **Segmen Program 4.7 Data Karyawan**

```

package com.androidigniter.loginandregistration;

import android.annotation.SuppressLint;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import com.androidigniter.loginandregistration.Api.BaseApiService;
import com.androidigniter.loginandregistration.Api.URL;

```

```

import okhttp3.ResponseBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class DataKaryawanDetail extends
AppCompatActivity {

    Context context;
    ProgressDialog pDialog;
    BaseApiService apiService;

    String id, nama, tanggal, nohp, alamat, status;
    TextView namax, tanggalx, nohpx, alamatx, statusx;
    LinearLayout wrap;

    @Override
    public void onCreate(@Nullable Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

setContentView(R.layout.activity_karyawan_detail);

getSupportActionBar().setDisplayHomeAsUpEnabled(true);

getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setTitle("Detail
Karyawan");

        apiService = URL.getAPIService();
        context = this;

        Intent detail = getIntent();
        id = detail.getStringExtra("id");
        nama = detail.getStringExtra("nama");
        tanggal = detail.getStringExtra("tanggal");
        nohp = detail.getStringExtra("nohp");
        alamat = detail.getStringExtra("alamat");
        status = detail.getStringExtra("status");

        namax = findViewById(R.id.nama);
        tanggalx = findViewById(R.id.tanggal);
        nohpx = findViewById(R.id.nohp);
        alamatx = findViewById(R.id.alamat);
        statusx = findViewById(R.id.status);

        namax.setText(nama);
        tanggalx.setText(tanggal);
        nohpx.setText(nohp);
        alamatx.setText(alamat);
        statusx.setText(status);

```

```

wrap = findViewById(R.id.wrap_button);

if (status.equals("Belum")) {
    wrap.setVisibility(View.VISIBLE);
} else {
    wrap.setVisibility(View.GONE);
}

Button btnYes =
findViewById(R.id.btnKonfirmasi);
Button btnNo = findViewById(R.id.btnTolak);

btnYes.setOnClickListener(v ->
onProcessApi("Terima", id));
btnNo.setOnClickListener(v ->
onProcessApi("Tolak", id));
}

public boolean onOptionsItemSelected(MenuItem item)
{
    if (item.getItemId() == android.R.id.home) {
        finish();
        return true;
    }

    return true;
}

@SuppressWarnings("SetTextI18n")
private void onProcessApi(String action, String id)
{
    displayLoader();

    new Handler().postDelayed(() ->
apiService.prosesKaryawan(action, id).enqueue(new
Callback<ResponseBody>() {
    @Override
    public void onResponse(Call<ResponseBody>
call, Response<ResponseBody> response) {
        pDialog.dismiss();
        statusx.setText(action.equals("Tolak") ?
"Ditolak" : "Konfirmasi");
        Toast.makeText(context, "Status karyawan
telah di" + action.toLowerCase(),
Toast.LENGTH_LONG).show();
        new Handler().postDelayed(() ->
finish(), 1800);
    }

    @Override
    public void

```

```

onFailure(Call<ResponseBody>call, Throwable t) {
            pDialog.dismiss();
            Toast.makeText(context, t.getMessage(),
Toast.LENGTH_SHORT).show();
        }
    }, 1000);
}

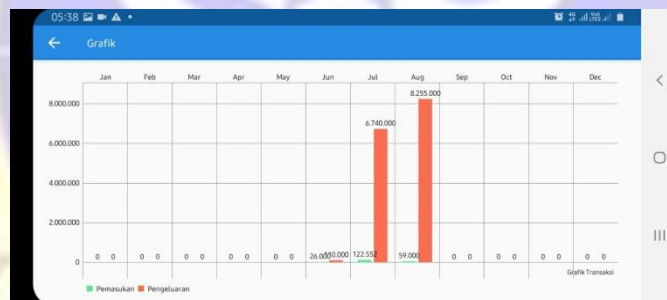
// Dialog
private void displayLoader() {
    pDialog = new ProgressDialog(context);
    pDialog.setMessage("Sedang memproses...");
    pDialog.setIndeterminate(false);
    pDialog.setCancelable(false);
    pDialog.show();
}
}
}

```

#### Segmen Program 4.8 DetailData Karyawan

##### g. Form Menu Grafik

Menu ini akan menampilkan grafik hasil pemasukan dan pengeluaran perbulannya.



**Gambar 4. 25** Menu Grafik

```

package com.androidigniter.loginandregistration;

import android.app.ProgressDialog;
import android.content.Context;
import android.graphics.Color;
import android.os.Bundle;
import android.os.Handler;
import android.util.Log;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.androidigniter.loginandregistration.Api.BaseApiService;

```

```

import com.androidigniter.loginandregistration.Api.URL;
import
com.androidigniter.loginandregistration.Model.ItemGrafik
;
import com.github.mikephil.charting.charts.BarChart;
import com.github.mikephil.charting.data.BarData;
import com.github.mikephil.charting.data.BarDataSet;
import com.github.mikephil.charting.data.BarEntry;
import
com.github.mikephil.charting.interfaces.datasets.IBarDat
aSet;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class Chart extends AppCompatActivity {

    Context context;
    ProgressDialog pDialog;
    BaseApiService apiService;

    BarChart chart;

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

        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setTitle("Grafik");

        apiService = URL.getAPIService();
        context = this;

        chart = (BarChart) findViewById(R.id.chart);
    }

    @Override
    protected void onResume() {
        super.onResume();
        getData();
    }

    @Override
    public boolean onSupportNavigateUp() {
        onBackPressed();
        return true;
    }
}

```

```

    }

    // Api
    private void getData() {
        displayLoader();

        new Handler().postDelayed(() ->
        apiService.getGrafikTransaksi().enqueue(new
        Callback<ItemGrafik>() {
            @Override
            public void onResponse(Call<ItemGrafik>
            call, Response<ItemGrafik> response) {
                pDialog.dismiss();
                setData(response.body());
            }

            @Override
            public void onFailure(Call<ItemGrafik> call,
            Throwable t) {
                pDialog.dismiss();
                Toast.makeText(context, t.getMessage(),
                Toast.LENGTH_SHORT).show();
            }
        })), 1000);

    // Chart
    private void setData(ItemGrafik body) {
        ArrayList<IBarDataSet> dataSets = null;

        // Pemasukan
        ArrayList<BarEntry> pemasukan = new
        ArrayList<>();
        for (int i = 0; i <
        body.getBarPemasukan().size(); i++) {
            BarEntry value = new
            BarEntry(body.getBarPemasukan().get(i), i); // Jan
            pemasukan.add(value);
        }
        BarDataSet pemasukanData = new
        BarDataSet(pemasukan, "Pemasukan");

        pemasukanData.setColor(Color.parseColor("#5AE991"));

        // Pengeluaran
        ArrayList<BarEntry> pengeluaran = new
        ArrayList<>();
        for (int i = 0; i <
        body.getBarPengeluaran().size(); i++) {
            BarEntry value = new
            BarEntry(body.getBarPengeluaran().get(i), i); // Jan
            pengeluaran.add(value);
        }
    }
}

```

```

    }
    BarDataSet pengeluaranData = new
BarDataSet(pengeluaran, "Pengeluaran");

pengeluaranData.setColor(Color.parseColor("#FC6E52"));

    dataSets = new ArrayList<>();
    dataSets.add(pemasukanData);
    dataSets.add(pengeluaranData);

    // Bulan on TOP
    ArrayList<String> xAxis = new ArrayList<>();
    for (String months : body.getBarMonths()) {
        Log.d("CHART_RESPONSE", "month: " + months);
        xAxis.add(months);
    }

    BarData data = new BarData(xAxis, dataSets);

    chart.setData(data);
    chart.setDescription("Grafik Transaksi");
    chart.getAxisRight().setEnabled(false);
    chart.animateXY(2000, 2000);
    chart.invalidate();
}

// Add-ons
private void displayLoader() {
    pDialog = new ProgressDialog(context);
    pDialog.setMessage("Sedang memuat daftar...");
    pDialog.setIndeterminate(false);
    pDialog.setCancelable(false);
    pDialog.show();
}
}

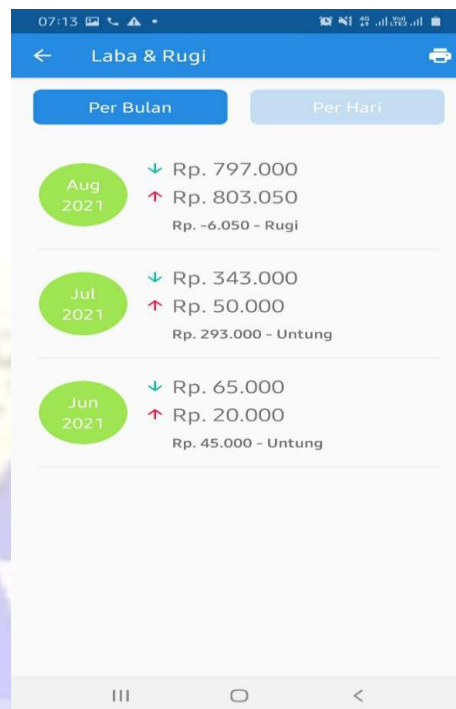
```

#### Segmen Program 4.9 Grafik Pendapatan Selama Satu Tahun



h. Form Menu Laporan Laba Rugi

Menu ini menampilkan laporan laba rugi pemasukan perbulannya dan perharinya laporan ini juga bisa *diprint out*.



**Gambar 4. 26** Menu Laporan

```
package
com.androidigniter.loginandregistration;

import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.os.Handler;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import
androidx.appcompat.app.AppCompatActivity;
import androidx.cardview.widget.CardView;
```

```
import androidx.core.content.ContextCompat;
import
androidx.recyclerview.widget.DefaultItemAnimato
r;
import
androidx.recyclerview.widget.LinearLayoutManage
r;
import
androidx.recyclerview.widget.RecyclerView;

import
com.androidigniter.loginandregistration.Adapter
.AdapterLabaRugi;
import
com.androidigniter.loginandregistration.Api.Bas
eApiService;
import
com.androidigniter.loginandregistration.Api.URL
;
import
com.androidigniter.loginandregistration.Model.I
temLabaRugi;

import org.json.JSONArray;
import org.json.JSONObject;

import java.util.ArrayList;

import okhttp3.ResponseBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class Laba extends AppCompatActivity {

    RecyclerView lvhape;
    CardView btnBulan, btnHari;

    ArrayList<ItemLabaRugi> listItem = new
ArrayList<>();

    Context context;
    ProgressDialog pDialog;
    BaseApiService apiService;
    AdapterLabaRugi adapter;

    String from = "bulan";

    @Override
    protected void onCreate(@Nullable Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```

        setContentView(R.layout.activity_Laba);

        getSupportActionBar().setDisplayHomeAsUpEnabled(
            true);

        getSupportActionBar().setDisplayHomeAsUpEnabled(
            true);
        getSupportActionBar().setTitle("Laba &
        Rugi");

        apiService = URL.getAPIService();
        context = this;

        adapter = new AdapterLabaRugi(context,
        listItem);
        lvhape = findViewById(R.id.lvhape);
        lvhape.setLayoutManager(new
        LinearLayoutManager(context));
        lvhape.setItemAnimator(new
        DefaultItemAnimator());
        lvhape.setAdapter(adapter);

        btnBulan =
        findViewById(R.id.btn_bulan);
        btnHari = findViewById(R.id.btn_hari);

        btnBulan.setOnClickListener(v ->
        getData("bulan"));
        btnHari.setOnClickListener(v ->
        getData("hari"));
    }

    @Override
    protected void onResume() {
        super.onResume();
        getData("bulan");
    }

    private void getData(String metode) {
        listItem.clear();
        displayLoader();

        if (metode.equals("hari")) {
            from = "hari";

            btnBulan.setCardBackgroundColor(ContextCompat.g
            etColor(context, R.color.colorPrimaryOut));

            btnHari.setCardBackgroundColor(ContextCompat.ge
            tColor(context, R.color.colorPrimary));
        } else {
            from = "bulan";
        }
    }

```

```
btnBulan.setCardBackgroundColor(ContextCompat.g  
etColor(context, R.color.colorPrimary));
```

```
btnHari.setCardBackgroundColor(ContextCompat.g  
etColor(context, R.color.colorPrimaryOut));  
}
```

```
new Handler().postDelayed(() ->  
apiService.getLaporanLabaRugi(metode).enqueue(n  
ew Callback<ResponseBody>() {
```

```
    @Override  
    public void  
onResponse(Call<ResponseBody> call,  
Response<ResponseBody> response) {  
    progressDialog.dismiss();
```

```
    try {  
        JSONObject object = new  
JSONObject(response.body().string());
```

```
        JSONArray array =  
object.getJSONArray("data");  
        for (int i = 0; i <  
array.length(); i++) {
```

```
            JSONObject list =  
(JSONObject) array.get(i);  
  
            ItemLabaRugi item = new  
ItemLabaRugi();
```

```
            item.setTanggal(list.getString("tanggal");  
            item.setPemasukan(list.getString("pemasukan");  
            item.setPengeluaran(list.getString("pengeluaran  
"));
```

```
            item.setHasil(list.getString("hasil");  
            item.setStatus(list.getString("status");  
            listItem.add(item);  
        }  
    }
```

```
adapter.notifyDataSetChanged();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

```
    @Override  
    public void  
onFailure(Call<ResponseBody> call, Throwable t)
```

```

        {
            pDialog.dismiss();
            Toast.makeText(context,
t.getMessage(), Toast.LENGTH_SHORT).show();
        }
    }, 300);
}

@Override
public boolean onCreateOptionsMenu(Menu
menu) {
    MenuInflater inflater =
getMenuInflater();
    inflater.inflate(R.menu.optionpdf,
menu);
    return true;
}

@Override
public boolean
onOptionsItemSelected(@NonNull MenuItem item) {
    if (item.getItemId() == R.id.menu) {
        displayLoader();

        Uri uri = Uri.parse(URL.BASE_URL +
"laporan_print.php?from=" + from);
        Intent intent = new
Intent(Intent.ACTION_VIEW, uri);
        startActivity(intent);

        new Handler().postDelayed(() ->
pDialog.dismiss(), 2000);
    } else if (item.getItemId() ==
android.R.id.home) {
        finish();
        return true;
    }
}

return true;
}

private void displayLoader() {
    pDialog = new ProgressDialog(context);
    pDialog.setMessage("Sedang memuat
daftar...");
    pDialog.setIndeterminate(false);
    pDialog.setCancelable(false);
    pDialog.show();
}
}
}

```

**Segmen Program 4.10**Laporan Laba Rugi Perbulan dan Perhari

#### 1.4 Pengujian sistem

Pengujian yang digunakan yaitu metode black-box, bertujuan untuk mengetahui data yang dihasilkan sesuai dengan yang diharapkan oleh pengguna sistem.

##### a. Pengujian Internal

**Tabel 4. 8** Tabel Pengujian Internal

No	Form yang diuji	Fungsi	Hasil yang diharapkan	Hasil	
				sesuai	tidak
1	Form Login	Username dan Password benar kemudian klik Tombol Login	Menampilkan Halaman Utama	✓	
2	Form Login	Username dan Password salah	Menampilkan pesan	✓	
3	Halaman Menu	untuk tampilan menu	Menampilkan halaman menu	✓	
4	Data List Karyawan	Bila ingin mengkonfirmasi data karyawan	Untuk melihat data – data karyawan	✓	
5	Menu pemasukan	Untuk melihat pemasukan perbulannya	Sistem akan menampilkan laporan pemasukan.	✓	
6	Menu Pengeluaran	Untuk melihat pengeluaran	Sistem akan menampilkan laporan pengeluaran yang harus di konfirmasi	✓	
7	Menu data Grafik	Pilih menu grafik	Menampilkan grafik data pemasukan dan pengeluaran perbulannya	✓	

### b. Pengujian Eksternal dengan Angket

Pengujian eksternal diujikan kepada 3responden sebagai user dalam aplikasi ini, dibawah ini adalah beberapa pertanyaan yang diajukan kepada user untuk pengujian kelayakan dari aplikasi ini.

**Tabel 4. 9** Tabel Pengujian Eksternal dengan Angket

No	Pertanyaan	Sangat baik	Baik	Cukup	Kurang	Sangat kurang
1	Apakah aplikasi ini mudah untuk dioperasikan?	2	1			
2	Apakah dengan adanya aplikasi ini membantu owner dalam memonitoring wartel dalam pelaporan keuangan?	1	1	1		
3	Apakah sistem monitoring ini membantu koordinator dan petugas wartel?	3				
4	Apakah desain dari sistem ini sudah menarik dan sesuai dengan yang diharapkan?	1	1	1		
5	Apakah sistem ini layak untuk digunakan?	2		1		
TOTAL		9	3	3	0	0

#### Perhitungan Rumus Skala Likert :

Jumlah sampel : 3 orang

Jumlah pertanyaan : 5

Responden yang menjawab sangat baik(skor 5) berjumlah 9

Responden yang menjawab baik(skor 4) berjumlah 3

Responden yang menjawab cukup (skor 3) berjumlah 3

Responden yang menjawab kurang (skor 2) berjumlah 0

Responden yang menjawab sangat kurang (skor 1) berjumlah 0

### Rumus $T \times P_n$

Responden yang menjawab sangat baik (skor 5) berjumlah  $9 \times 5 = 45$

Responden yang menjawab baik (skor 4) berjumlah  $3 \times 4 = 12$

Responden yang menjawab cukup (skor 3) berjumlah  $3 \times 3 = 9$

Responden yang menjawab kurang (skor 2) berjumlah  $0 \times 2 = 0$

Responden yang menjawab sangat kurang (skor 1) berjumlah  $0 \times 1 = 0$

Semua hasil dijumlahkan, total skor = 66

Interpretasi Skor Perhitungan

$Y = \text{Skor tertinggi likert} \times \text{Jumlah soal} \times \text{Jumlah responden}$

$$5 \times 5 \times 3 = 75$$

$X = \text{Skor terendah likert} \times \text{Jumlah soal} \times \text{Jumlah responden}$

$$1 \times 5 \times 3 = 15$$

Rumus interval

$$I = 100/5$$

$I = 20$  (Intervalnya jarak dari terendah 0% hingga tertinggi 100%)

**Tabel 4. 10** Rumus Interval

No	Kategori	Presentase
1	Angka 0% - 19,99%	Sangat kurang
2	Angka 20% - 39,99%	Kurang
3	Angka 40% - 59,99%	Cukup
4	Angka 60% - 79,99%	Baik
5	Angka 80% - 100%	Sangat Baik

Dari 5 pertanyaan yang telah diisi oleh 3 responden sebagai user, diperoleh nilai akhir (sangat baik = 9), (baik 3), (cukup 3).



**Tabel 4. 11** Tabel Hasil dari rumus T x Pn

Skala jawaban	Frekuensi	Hasil
Sangat baik	9 x 5	45
Baik	3 x 4	12
Cukup	3 x 3	9
Kurang	0 x 2	0
Sangat Kurang	0 x 1	0
Total		66

Jadi total perhitungan skor didapat sebanyak 66

Penyelesaian Akhir

$$\text{Rumus index} \times 100\% = \text{Total Skor} / Y \times 100\%$$

$$= 66 / 75 \times 100\% = 88\%$$

Jadi dari hasil tersebut menunjukkan 88% responden dengan kategori yang sering dipilih yaitu sangat baik “perancangan sistem monitoring wartel wilayah azzainiyah berbasis android”. Dapat disimpulkan bahwa aplikasi yang dibuat dapat memonitoring wartel azzainiyah.