Script menu activity

```
package com.dilla.a220280030.woparepare.activity;


import android.annotation.SuppressLint;

import android.content.Intent;

import android.graphics.Color;

import android.os.Bundle;

import android.os.Handler;

import android.view.Gravity;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.RelativeLayout;

import android.widget.TextView;

import android.widget.Toast;


import androidx.annotation.NonNull;

import androidx.appcompat.app.ActionBar;

import androidx.appcompat.app.AppCompatActivity;

import androidx.appcompat.widget.Toolbar;

import
androidx.recyclerview.widget.GridLayoutManager;

import
androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import
androidx.swiperefreshlayout.widget.SwipeRefreshLayo
ut;


import
com.dilla.a220280030.woparepare.BackToLogin;

import com.dilla.a220280030.woparepare.R;

import
com.dilla.a220280030.woparepare.SessionManager;

import
com.dilla.a220280030.woparepare.adapter.MenuAdapte
r;

import
com.dilla.a220280030.woparepare.api.ApiClient;

import
com.dilla.a220280030.woparepare.api.ApiInterface;

import
com.dilla.a220280030.woparepare.model.ResultData;


import java.util.List;


import retrofit2.Call;

import retrofit2.Callback;

import retrofit2.Response;


public class MenuActivity extends AppCompatActivity
{


    private RecyclerView recyclerView;

    private SwipeRefreshLayout mSwipeRefreshLayout;

    private RelativeLayout errorlayout;

    private ImageView imgerror;

    private TextView errortitle, errormessage,
textCartItemCount;

    private Button btnretry;

    MenuAdapter adapter;

    MenuAdapter.ItemClickListener itemClickListener;

    List<ResultData> resultDataList;

    String email, nohp, nama, alamat;

    SessionManager sessionManager;

    ApiInterface apiInterface;


    @SuppressLint({"CutPasteId", "SetTextI18n"})

    @Override

    protected void onCreate(Bundle savedInstanceState)
{

        super.onCreate(savedInstanceState);
```

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



        sessionManager = new
SessionManager(MenuActivity.this);

        if(!sessionManager.isLoggedIn()){


BackToLogin.kembaliKeHalamanMasuk(MenuActivity
.this);

        }

        email =
sessionManager.getUserDetail().get(SessionManager.E
MAIL);

        nohp =
sessionManager.getUserDetail().get(SessionManager.N
OHP);

        nama =
sessionManager.getUserDetail().get(SessionManager.N
AMA);

        alamat =
sessionManager.getUserDetail().get(SessionManager.A
LAMAT);



        errorlayout = findViewById(R.id.rel_error);

        imgerror = findViewById(R.id.imgerror);

        errortitle = findViewById(R.id.tverror);

        errormessage = findViewById(R.id.tvsuberror);

        btnretry = findViewById(R.id.btnretry);



        mSwipeRefreshLayout =
findViewById(R.id.swaplistmenu);

        recyclerView = findViewById(R.id.listviewmenu);

        recyclerView.setLayoutManager(new
LinearLayoutManager(this));

        recyclerView.setHasFixedSize(true);

        recyclerView.setLayoutManager(new
GridLayoutManager(this, 2));



        ambilmenu();



mSwipeRefreshLayout.setColorSchemeResources(R.co
lor.colorPrimary);

mSwipeRefreshLayout.setOnRefreshListener(this::ambi
lmenu);



        itemClickListener = ((view, position) -> {

            String idkate =
resultDataList.get(position).getIdKategori();

            String nmkate =
resultDataList.get(position).getNamaKategori();

            Intent it = new Intent(MenuActivity.this,
MenuDaftarActivity.class);

            it.putExtra("idkate", idkate);

            it.putExtra("nmkate", nmkate);

            startActivity(it);

        });



        Toolbar toolbar = findViewById(R.id.toolbar);

        setSupportActionBar(toolbar);

        ActionBar mAction = getSupportActionBar();

        assert mAction != null;

        TextView titleTextView = new
TextView(getApplicationContext());

        titleTextView.setText("Daftar
Menu".toUpperCase());

        titleTextView.setTextSize(17);

        titleTextView.setTextColor(Color.WHITE);

        titleTextView.setGravity(Gravity.CENTER);


mAction.setDisplayOptions(ActionBar.DISPLAY_SH
OW_CUSTOM);

        mAction.setCustomView(titleTextView, new
ActionBar.LayoutParams(ActionBar.LayoutParams.W
RAP_CONTENT,
ActionBar.LayoutParams.WRAP_CONTENT,
Gravity.CENTER));

    }


    @Override

    public boolean onCreateOptionsMenu(Menu menu) {

        getMenuInflater().inflate(R.menu.data_diri,
menu);
```

```
        final MenuItem menuItem = menu.findItem(R.id.keranjang);

        View actionView = menuItem.getActionView();

        textCartItemCount = actionView.findViewById(R.id.cart_badge);

        badgekeranjang(email);

        Handler handler = new Handler();

        handler.postDelayed(new Runnable() {

            @Override

            public void run() {

                badgekeranjang(email);

                handler.postDelayed(this, 1000);

            }

        }, 1000);

        actionView.setOnClickListener(v -> onOptionsItemSelected(menuItem));

        MenuItem listmenu = menu.findItem(R.id.listmenu);

        listmenu.setVisible(false);

        return true;

    }



    @SuppressLint("SetTextI18n")

    @Override

    public boolean onOptionsItemSelected(MenuItem item) {

        if (item.getItemId() == R.id.data_diri) {

            Intent it = new Intent(MenuActivity.this, ProfilActivity.class);

            startActivity(it);

        } else if (item.getItemId() == R.id.keranjang) {

            startActivity(new Intent(MenuActivity.this, KeranjangActivity.class));

        }

        return super.onOptionsItemSelected(item);

    }
```

```
    private void ambilmenu(){

        errorlayout.setVisibility(View.GONE);

        showLoading();

        apiInterface = ApiClient.getApiClient().create(ApiInterface.class);

        Call<List<ResultData>> call = apiInterface.getMenu();

        call.enqueue(new Callback<List<ResultData>>() {



            @Override

            public void onResponse(@NonNull Call<List<ResultData>> call, @NonNull Response<List<ResultData>> response) {

                hideLoading();

                if(response.isSuccessful()){

                    if (response.body() != null) {

                        List<ResultData> resultDataList = response.body();

                        if (resultDataList != null && !resultDataList.isEmpty()) {

                            // Respons berhasil, dan data tidak kosong

                            onGetResult(resultDataList);

                        } else {

                            // Respons berhasil, tetapi data kosong


showErrorMessage(R.drawable.no_result, "No Result", "No menu available.");

                        }

                    } else {

                        // Respons berhasil, tetapi body kosong


showErrorMessage(R.drawable.no_result, "No Result", "No menu available.");

                    }

//                  onGetResult(response.body());

                }else{

                    String errorCode;

                    switch (response.code()){
```

```
                case 404:
                    errorCode = "404 not found";
                    break;
                case 500:
                    errorCode = "500 server broken";
                    break;
                default:
                    errorCode = "Unknown error";
                    break;
            }
            showErrorMessage(R.drawable.no_result,
"No Result", "Please Try Again!\n"+ errorCode);
        }
    }

    @Override
    public void onFailure(@NonNull
Call<List<ResultData>> call, @NonNull Throwable t)
{
        hideLoading();
        onErrorLoading(t.toString());
    }
  });
 }

 private void badgekeranjang(String email){
  apiInterface =
ApiClient.getApiClient().create(ApiInterface.class);
  Call<List<ResultData>> call =
apiInterface.getCountKeranjang(email);
  call.enqueue(new Callback<List<ResultData>>() {

    @Override
    public void onResponse(@NonNull
Call<List<ResultData>> call, @NonNull
Response<List<ResultData>> response) {
        if(response.isSuccessful() &&
response.body() != null){
            int jml =
response.body().get(0).getIsiKeranjang();
            setupBadge(jml);
        }else{
            showToast("Gagal meload data",
R.layout.toast_gagal);
        }
    }

    @Override
    public void onFailure(@NonNull
Call<List<ResultData>> call, @NonNull Throwable t)
{
        showToast(t.toString(), R.layout.toast_gagal);
    }
  });
 }

 private void showToast(String message, int layoutId)
{
  final Toast toast = new
Toast(getApplicationContext());
  toast.setDuration(Toast.LENGTH_LONG);
  @SuppressLint("InflateParams") View
custom_view = getLayoutInflater().inflate(layoutId,
null);
  TextView tvMessage =
custom_view.findViewById(R.id.message);
  tvMessage.setText(message);
  toast.setView(custom_view);
  toast.setGravity(Gravity.CENTER, 0, 0);
  toast.show();
 }

 private void setupBadge(Integer mCartItemCount) {
  if (textCartItemCount != null) {
    if (mCartItemCount == 0) {
      if (textCartItemCount.getVisibility() !=
View.GONE) {
```

```
textCartItemCount.setVisibility(View.GONE);
            }
        } else {
            int maxLength = 2;
            String itemCountText = String.valueOf(Math.min(mCartItemCount, 99));
            if (itemCountText.length() > maxLength) {
                itemCountText = itemCountText.substring(0, maxLength);
            }
            if (mCartItemCount > 99) {
                itemCountText += "+";
            }
            textCartItemCount.setText(itemCountText);
            textCartItemCount.setPadding(0, 0, 0, 0);
            if (textCartItemCount.getVisibility() != View.VISIBLE) {
textCartItemCount.setVisibility(View.VISIBLE);
            }
        }
    }
  }

  private void showLoading() {
    mSwipeRefreshLayout.setRefreshing(true);
  }

  private void hideLoading() {
    mSwipeRefreshLayout.setRefreshing(false);
  }

  @SuppressLint("NotifyDataSetChanged")
  private void onGetResult(List<ResultData> values) {
    adapter = new MenuAdapter(this, values, itemClickListener);
    adapter.notifyDataSetChanged();
    recyclerView.setAdapter(adapter);
    resultDataList = values;
  }

  private void onErrorLoading(String message) {
    showErrorMessage(R.drawable.no_result, "Oops...", "Network failure, Please Try Again!.\n"+ message);
  }

  private void showErrorMessage(int imageView, String title, String message){

    if(errorlayout.getVisibility() == View.GONE){
      errorlayout.setVisibility(View.VISIBLE);
    }

    imgerror.setImageResource(imageView);
    errortitle.setText(title);
    errormessage.setText(message);

    btnretry.setOnClickListener(view -> ambilmenu());
  }
}
```

Script daftar activity

```
package com.dilla.a220280030.woparepare.activity;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.graphics.Paint;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
```

```
import android.text.method.HideReturnsTransformationMethod;

import android.text.method.PasswordTransformationMethod;

import android.view.Gravity;

import android.view.View;

import android.widget.ImageView;

import android.widget.RelativeLayout;

import android.widget.TextView;

import android.widget.Toast;


import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;


import com.dilla.a220280030.woparepare.R;

import com.dilla.a220280030.woparepare.api.ApiClient;

import com.dilla.a220280030.woparepare.api.ApiInterface;

import com.dilla.a220280030.woparepare.model.Data;

import com.dilla.a220280030.woparepare.model.ResultData;

import com.google.android.material.textfield.TextInputEditText;

import com.google.android.material.textfield.TextInputLayout;


import java.util.Objects;


import retrofit2.Call;

import retrofit2.Callback;

import retrofit2.Response;


public class DaftarActivity extends AppCompatActivity implements View.OnClickListener {

    private TextInputLayout textemail, textnohp, textnama, textpass, textulangpass;

    private TextInputEditText edtemail, edtnohp, edtnama, edtpass, edtulangpass;

    private ImageView imgpass, imgulangpass;

    private RelativeLayout btndaftar;

    String email, nohp, nama, pass, ulangpass;

    private TextView tvlogin;

    ApiInterface apiInterface;


    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_daftar);


        textemail = findViewById(R.id.textemail);

        textnohp = findViewById(R.id.textnohp);

        textnama = findViewById(R.id.textnama);

        textpass = findViewById(R.id.textpass);

        textulangpass = findViewById(R.id.textulangpass);

        edtemail = findViewById(R.id.edtemail);

        edtemail.addTextChangedListener(new TextWatcher() {

            @Override

            public void beforeTextChanged(CharSequence s, int start, int count, int after) {


            }


            @Override

            public void onTextChanged(CharSequence s, int start, int before, int count) {

                validemail();

            }
```

```
        @Override
        public void afterTextChanged(Editable s) {

        }
    });
    edtnohp = findViewById(R.id.edtnohp);
    edtnohp.addTextChangedListener(new
TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence
s, int start, int count, int after) {

        }

        @Override
        public void onTextChanged(CharSequence s, int
start, int before, int count) {
            validnohp();
        }

        @Override
        public void afterTextChanged(Editable s) {

        }
    });
    edtnama = findViewById(R.id.edtnama);
    edtnama.addTextChangedListener(new
TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence
s, int start, int count, int after) {

        }

        @Override
        public void onTextChanged(CharSequence s, int
start, int before, int count) {
            validnama();
        }

        @Override
        public void afterTextChanged(Editable s) {

        }
    });
    edtpass = findViewById(R.id.edtpass);
    edtpass.addTextChangedListener(new
TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence
s, int start, int count, int after) {

        }

        @Override
        public void onTextChanged(CharSequence s, int
start, int before, int count) {
            validpass();
        }

        @Override
        public void afterTextChanged(Editable s) {

        }
    });
    edtulangpass = findViewById(R.id.edtulangpass);
    edtulangpass.addTextChangedListener(new
TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence
s, int start, int count, int after) {

        }
```

```
        @Override
        public void onTextChanged(CharSequence s, int start, int before, int count) {
            validulangpass();
        }

        @Override
        public void afterTextChanged(Editable s) {

        }
    });

    imgpass = findViewById(R.id.show_pass_btn);
    imgpass.setOnClickListener(v -> {
        if (edtpass.getTransformationMethod().equals(PasswordTransformationMethod.getInstance())) {
            imgpass.setImageResource(R.drawable.hide_password);
            edtpass.setTransformationMethod(HideReturnsTransformationMethod.getInstance());
        } else {
            imgpass.setImageResource(R.drawable.show_password);
            edtpass.setTransformationMethod(PasswordTransformationMethod.getInstance());
        }
    });

    imgulangpass = findViewById(R.id.show_ulangpass_btn);
    imgulangpass.setOnClickListener(v -> {
        if (edtulangpass.getTransformationMethod().equals(PasswordTransformationMethod.getInstance())) {
            imgulangpass.setImageResource(R.drawable.hide_password);
            edtulangpass.setTransformationMethod(HideReturnsTransformationMethod.getInstance());
        } else {
            imgulangpass.setImageResource(R.drawable.show_password);
            edtulangpass.setTransformationMethod(PasswordTransformationMethod.getInstance());
        }
    });

    btndaftar = findViewById(R.id.btndaftar);
    btndaftar.setOnClickListener(this);
    tvlogin = findViewById(R.id.masuk);
    tvlogin.setPaintFlags(Paint.UNDERLINE_TEXT_FLAG);
    tvlogin.setOnClickListener(this);
  }

  private void daftar(String email, String nohp, String nama, String pass, String ulangpass) {
    apiInterface = ApiClient.getApiClient().create(ApiInterface.class);
    Call<Data> daftarCall = apiInterface.daftarRespone(email, nohp, nama, pass, ulangpass);
    daftarCall.enqueue(new Callback<Data>() {
        @Override
        public void onResponse(@NonNull Call<Data> call, @NonNull Response<Data> response) {
            if (response.isSuccessful() && response.body() != null) {
                Data responseData = response.body();
                if (responseData.isStatus()) {
                    ResultData resultData = responseData.getResultData();
```

```
                if (resultData != null) {

                    String message = responseData.getMessage();

                    if ("Akun Anda belum aktif, Silahkan cek Email untuk melakukan Verifikasi Akun.".equals(message)) {

                        showToast(message, R.layout.toast_info);

                    } else {

                        showToast(message, R.layout.toast_berhasil);

                    }

                    Intent intent = new Intent(DaftarActivity.this, OTPNumberActivity.class);

                    intent.putExtra("EmailNoHP", email);

                    intent.putExtra("KodeOTP", resultData.getOTPNumber());

                    startActivity(intent);

                    finish();

                }

            } else {

                showToast(responseData.getMessage(), R.layout.toast_gagal);

            }

        } else {

            showToast("Terjadi kesalahan, silakan coba lagi.", R.layout.toast_gagal);

        }

    }



    @SuppressLint("SetTextI18n")

    @Override

    public void onFailure(@NonNull Call<Data> call, @NonNull Throwable t) {

        showToast("Tidak ada koneksi internet, silahkan coba lagi!", R.layout.toast_gagal);

    }

  });

 }
```

```
 @Override

 public void onClick(View v) {

  if(v.getId() == R.id.btndaftar) {

   try {

    if (!validemail() | !validnohp() | !validnama() | !validpass() | !validulangpass()) {

     return;

    }

    email = Objects.requireNonNull(edtemail.getText()).toString();

    nohp = Objects.requireNonNull(edtnohp.getText()).toString();

    nama = Objects.requireNonNull(edtnama.getText()).toString();

    pass = Objects.requireNonNull(edtpass.getText()).toString();

    ulangpass = Objects.requireNonNull(edtulangpass.getText()).toString();

    daftar(email, nohp, nama, pass, ulangpass);

   } catch (Exception e) {

    e.printStackTrace();

   }

  }else if(v.getId() == R.id.masuk){

   startActivity(new Intent(DaftarActivity.this, MasukActivity.class));

   finish();

  }

 }



 private void showToast(String message, int layoutId) {

  final Toast toast = new Toast(getApplicationContext());

  toast.setDuration(Toast.LENGTH_LONG);

  @SuppressLint("InflateParams") View custom_view = getLayoutInflater().inflate(layoutId, null);
```

```
        TextView tvMessage = custom_view.findViewById(R.id.message);
        tvMessage.setText(message);
        toast.setView(custom_view);
        toast.setGravity(Gravity.CENTER, 0, 0);
        toast.show();
    }

    private boolean validemail() {
        if (Objects.requireNonNull(edtemail.getText()).toString().trim().isEmpty()) {
            textemail.setError("Masukkan Email");
            return false;
        } else {
            textemail.setError(null);
            return true;
        }
    }
    private boolean validnohp() {
        if (Objects.requireNonNull(edtnohp.getText()).toString().trim().isEmpty()) {
            textnohp.setError("Masukkan No. HP");
            return false;
        } else {
            textnohp.setError(null);
            return true;
        }
    }
    private boolean validnama() {
        if (Objects.requireNonNull(edtnama.getText()).toString().trim().isEmpty()) {
            textnama.setError("Masukkan Nama");
            return false;
        } else {
            textnama.setError(null);
            return true;
        }
    }
    private boolean validpass() {
        if (Objects.requireNonNull(edtpass.getText()).toString().trim().isEmpty()) {
            textpass.setError("Masukkan Password");
            return false;
        } else {
            textpass.setError(null);
            return true;
        }
    }
    private boolean validulangpass() {
        if (Objects.requireNonNull(edtulangpass.getText()).toString().trim().isEmpty()) {
            textulangpass.setError("Masukkan Konfirmasi Password");
            return false;
        } else {
            textulangpass.setError(null);
            return true;
        }
    }
}
```

Script OTP number activity

```
package com.dilla.a220280030.woparepare.activity;

import android.app.ProgressDialog;
import android.content.Intent;
import android.graphics.Color;
import android.os.Bundle;
import android.os.CountDownTimer;
```

```
import android.view.Gravity;

import android.view.View;

import android.widget.TextView;

import android.widget.Toast;


import androidx.appcompat.app.ActionBar;

import androidx.appcompat.app.AppCompatActivity;

import androidx.appcompat.widget.Toolbar;


import com.dilla.a220280030.woparepare.R;

import
com.dilla.a220280030.woparepare.api.ApiClient;

import
com.dilla.a220280030.woparepare.api.ApiInterface;

import com.dilla.a220280030.woparepare.model.Data;

import
com.dilla.a220280030.woparepare.model.ResultData;


import org.jetbrains.annotations.NotNull;


import in.aabhasjindal.otptextview.OTPListener;

import in.aabhasjindal.otptextview.OtpTextView;

import retrofit2.Call;

import retrofit2.Callback;

import retrofit2.Response;


public class OTPNumberActivity extends
AppCompatActivity {


   private OtpTextView otpTextView;

   private ProgressDialog progress;

   private TextView timerTextView;

   private CountDownTimer countDownTimer;

   private boolean isTimerRunning = true;

   String emailnohp, kodeotp;

   ApiInterface apiInterface;


   @Override

   protected void onCreate(Bundle savedInstanceState)
{

      super.onCreate(savedInstanceState);

      setContentView(R.layout.activity_otpnumber);


      timerTextView                                =
findViewById(R.id.timer_text_view);


      startCountdownTimer();


      Intent geteml = getIntent();

      emailnohp                                    =
geteml.getStringExtra("EmailNoHP");

      kodeotp = geteml.getStringExtra("KodeOTP");


      progress                =                new
ProgressDialog(OTPNumberActivity.this);

      progress.setCancelable(false);

      progress.setMessage("Mohon Tunggu...");


      otpTextView = findViewById(R.id.otp_view);

      otpTextView.setOtpListener(new OTPListener() {

         @Override

         public void onInteractionListener() {}


         @Override

         public void onOTPComplete(String otp) {

            if (!isTimerRunning) {

               showToast("Waktu verifikasi habis.
Silakan kirim ulang kode OTP.", R.layout.toast_gagal);

               return;

            }


            if (otp.equals(kodeotp)) {
```

```
            otpTextView.showSuccess();

            cekkodeotp(emailnohp, kodeotp);

        } else {

            otpTextView.showError();

            showToast("Kode OTP Salah", R.layout.toast_gagal);

        }

    }

});



findViewById(R.id.btnkirimulang).setOnClickListener(v -> kirimulangotp(emailnohp));



    Toolbar toolbar = findViewById(R.id.toolbar);

    setSupportActionBar(toolbar);

    ActionBar mAction = getSupportActionBar();

    assert mAction != null;

    TextView titleTextView = new TextView(getApplicationContext());

    titleTextView.setText("Verifikasi Kode OTP".toUpperCase());

    titleTextView.setTextSize(17);

    titleTextView.setTextColor(Color.WHITE);

    titleTextView.setGravity(Gravity.CENTER);


mAction.setDisplayOptions(ActionBar.DISPLAY_SHOW_CUSTOM);

    mAction.setCustomView(titleTextView, new ActionBar.LayoutParams(ActionBar.LayoutParams.WRAP_CONTENT,
ActionBar.LayoutParams.WRAP_CONTENT,
Gravity.CENTER));

}



private void startCountdownTimer() {

    countDownTimer = new CountDownTimer(60000, 1000) {

        @Override

        public void onTick(long millisUntilFinished) {

            int minutes = (int) (millisUntilFinished / 1000) / 60;

            int seconds = (int) (millisUntilFinished / 1000) % 60;


timerTextView.setText(String.format("%02d:%02d", minutes, seconds));

        }



        @Override

        public void onFinish() {

            timerTextView.setText("00:00");

            isTimerRunning = false;

            otpTextView.setEnabled(false);

            showToast("Waktu verifikasi habis. Silakan kirim ulang kode OTP.", R.layout.toast_gagal);

        }

    }.start();

}



private void kirimulangotp(String email) {

    progress.show();

    apiInterface = ApiClient.getApiClient().create(ApiInterface.class);

    Call<Data> masukCall = apiInterface.kirimotpRespone(email);

    masukCall.enqueue(new Callback<Data>() {

        @Override

        public void onResponse(@NotNull Call<Data> call, @NotNull Response<Data> response) {

            progress.hide();

            if (response.isSuccessful() && response.body() != null) {

                Data responseData = response.body();

                if (responseData.isStatus()) {

                    ResultData resultData = responseData.getResultData();

                    if (resultData != null) {
```

```
                    kodeotp = resultData.getOTPNumber();

showToast(responseData.getMessage(), R.layout.toast_berhasil);

                    countDownTimer.cancel();

                    isTimerRunning = true;

                    otpTextView.setEnabled(true);

                    startCountdownTimer();

                } else {

                    showToast("Failed to get OTP number from the server.", R.layout.toast_gagal);

                }

            } else {

                showToast(responseData.getMessage(), R.layout.toast_gagal);

            }

        } else {

            showToast("Terjadi kesalahan, silakan coba lagi.", R.layout.toast_gagal);

        }

    }



    @Override

    public void onFailure(@NotNull Call<Data> call, @NotNull Throwable t) {

        progress.hide();

        showToast("Tidak ada koneksi internet, silahkan coba lagi!", R.layout.toast_gagal);

    }

  });

 }



 private void cekkodeotp(String email, String kodeotp) {

    apiInterface = ApiClient.getApiClient().create(ApiInterface.class);

    Call<Data> masukCall = apiInterface.cekotpRespone(email, kodeotp);

    masukCall.enqueue(new Callback<Data>() {

      @Override

      public void onResponse(@NotNull Call<Data> call, @NotNull Response<Data> response) {

        if(response.body() != null && response.isSuccessful() && response.body().isStatus()) {

            showToast(response.body().getMessage(), R.layout.toast_berhasil);

            countDownTimer.cancel();

            Intent intent = new Intent(OTPNumberActivity.this, MasukActivity.class);

            startActivity(intent);

            finish();

        }else{

            assert response.body() != null;

            showToast(response.body().getMessage(), R.layout.toast_gagal);

        }

      }



      @Override

      public void onFailure(@NotNull Call<Data> call, @NotNull Throwable t) {

        showToast("Tidak ada koneksi internet, silahkan coba lagi!", R.layout.toast_gagal);

      }

    });

  }



 private void showToast(String message, int layoutId) {

    final Toast toast = new Toast(getApplicationContext());

    toast.setDuration(Toast.LENGTH_LONG);

    View custom_view = getLayoutInflater().inflate(layoutId, null);

    TextView tvMessage = custom_view.findViewById(R.id.message);

    tvMessage.setText(message);
```

```
        toast.setView(custom_view);
        toast.setGravity(Gravity.CENTER, 0, 0);
        toast.show();
    }
}
```

Script masuk activity

```
package com.dilla.a220280030.woparepare.activity;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.graphics.Paint;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.text.method.HideReturnsTransformationMethod;
import android.text.method.PasswordTransformationMethod;
import android.view.Gravity;
import android.view.View;
import android.widget.ImageView;
import android.widget.RelativeLayout;
import android.widget.TextView;
import android.widget.Toast;

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

import com.dilla.a220280030.woparepare.R;
import com.dilla.a220280030.woparepare.SessionManager;
import com.dilla.a220280030.woparepare.api.ApiClient;
import com.dilla.a220280030.woparepare.api.ApiInterface;
import com.dilla.a220280030.woparepare.model.Data;
import com.dilla.a220280030.woparepare.model.ResultData;
import com.google.android.material.textfield.TextInputEditText;
import com.google.android.material.textfield.TextInputLayout;

import java.util.Objects;

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

public class MasukActivity extends AppCompatActivity implements View.OnClickListener {

    private TextInputLayout textemlnohp, textpass;
    private TextInputEditText edtemlnohp, edtpass;
    private ImageView imgpass;
    private RelativeLayout btnmasuk;
    String emlnohp, pass;
    private TextView tvdaftar;
    ApiInterface apiInterface;
    SessionManager sessionManager;

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

        textemlnohp = findViewById(R.id.textemlnohp);
        textpass = findViewById(R.id.textpass);
        edtemlnohp = findViewById(R.id.edtemlnohp);
        edtemlnohp.addTextChangedListener(new TextWatcher() {
```

```
        @Override
        public void beforeTextChanged(CharSequence s, int start, int count, int after) {

        }

        @Override
        public void onTextChanged(CharSequence s, int start, int before, int count) {
            validemlnohp();
        }

        @Override
        public void afterTextChanged(Editable s) {

        }
    });
    edtpass = findViewById(R.id.edtpass);
    edtpass.addTextChangedListener(new TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence s, int start, int count, int after) {

        }

        @Override
        public void onTextChanged(CharSequence s, int start, int before, int count) {
            validpass();
        }

        @Override
        public void afterTextChanged(Editable s) {

        }
    });

    imgpass = findViewById(R.id.show_pass_btn);
    imgpass.setOnClickListener(v -> {
        if(v.getId()==R.id.show_pass_btn){
if(edtpass.getTransformationMethod().equals(PasswordTransformationMethod.getInstance())){
imgpass.setImageResource(R.drawable.hide_password);
edtpass.setTransformationMethod(HideReturnsTransformationMethod.getInstance());
            } else{
imgpass.setImageResource(R.drawable.show_password);
edtpass.setTransformationMethod(PasswordTransformationMethod.getInstance());
            }
        }
    });

    btnmasuk = findViewById(R.id.btnmasuk);
    btnmasuk.setOnClickListener(this);
    tvdaftar = findViewById(R.id.daftar);
tvdaftar.setPaintFlags(Paint.UNDERLINE_TEXT_FLAG);
    tvdaftar.setOnClickListener(this);
  }

  private void masuk(String emlnohp, String pass) {
    apiInterface = ApiClient.getApiClient().create(ApiInterface.class);
    Call<Data> masukCall = apiInterface.masukRespone(emlnohp, pass);
    masukCall.enqueue(new Callback<Data>() {
        @Override
```

```
        public void onResponse(@NonNull Call<Data> call, @NonNull Response<Data> response) {

            if (response.isSuccessful() && response.body() != null) {

                Data responseData = response.body();

                if (responseData.isStatus()) {

                    ResultData rsData = responseData.getResultData();

                    if (rsData != null) {

                        String message = responseData.getMessage();

                        if ("Akun Anda belum aktif, Silahkan cek Email untuk melakukan Verifikasi Akun.".equals(message)) {

                            showToast(message, R.layout.toast_info);

                            String kodeotp = rsData.getOTPNumber();

                            Intent intent = new Intent(MasukActivity.this, OTPNumberActivity.class);

                            intent.putExtra("EmailNoHP", emlnohp);

                            intent.putExtra("KodeOTP", kodeotp);

                            startActivity(intent);

                            finish();

                        } else {

                            sessionManager = new SessionManager(MasukActivity.this);


sessionManager.createLoginSession(

                                    rsData.getEmail(),

                                    rsData.getNoHP(),

                                    rsData.getNama(),

                                    rsData.getAlamat()

                            );

                            Intent intent = new Intent(MasukActivity.this, MenuActivity.class);

                            startActivity(intent);

                            finish();

                        }

                    } else {

                        showToast(responseData.getMessage(), R.layout.toast_gagal);

                    }

                } else {

                    showToast("Terjadi kesalahan, silakan coba lagi.", R.layout.toast_gagal);

                }

            }



            @Override

            public void onFailure(@NonNull Call<Data> call, @NonNull Throwable t) {

                showToast("Tidak ada koneksi internet, silahkan coba lagi!", R.layout.toast_gagal);

            }

        });

    }



    @Override

    public void onClick(View v){

        if(v.getId() == R.id.btnmasuk) {

            try {

                if (!validemlnohp() | !validpass()) {

                    return;

                }

                emlnohp = Objects.requireNonNull(edtemlnohp.getText()).toString();

                pass = Objects.requireNonNull(edtpass.getText()).toString();

                masuk(emlnohp, pass);

            } catch (Exception e) {

                e.printStackTrace();

            }

        }else if(v.getId() == R.id.daftar){
```

```
        startActivity(new Intent(MasukActivity.this, DaftarActivity.class));
        finish();
      }
   }

   private void showToast(String message, int layoutId) {
      final Toast toast = new Toast(getApplicationContext());
      toast.setDuration(Toast.LENGTH_LONG);
      @SuppressLint("InflateParams") View custom_view = getLayoutInflater().inflate(layoutId, null);
      TextView tvMessage = custom_view.findViewById(R.id.message);
      tvMessage.setText(message);
      toast.setView(custom_view);
      toast.setGravity(Gravity.CENTER, 0, 0);
      toast.show();
   }

   private boolean validemlnohp(){
      if (Objects.requireNonNull(edtemlnohp.getText()).toString().trim().isEmpty()) {
         textemlnohp.setError("Masukkan Email / No. HP");
         return false;
      } else {
         textemlnohp.setError(null);
         return true;
      }
   }

   private boolean validpass(){
      if (Objects.requireNonNull(edtpass.getText()).toString().trim().isEmpty()) {
         textpass.setError("Masukkan Password");
         return false;
      } else {
         textpass.setError(null);
         return true;
      }
   }
}
```

Script keranjang activity

```
package com.dilla.a220280030.woparepare.activity;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.graphics.Color;
import android.graphics.drawable.Drawable;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.Gravity;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
```

```
import androidx.appcompat.widget.Toolbar;

import androidx.cardview.widget.CardView;

import
androidx.core.graphics.drawable.DrawableCompat;

import
androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import
androidx.swiperefreshlayout.widget.SwipeRefreshLayo
ut;



import com.dilla.a220280030.woparepare.R;

import
com.dilla.a220280030.woparepare.SessionManager;

import
com.dilla.a220280030.woparepare.adapter.KeranjangA
dapter;

import
com.dilla.a220280030.woparepare.api.ApiClient;

import
com.dilla.a220280030.woparepare.api.ApiInterface;

import com.dilla.a220280030.woparepare.model.Data;

import
com.dilla.a220280030.woparepare.model.ResultData;



import org.jetbrains.annotations.NotNull;



import java.text.DecimalFormat;

import java.util.ArrayList;

import java.util.List;



import retrofit2.Call;

import retrofit2.Callback;

import retrofit2.Response;



public class KeranjangActivity extends
AppCompatActivity {



    private RecyclerView recyclerView;

    private SwipeRefreshLayout mSwipeRefreshLayout;

    private RelativeLayout errorlayout;

    private ImageView imgerror;

    private TextView errortitle, errormessage, totalhrg,
jmlcheckout;

    private Button btnretry;

    private CheckBox checkall;

    private CardView cardtransaksi;

    String email, nohp;

    KeranjangAdapter adapter;

    KeranjangAdapter.ItemClickListener
itemClickListener;

    List<ResultData> resultDataList;

    ApiInterface apiInterface;

    SessionManager sessionManager;



    @Override

    protected void onCreate(Bundle savedInstanceState)
{

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_keranjang);



        sessionManager = new
SessionManager(KeranjangActivity.this);

        email =
sessionManager.getUserDetail().get(SessionManager.E
MAIL);

        nohp =
sessionManager.getUserDetail().get(SessionManager.N
OHP);



        errorlayout = findViewById(R.id.rel_error);

        imgerror = findViewById(R.id.imgerror);

        errortitle = findViewById(R.id.tverror);

        errormessage = findViewById(R.id.tvsuberror);

        btnretry = findViewById(R.id.btnretry);

        checkall = findViewById(R.id.checkall);

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

```
        LinearLayout checkout = findViewById(R.id.checkout);

        totalhrg = findViewById(R.id.tvtotal);

        jmlcheckout = findViewById(R.id.tvcheckout);



        mSwipeRefreshLayout = findViewById(R.id.swaplistkeranjang);

        recyclerView = findViewById(R.id.listviewkeranjang);

        recyclerView.setLayoutManager(new LinearLayoutManager(this));

        recyclerView.setHasFixedSize(true);



        lihatisikeranjang(email);



mSwipeRefreshLayout.setColorSchemeResources(R.color.colorPrimary);

        mSwipeRefreshLayout.setOnRefreshListener(() -> {

            lihatisikeranjang(email);

            checkall.setChecked(false);

            totalhrg.setText("Rp0");

            jmlcheckout.setText("Checkout (0)");

        });



        itemClickListener = new KeranjangAdapter.ItemClickListener() {

            @Override

            public void OnItemClickKurang(View view, int position) {

                String idkategori = resultDataList.get(position).getIdKategori();

                String idproduk = resultDataList.get(position).getIdProduk();

                int jmll = Integer.parseInt(resultDataList.get(position).getJumlahData());

                if (jmll == 1){

                    String hexColor = "#B10000";

                    int color = Color.parseColor(hexColor);

                    AlertDialog alertDialog = new AlertDialog.Builder(KeranjangActivity.this)

                            .setTitle("Konfirmasi Hapus!")

                            .setMessage("Apakah Anda yakin untuk menghapus?")

                            .setPositiveButton("Ya", (dialog, which) -> {

                                hapuskeranjang(idkategori, idproduk, email);

                                resultDataList.remove(position);


adapter.notifyItemRemoved(position);

                                updateTotals();

                            })

                            .setNegativeButton("Tidak", null)

                            .setIcon(R.drawable.ic_close)

                            .show();

                    Drawable icon = getResources().getDrawable(R.drawable.ic_close);

                    DrawableCompat.setTint(icon, color);

                    alertDialog.setIcon(icon);

                } else {

                    jmll--;


resultDataList.get(position).setJumlahData(String.valueOf(jmll));

                    ResultData movedItem = resultDataList.remove(position);

                    resultDataList.add(0, movedItem);

                    adapter.notifyItemMoved(position, 0);

                    adapter.notifyItemChanged(0);

                    hapuskeranjang(idkategori, idproduk, email);

                    updateTotals();

                }

            }


            @Override
```

```
        public void OnItemClickTambah(View view, int position) {

            String idkategori = resultDataList.get(position).getIdKategori();

            String idproduk = resultDataList.get(position).getIdProduk();

            int jmll = Integer.parseInt(resultDataList.get(position).getJumlahData());

            tambahkeranjang(email, nohp, idkategori, idproduk);

            jmll++;


resultDataList.get(position).setJumlahData(String.valueOf(jmll));

            ResultData movedItem = resultDataList.remove(position);

            resultDataList.add(0, movedItem);

            adapter.notifyItemMoved(position, 0);

            adapter.notifyItemChanged(0);

            updateTotals();

        }



        @Override

        public void OnItemClickCheckItem(View view, int position) {

            updateTotals();

            boolean allUnchecked = true;

            for (ResultData item : resultDataList) {

                if (item.isChecked()) {

                    allUnchecked = false;

                    break;

                }

            }

            if (allUnchecked) {

                checkall.setChecked(false);

            }

        }

    };
```

```
        totalhrg.setText("Rp0");

        jmlcheckout.setText("Checkout (0)");


checkall.setOnCheckedChangeListener((buttonView, isChecked) -> {

            for (ResultData item : resultDataList) {

                item.setChecked(isChecked);

            }

            adapter.notifyDataSetChanged();

            updateTotals();

        });

        checkout.setOnClickListener(v -> {

            ArrayList<String> idKategoriList = new ArrayList<>();

            ArrayList<String> idProdukList = new ArrayList<>();

            ArrayList<String> namaKategoriList = new ArrayList<>();

            ArrayList<String> namaProdukList = new ArrayList<>();

            ArrayList<String> hargaList = new ArrayList<>();

            ArrayList<String> jumlahList = new ArrayList<>();

            ArrayList<String> gambarList = new ArrayList<>();

            int sumjml = 0;

            int sumttl = 0;



            for (ResultData item : resultDataList) {

                if (item.isChecked()) {

                    sumjml += Integer.parseInt(item.getJumlahData());

                    sumttl += Integer.parseInt(item.getJumlahData()) * Integer.parseInt(item.getHargaProduk());

                    idKategoriList.add(item.getIdKategori());

                    idProdukList.add(item.getIdProduk());


namaKategoriList.add(item.getNamaKategori());
```

```
namaProdukList.add(item.getNamaProduk());
                hargaList.add(item.getHargaProduk());
                jumlahList.add(item.getJumlahData());
                gambarList.add(item.getGambarProduk());
            }
        }

        if (sumjml == 0) {
            showToast("Produk belum dipilih", R.layout.toast_info);
        } else {
            String idtegori = TextUtils.join(", ", idKategoriList);
            String idproduk = TextUtils.join(", ", idProdukList);
            String namategori = TextUtils.join(", ", namaKategoriList);
            String namaproduk = TextUtils.join(", ", namaProdukList);
            String harga = TextUtils.join(", ", hargaList);
            String jml = TextUtils.join(", ", jumlahList);
            String gambar = TextUtils.join(", ", gambarList);

            Intent intent = new Intent(KeranjangActivity.this, CheckoutActivity.class);
            intent.putExtra("IDKATEGORI", idtegori);
            intent.putExtra("IDPRODUK", idproduk);
            intent.putExtra("NAMAKATEGORI", namategori);
            intent.putExtra("NAMAPRODUK", namaproduk);
            intent.putExtra("HARGA", harga);
            intent.putExtra("JUMLAH", jml);
            intent.putExtra("GAMBAR", gambar);
            intent.putExtra("TOTAL_JUMLAH", sumjml);
            intent.putExtra("TOTAL_HARGA", sumttl);
            startActivity(intent);
        }
    });

    Toolbar toolbar = findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);
    ActionBar mAction = getSupportActionBar();
    assert mAction != null;
    TextView titleTextView = new TextView(getApplicationContext());

titleTextView.setText("Keranjang".toUpperCase());
    titleTextView.setTextSize(17);
    titleTextView.setTextColor(Color.WHITE);
    titleTextView.setGravity(Gravity.CENTER);

mAction.setDisplayOptions(ActionBar.DISPLAY_SHOW_CUSTOM);
    mAction.setCustomView(titleTextView, new ActionBar.LayoutParams(ActionBar.LayoutParams.WRAP_CONTENT,
ActionBar.LayoutParams.WRAP_CONTENT,
Gravity.CENTER));
    mAction.setDisplayHomeAsUpEnabled(true);
    toolbar.setNavigationOnClickListener(v -> onBackPressed());
  }

  private void updateTotals() {
    int sumjml = 0;
    int sumttl = 0;
    for (ResultData item : resultDataList) {
      if (item.isChecked()) {
        sumjml += Integer.parseInt(item.getJumlahData());
        sumttl += Integer.parseInt(item.getJumlahData()) * Integer.parseInt(item.getHargaProduk());
      }
    }
```

```
        String hargaFormatted = formatRupiah(sumttl);
        totalhrg.setText(hargaFormatted);
        jmlcheckout.setText("Checkout (" + sumjml + ")");
    }

    private String formatRupiah(int number) {
        DecimalFormat formatter = new DecimalFormat("#,###,###,###");
        return "Rp" + formatter.format(number);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.data_diri, menu);
        MenuItem keranjangItem = menu.findItem(R.id.keranjang);
        keranjangItem.setVisible(false);
        MenuItem datadiri = menu.findItem(R.id.data_diri);
        datadiri.setVisible(false);
        return true;
    }

    @SuppressLint("SetTextI18n")
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        if (item.getItemId() == android.R.id.home) {
            onBackPressed();
        } else if (item.getItemId() == R.id.listmenu) {
            startActivity(new Intent(KeranjangActivity.this, MenuActivity.class));
            finish();
        }
        return super.onOptionsItemSelected(item);
    }

    private void lihatisikeranjang(String email) {
        errorlayout.setVisibility(View.GONE);
        showLoading();
        apiInterface = ApiClient.getApiClient().create(ApiInterface.class);
        Call<List<ResultData>> call = apiInterface.getTampilKeranjang(email);
        call.enqueue(new Callback<List<ResultData>>() {

            @Override
            public void onResponse(@NonNull Call<List<ResultData>> call, @NonNull Response<List<ResultData>> response) {
                hideLoading();
                if(response.isSuccessful() && response.body() != null){
                    onGetResult(response.body());

if(response.body().get(0).getIfnya().equals("Tidak")){
                        cardtransaksi.setVisibility(View.GONE);
                    } else {

cardtransaksi.setVisibility(View.VISIBLE);
                    }
                }else{
                    String errorCode;
                    switch (response.code()){
                        case 404:
                            errorCode = "404 not found";
                            break;
                        case 500:
                            errorCode = "500 server broken";
                            break;
                        default:
                            errorCode = "Unknown error";
```

```
                    break;

                }

                showErrorMessage(R.drawable.no_result,
"No Result", "Please Try Again!-"+ errorCode);

            }

        }



        @Override

        public void onFailure(@NonNull
Call<List<ResultData>> call, @NonNull Throwable t)
{

            hideLoading();

            onErrorLoading(t.toString());

        }

    });

  }



  private void tambahkeranjang(String email, String
nohp, String idkate, String idproduk) {

    apiInterface =
ApiClient.getApiClient().create(ApiInterface.class);

    Call<Data> masukCall =
apiInterface.tambahkeranjangRespone(idkate, idproduk,
email, nohp);

    masukCall.enqueue(new Callback<Data>() {

        @Override

        public void onResponse(@NotNull Call<Data>
call, @NotNull Response<Data> response) {

            if(response.body() != null &&
response.isSuccessful() && response.body().isStatus())
{

//                showToast(response.body().getMessage(),
R.layout.toast_berhasil);

            }else{

                assert response.body() != null;

                showToast(response.body().getMessage(),
R.layout.toast_gagal);

            }

        }

        @SuppressLint("SetTextI18n")

        @Override

        public void onFailure(@NotNull Call<Data>
call, @NotNull Throwable t) {

            showToast("Tidak ada koneksi internet,
silahkan coba lagi!", R.layout.toast_gagal);

        }

    });

  }



  private void hapuskeranjang(String idkategori, String
idproduk, String email) {

    apiInterface =
ApiClient.getApiClient().create(ApiInterface.class);

    Call<Data> masukCall =
apiInterface.hapuskeranjangRespone(idkategori,
idproduk, email);

    masukCall.enqueue(new Callback<Data>() {

        @Override

        public void onResponse(@NotNull Call<Data>
call, @NotNull Response<Data> response) {

            if(response.body() != null &&
response.isSuccessful() && response.body().isStatus())
{

//                showToast(response.body().getMessage(),
R.layout.toast_hijau);

            }else{

                assert response.body() != null;

                showToast(response.body().getMessage(),
R.layout.toast_gagal);

            }

        }

        @SuppressLint("SetTextI18n")

        @Override

        public void onFailure(@NotNull Call<Data>
call, @NotNull Throwable t) {

            showToast("Tidak ada koneksi internet,
silahkan coba lagi!", R.layout.toast_gagal);

        }
```

```
        });

    }


    private void showToast(String message, int layoutId)
{

        final Toast toast = new
Toast(getApplicationContext());

        toast.setDuration(Toast.LENGTH_LONG);

        @SuppressLint("InflateParams") View
custom_view = getLayoutInflater().inflate(layoutId,
null);

        TextView tvMessage =
custom_view.findViewById(R.id.message);

        tvMessage.setText(message);

        toast.setView(custom_view);

        toast.setGravity(Gravity.CENTER, 0, 0);

        toast.show();

    }


    private void showLoading() {

        mSwipeRefreshLayout.setRefreshing(true);

    }


    private void hideLoading() {

        mSwipeRefreshLayout.setRefreshing(false);

    }


    @SuppressLint("NotifyDataSetChanged")

    private void onGetResult(List<ResultData> values) {

        adapter = new KeranjangAdapter(this, values,
itemClickListener);

        adapter.notifyDataSetChanged();

        recyclerView.setAdapter(adapter);

        resultDataList = values;

    }


    private void onErrorLoading(String message) {

        showErrorMessage(R.drawable.no_result,
"Oops...", "Network failure, Please Try Again!.-"+
message);

    }


    private void showErrorMessage(int imageView,
String title, String message){


        if(errorlayout.getVisibility() == View.GONE){

            errorlayout.setVisibility(View.VISIBLE);

        }


        imgerror.setImageResource(imageView);

        errortitle.setText(title);

        errormessage.setText(message);


        btnretry.setOnClickListener(view ->
lihatisikeranjang(email));

    }

}
```

Script konfirmasi bayar activity

```
package com.dilla.a220280030.woparepare.activity;


import android.Manifest;

import android.annotation.SuppressLint;

import android.app.AlertDialog;

import android.app.ProgressDialog;

import android.content.ClipData;

import android.content.ClipboardManager;

import android.content.ContentValues;

import android.content.Context;

import android.content.Intent;

import android.content.pm.PackageManager;

import android.database.Cursor;

import android.graphics.Color;

import android.net.Uri;
```

```
import android.os.Bundle;
import android.provider.MediaStore;
import android.view.Gravity;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.core.content.ContextCompat;

import com.dilla.a220280030.woparepare.R;
import com.dilla.a220280030.woparepare.api.ApiClient;
import com.dilla.a220280030.woparepare.api.ApiInterface;
import com.dilla.a220280030.woparepare.model.Data;
import com.google.android.material.bottomsheet.BottomSheetDialog;

import org.jetbrains.annotations.NotNull;

import java.io.File;
import java.text.DecimalFormat;

import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class KonfirmasiBayarActivity extends AppCompatActivity {

    String idtransaksi, metodebyr, total, part_image;
    private ProgressDialog progress;
    private ImageView imgunggahbukti;
    private TextView tvnmbank;
    ApiInterface apiInterface;
    private static final int CAMERA_REQUEST = 100;
    private static final int STORAGE_REQUEST = 101;
    private static final int IMAGE_CAPTURE_CODE = 102;
    private static final int PICK_IMAGE_CODE = 103;
    private Uri imageUri;

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

        Intent get = getIntent();
        idtransaksi = get.getStringExtra("IdTransaksi");
        total = get.getStringExtra("TotalHarga");
        metodebyr = get.getStringExtra("MetodeBayar");

        ImageView imgbank = findViewById(R.id.imgbank);
        tvnmbank = findViewById(R.id.tvnmbank);
        TextView tvatsnm = findViewById(R.id.tvatsnm);
        TextView tvnorekbank = findViewById(R.id.tvnorekbank);
        tvatsnm.setText("An: Indobotting Parepare");
```

```
        tvnmbank.setText(metodebyr);

        setBankDetails(metodebyr, imgbank, tvnorekbank);

        TextView tvnominarbyar = findViewById(R.id.tvnominarbyar);

        String rupiah = formatRupiah(Integer.parseInt(total));

        tvnominarbyar.setText(rupiah);

        ImageButton btnsalinnorekbank = findViewById(R.id.btnsalinnorekbank);

        btnsalinnorekbank.setOnClickListener(v -> {

            String bankAccountNumber = tvnorekbank.getText().toString();

            ClipboardManager clipboard = (ClipboardManager) getSystemService(Context.CLIPBOARD_SERVICE);

            ClipData clip = ClipData.newPlainText("Nomor Rekening", bankAccountNumber);

            clipboard.setPrimaryClip(clip);

            showToast("Nomor rekening berhasil disalin", R.layout.toast_info);

        });

        Button btnubahmetodebyr = findViewById(R.id.btnubahmetodebyr);

        btnubahmetodebyr.setOnClickListener(v -> {

            BottomSheetDialog bottomSheetDialog = new BottomSheetDialog(KonfirmasiBayarActivity.this);


bottomSheetDialog.setContentView(R.layout.metode_bayar);



//                TextView paymentMethodCOD = bottomSheetDialog.findViewById(R.id.payment_method_cod);

            TextView paymentBSI = bottomSheetDialog.findViewById(R.id.payment_method_bsi);

            TextView paymentBRI = bottomSheetDialog.findViewById(R.id.payment_method_bri);

            TextView paymentBCA = bottomSheetDialog.findViewById(R.id.payment_method_bca);

            TextView paymentBNI = bottomSheetDialog.findViewById(R.id.payment_method_bni);

            TextView paymentMandiri = bottomSheetDialog.findViewById(R.id.payment_method_mandiri);



            final TextView[] lastSelected = {null};

            View.OnClickListener onClickListener = view -> {

                TextView selectedTextView = (TextView) view;

                String selectedMethod = selectedTextView.getText().toString();

//                if (selectedTextView == paymentMethodCOD) {

//                    tvnmbank.setText(selectedMethod);

//                } else {

                tvnmbank.setText(selectedMethod);

                setBankDetails(selectedMethod, imgbank, tvnorekbank);

//                }

                lastSelected[0] = selectedTextView;

                bottomSheetDialog.dismiss();

            };



//
paymentMethodCOD.setOnClickListener(onClickListener);


paymentBSI.setOnClickListener(onClickListener);


paymentBRI.setOnClickListener(onClickListener);


paymentBCA.setOnClickListener(onClickListener);


paymentBNI.setOnClickListener(onClickListener);


paymentMandiri.setOnClickListener(onClickListener);

            bottomSheetDialog.show();

        });



        progress = new ProgressDialog(this);

        progress.setCancelable(false);
```

```
        progress.setMessage("Mohon Tunggu...");



        imgunggahbukti = findViewById(R.id.imgunggahberkas);

        imgunggahbukti.setOnClickListener(v -> tampilGambarDialog());


findViewById(R.id.btnunggahbukti).setOnClickListener(v -> {

            if(imgunggahbukti.getDrawable() == null){

                showToast("Belum Ada File Yang Dipilih", R.layout.toast_info);

            }else{

                unggahbuktitransfer();

            }

        });



        Toolbar toolbar = findViewById(R.id.toolbar);

        setSupportActionBar(toolbar);

        ActionBar mAction = getSupportActionBar();

        assert mAction != null;

        TextView titleTextView = new TextView(getApplicationContext());

        titleTextView.setText("Konfirmasi Pembayaran".toUpperCase());

        titleTextView.setTextSize(17);

        titleTextView.setTextColor(Color.WHITE);

        titleTextView.setGravity(Gravity.CENTER);


mAction.setDisplayOptions(ActionBar.DISPLAY_SHOW_CUSTOM);

        mAction.setCustomView(titleTextView, new ActionBar.LayoutParams(ActionBar.LayoutParams.WRAP_CONTENT,
ActionBar.LayoutParams.WRAP_CONTENT,
Gravity.CENTER));

        mAction.setDisplayHomeAsUpEnabled(true);

        toolbar.setNavigationOnClickListener(v -> onBackPressed());

    }

    @SuppressLint("SetTextI18n")

    @Override

    public boolean onOptionsItemSelected(MenuItem item) {

        if (item.getItemId() == android.R.id.home) {

            onBackPressed();

        }

        return super.onOptionsItemSelected(item);

    }



    private void setBankDetails(String metodebyr, ImageView imgnya, TextView tvnya) {

        switch (metodebyr) {

            case "Bank Syariah Indonesia (BSI)":


imgnya.setBackgroundResource(R.drawable.bsi);

                tvnya.setText("45863202010");

                break;

            case "Bank Rakyat Indonesia (BRI)":


imgnya.setBackgroundResource(R.drawable.bri);

                tvnya.setText("01660102020588");

                break;

            case "Bank Central Asia (BCA)":


imgnya.setBackgroundResource(R.drawable.bca);

                tvnya.setText("5220201023");

                break;

            case "Bank Negara Indonesia (BNI)":


imgnya.setBackgroundResource(R.drawable.bni);

                tvnya.setText("6320152020");

                break;

            case "Bank Mandiri":


imgnya.setBackgroundResource(R.drawable.mandiri);

                tvnya.setText("5301202010230");
```

```
            break;

        }

    }



    private void showToast(String message, int layoutId)
{

        final Toast toast = new
Toast(getApplicationContext());

        toast.setDuration(Toast.LENGTH_LONG);

        @SuppressLint("InflateParams") View
custom_view = getLayoutInflater().inflate(layoutId,
null);

        TextView tvMessage =
custom_view.findViewById(R.id.message);

        tvMessage.setText(message);

        toast.setView(custom_view);

        toast.setGravity(Gravity.CENTER, 0, 0);

        toast.show();

    }



    private String formatRupiah(int number) {

        DecimalFormat formatter = new
DecimalFormat("#,###,###,###");

        return "Rp" + formatter.format(number);

    }



    private void unggahbuktitransfer(){

        progress.show();

        File imagefile = new File(part_image);

        String nmbank = tvnmbank.getText().toString();

        RequestBody idnya =
RequestBody.create(MultipartBody.FORM,idtransaksi)
;

        RequestBody metodenya =
RequestBody.create(MultipartBody.FORM,nmbank);

        RequestBody reqBody =
RequestBody.create(MediaType.parse("multipart/form-
file"),imagefile);

        MultipartBody.Part partImage =
MultipartBody.Part.createFormData("UnggahBuktiTran
sfer", imagefile.getName(),reqBody);



        apiInterface =
ApiClient.getApiClient().create(ApiInterface.class);

        Call<Data> masukCall =
apiInterface.unggahbuktiRespone(idnya,metodenya,part
Image);

        masukCall.enqueue(new Callback<Data>() {

            @Override

            public void onResponse(@NotNull Call<Data>
call, @NotNull Response<Data> response) {

                progress.hide();

                if(response.body() != null &&
response.isSuccessful() && response.body().isStatus())
{

                    showToast(response.body().getMessage(),
R.layout.toast_berhasil);

                    Intent it = new
Intent(KonfirmasiBayarActivity.this,
TransaksiActivity.class);

                    it.putExtra("SELECTED_TAB", 0);

                    startActivity(it);

                    finish();

                }else{

                    showToast(response.body().getMessage(),
R.layout.toast_gagal);

                }

            }



            @SuppressLint("SetTextI18n")

            @Override

            public void onFailure(@NotNull Call<Data>
call, @NotNull Throwable t) {

                progress.hide();

                showToast("Tidak Ada Koneksi Internet,
Silahkan coba lagi!", R.layout.toast_gagal);

            }

        });

    }
```

```
    @SuppressLint("SetTextI18n")

    private void tampilGambarDialog() {

        String[] options = {"Kamera", "Galeri"};

        AlertDialog.Builder builder = new
AlertDialog.Builder(this);

        builder.setTitle("Pilih berkas dari:");

        builder.setItems(options, (dialog, which) -> {

            if (which == 0) {

                if (!checkCameraPermission()) {

                    requestCameraPermission();

                } else {

                    openCamera();

                }

            } else if (which == 1) {

                if (!checkStoragePermission()) {

                    requestStoragePermission();

                } else {

                    openGallery();

                }

            }

        });

        builder.create().show();

    }


    private Boolean checkStoragePermission() {

        return ContextCompat.checkSelfPermission(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE
) == PackageManager.PERMISSION_GRANTED;

    }


    private void requestStoragePermission() {

        requestPermissions(new
String[]{Manifest.permission.WRITE_EXTERNAL_S
TORAGE}, STORAGE_REQUEST);

    }


    private Boolean checkCameraPermission() {

        return ContextCompat.checkSelfPermission(this,
Manifest.permission.CAMERA) ==
PackageManager.PERMISSION_GRANTED;

    }


    private void requestCameraPermission() {

        requestPermissions(new
String[]{Manifest.permission.CAMERA},
CAMERA_REQUEST);

    }


    private void openCamera() {

        ContentValues values = new ContentValues();

        values.put(MediaStore.Images.Media.TITLE,
"New Picture");


values.put(MediaStore.Images.Media.DESCRIPTION,
"From the Camera");

        imageUri =
getContentResolver().insert(MediaStore.Images.Media.
EXTERNAL_CONTENT_URI, values);

        Intent takePicture = new
Intent(MediaStore.ACTION_IMAGE_CAPTURE);


takePicture.putExtra(MediaStore.EXTRA_OUTPUT,
imageUri);

        startActivityForResult(takePicture,
IMAGE_CAPTURE_CODE);

    }


    private void openGallery() {

        Intent pickPhoto = new
Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_U
RI);

        startActivityForResult(pickPhoto,
PICK_IMAGE_CODE);

    }


    @Override
```

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

        super.onRequestPermissionsResult(requestCode, permissions, grantResults);

        if (requestCode == CAMERA_REQUEST) {

            if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {

                openCamera();

            } else {

                showToast("Harap Aktifkan Izin Kamera", R.layout.toast_info);

            }

        } else if (requestCode == STORAGE_REQUEST) {

            if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {

                openGallery();

            } else {

                showToast("Harap Aktifkan Izin Penyimpanan", R.layout.toast_gagal);

            }

        }

    }



    @Override

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

        super.onActivityResult(requestCode, resultCode, data);

        if (resultCode == RESULT_OK) {

            if (requestCode == IMAGE_CAPTURE_CODE) {

                imgunggahbukti.setImageURI(imageUri);

                part_image = getPathFromUri(imageUri);

            } else if (requestCode == PICK_IMAGE_CODE && data != null) {

                Uri dataImage = data.getData();

                imgunggahbukti.setImageURI(dataImage);

                part_image = getPathFromUri(dataImage);

            }

        }

    }



    private String getPathFromUri(Uri uri) {

        String[] projection = {MediaStore.Images.Media.DATA};

        Cursor cursor = getContentResolver().query(uri, projection, null, null, null);

        if (cursor != null) {

            cursor.moveToFirst();

            int index = cursor.getColumnIndex(projection[0]);

            String path = cursor.getString(index);

            cursor.close();

            return path;

        }

        return null;

    }



//    @SuppressLint("SetTextI18n")

//    private void tampilGambarDialog() {

//        String[] options = {"Kamera", "Galeri"};

//                AlertDialog.Builder builder = new AlertDialog.Builder(this);

//        builder.setTitle("Pilih Gambar Dari :");

//        builder.setItems(options, (dialog, which) -> {

//            if (which == 0) {

//                if (!checkCameraPermission()) {

//                    requestCameraPermission();

//                } else {

//                                Intent takePicture = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);

//                    startActivityForResult(takePicture, CAMERA_REQUEST);

//                }
```

```
//// showToast("Kamera Belum Bisa Digunakan", R.layout.toast_info);

//         } else if (which == 1) {

//             if (!checkStoragePermission()) {

//                 requestStoragePermission();

//             } else {

//                                 Intent pickPhoto = new Intent(Intent.ACTION_PICK, MediaStore.Images.Media.EXTERNAL_CONTENT_URI);

//                         startActivityForResult(pickPhoto , STORAGE_REQUEST);

//             }

//         }

//     });

//     builder.create().show();

// }

//

// private Boolean checkStoragePermission() {

//     return ContextCompat.checkSelfPermission(this, Manifest.permission.WRITE_EXTERNAL_STORAGE) == (PackageManager.PERMISSION_GRANTED);

// }

//

// private void requestStoragePermission() {

//                 if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {

//                                 requestPermissions(new String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE}, STORAGE_REQUEST);

//     }

// }

//

// private Boolean checkCameraPermission() {

//     return ContextCompat.checkSelfPermission(this, Manifest.permission.CAMERA) == (PackageManager.PERMISSION_GRANTED);

// }

//

// private void requestCameraPermission() {

//                 if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {

//                                 requestPermissions(new String[]{Manifest.permission.CAMERA}, CAMERA_REQUEST);

//     }

// }

//

// @SuppressLint("SetTextI18n")

// @Override

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

//     super.onRequestPermissionsResult(requestCode, permissions, grantResults);

//     switch (requestCode) {

//         case CAMERA_REQUEST: {

//             if (grantResults.length > 0) {

//                                 boolean camera_accepted = grantResults[0] == PackageManager.PERMISSION_GRANTED;

//                 if (camera_accepted) {

//                                     Intent takePicture = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);

//                         startActivityForResult(takePicture, CAMERA_REQUEST);

//                 } else {

//                     showToast("Harap Aktifkan Izin Kamera", R.layout.toast_info);

//                 }

//// showToast("Kamera Belum Bisa Digunakan", R.layout.toast_info);

//             }

//         }

//         break;

//         case STORAGE_REQUEST: {

//             if (grantResults.length > 0) {

//                                 boolean writeStorageaccepted = grantResults[0] == PackageManager.PERMISSION_GRANTED;

//                 if (writeStorageaccepted) {
```

```
//                                  Intent pickPhoto = new Intent(Intent.ACTION_PICK, MediaStore.Images.Media.EXTERNAL_CONTENT_URI);

//                                  startActivityForResult(pickPhoto, STORAGE_REQUEST);

//                  } else {

//                                  showToast("Harap Aktifkan Izin Penyimpanan", R.layout.toast_gagal);

//                  }

//              }

//          }

//          break;

//      }

//  }

//

//  @SuppressLint("SetTextI18n")

//  @Override

//      protected void onActivityResult(int requestCode, int resultCode, Intent data) {

//          super.onActivityResult(requestCode, resultCode, data);

//          Uri dataimage;

//          if(requestCode == CAMERA_REQUEST) {

//              if(resultCode == RESULT_OK){

//                                  Bitmap photo = (Bitmap) data.getExtras().get("data");

//                  imgunggahbukti.setImageBitmap(photo);

//              }

//                                  }else if(requestCode == STORAGE_REQUEST){

//              if(resultCode == RESULT_OK){

//                  dataimage = data.getData();

//                                  String[] imageprojection = {MediaStore.Images.Media.DATA};

//                  @SuppressLint("Recycle") Cursor cursor = getContentResolver().query(dataimage,imageprojection,null,null,null);

//

//                  if (cursor != null){

//                      cursor.moveToFirst();

//                                  int indexImage = cursor.getColumnIndex(imageprojection[0]);

//                                  part_image = cursor.getString(indexImage);

//

//                  if(part_image != null){

//                      File image = new File(part_image);

//                                  double filesize = Double.parseDouble(String.valueOf(image.length() / 1024));

//                      if(filesize >= 1024) {

//                                  showToast("Ukuran Gambar Maksimal 1 MB", R.layout.toast_info);

//                      }else{

//  imgunggahbukti.setImageBitmap(BitmapFactory.decodeFile(image.getAbsolutePath()));

//                      }

//                  }

//              }

//          }

//      }

//  }

}
```

Script rincian pesanan activity

```
package com.dilla.a220280030.woparepare.activity;


import android.annotation.SuppressLint;

import android.content.Intent;

import android.graphics.Color;

import android.os.Bundle;

import android.view.Gravity;

import android.view.MenuItem;

import android.view.View;

import android.widget.LinearLayout;

import android.widget.TextView;
```

```
import android.widget.Toast;


import androidx.annotation.NonNull;

import androidx.appcompat.app.ActionBar;

import androidx.appcompat.app.AppCompatActivity;

import androidx.appcompat.widget.Toolbar;

import
androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;


import com.dilla.a220280030.woparepare.R;

import
com.dilla.a220280030.woparepare.SessionManager;

import
com.dilla.a220280030.woparepare.adapter.RincianPesa
nanAdapter;

import
com.dilla.a220280030.woparepare.api.ApiClient;

import
com.dilla.a220280030.woparepare.api.ApiInterface;

import
com.dilla.a220280030.woparepare.model.ResultData;


import java.text.DecimalFormat;

import java.util.List;


import retrofit2.Call;

import retrofit2.Callback;

import retrofit2.Response;


public class RincianPesananActivity extends
AppCompatActivity {


   private RecyclerView recyclerView;

   String email, nohp, nama, alamat, idtransaksi,
statusByr, statusBrng, totalJumlah, totalHarga;

   SessionManager sessionManager;

   RincianPesananAdapter adapter;

   List<ResultData> resultDataList;

   ApiInterface apiInterface;


   @Override

   protected void onCreate(Bundle savedInstanceState)
{

      super.onCreate(savedInstanceState);


setContentView(R.layout.activity_rincian_pesanan);


      sessionManager = new
SessionManager(RincianPesananActivity.this);

      email =
sessionManager.getUserDetail().get(SessionManager.E
MAIL);

      nohp =
sessionManager.getUserDetail().get(SessionManager.N
OHP);

      nama =
sessionManager.getUserDetail().get(SessionManager.N
AMA);

      alamat =
sessionManager.getUserDetail().get(SessionManager.A
LAMAT);


      Intent intent = getIntent();

      idtransaksi =
intent.getStringExtra("IDTRANSAKSI");

      statusByr =
intent.getStringExtra("STATUSBAYAR");

      statusBrng =
intent.getStringExtra("STATUSBARANG");

      totalJumlah =
intent.getStringExtra("TOTAL_JUMLAH");

      totalHarga =
intent.getStringExtra("TOTAL_HARGA");


      lihatrincianpesanan(email, idtransaksi);


      recyclerView =
findViewById(R.id.listviewrincian);

      recyclerView.setLayoutManager(new
LinearLayoutManager(this));
```

```
        recyclerView.setHasFixedSize(true);


        LinearLayout linearstatus = findViewById(R.id.linearstatus);
        TextView statusbarang = findViewById(R.id.statusbarang);
        switch (statusByr) {
            case "T":
                linearstatus.setBackgroundColor(Color.parseColor("#D85662"));
                statusbarang.setText("Pesanan Belum Dibayar");
                break;
            case "P":
                linearstatus.setBackgroundColor(Color.parseColor("#D8944B"));
                statusbarang.setText("Pembayaran Sedang Dicek");
                break;
            case "Y":
                switch (statusBrng) {
                    case "DP":
                        linearstatus.setBackgroundColor(Color.parseColor("#BED754"));
                        statusbarang.setText("Pesanan Sedang Diproses");
                        break;
                    case "DS":
                        linearstatus.setBackgroundColor(Color.parseColor("#9EC8B9"));
                        statusbarang.setText("Pesanan Sedang Disiapkan");
                        break;
                    case "ML":
                        linearstatus.setBackgroundColor(Color.parseColor("#F05941"));
                        statusbarang.setText("Pesanan Sedang Menuju Ke Lokasi");
                        break;
                    case "DT":
                        linearstatus.setBackgroundColor(Color.parseColor("#D27685"));
                        statusbarang.setText("Pesanan Sudah Dipasang/Diterima");
                        break;
                    case "SS":
                        linearstatus.setBackgroundColor(Color.parseColor("#52AF39"));
                        statusbarang.setText("Pesanan Selesai");
                        break;
                }
                break;
        }


        TextView tvalamat = findViewById(R.id.tvDetailAlamat);
        tvalamat.setText(nama + "\n" + email + " | " + nohp + "\n" + alamat);


        TextView tvTotalJml = findViewById(R.id.tvtotaljml);
        TextView tvTotalByr = findViewById(R.id.tvtotalbyr);
        tvTotalJml.setText("Total Pembelian (" + totalJumlah + " Produk):");
        tvTotalByr.setText(formatRupiah(Integer.parseInt(totalHarga)));


        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        ActionBar mAction = getSupportActionBar();
        assert mAction != null;
        TextView titleTextView = new TextView(getApplicationContext());
```

```
        titleTextView.setText("Rincian
Pesanan".toUpperCase());

        titleTextView.setTextSize(17);

        titleTextView.setTextColor(Color.WHITE);

        titleTextView.setGravity(Gravity.CENTER);


mAction.setDisplayOptions(ActionBar.DISPLAY_SH
OW_CUSTOM);

        mAction.setCustomView(titleTextView,  new
ActionBar.LayoutParams(ActionBar.LayoutParams.W
RAP_CONTENT,
ActionBar.LayoutParams.WRAP_CONTENT,
Gravity.CENTER));

        mAction.setDisplayHomeAsUpEnabled(true);

        toolbar.setNavigationOnClickListener(v  ->
onBackPressed());

    }



    @SuppressLint("SetTextI18n")

    @Override

    public  boolean  onOptionsItemSelected(MenuItem
item) {

        if (item.getItemId() == android.R.id.home) {

            onBackPressed();

        }

        return super.onOptionsItemSelected(item);

    }



    private void lihatrincianpesanan(String email, String
idtransaksi) {

        apiInterface                                =
ApiClient.getApiClient().create(ApiInterface.class);

        Call<List<ResultData>>          call          =
apiInterface.getTampilRincianPembayaran(email,
idtransaksi);

        call.enqueue(new Callback<List<ResultData>>() {

            @Override

            public  void  onResponse(@NonNull
Call<List<ResultData>>  call,  @NonNull
Response<List<ResultData>> response) {

                if(response.isSuccessful()  &&
response.body() != null){

                    onGetResult(response.body());

                }else{

                    String errorCode;

                    switch (response.code()){

                        case 404:

                            errorCode = "404 not found";

                            break;

                        case 500:

                            errorCode = "500 server broken";

                            break;

                        default:

                            errorCode = "Unknown error";

                            break;

                    }

                    showToast("No Result\nPlease Try Again!-
"+ errorCode, R.layout.toast_gagal);

                }

            }



            @Override

            public  void  onFailure(@NonNull
Call<List<ResultData>> call, @NonNull Throwable t)
{

                showToast(t.toString(), R.layout.toast_gagal);

            }

        });

    }



    private String formatRupiah(int number) {

        DecimalFormat  formatter  =  new
DecimalFormat("#,###,###,###");

        return "Rp" + formatter.format(number);

    }



    private void showToast(String message, int layoutId)
{
```

```
        final Toast toast = new Toast(getApplicationContext());

        toast.setDuration(Toast.LENGTH_LONG);

        @SuppressLint("InflateParams") View custom_view = getLayoutInflater().inflate(layoutId, null);

        TextView tvMessage = custom_view.findViewById(R.id.message);

        tvMessage.setText(message);

        toast.setView(custom_view);

        toast.setGravity(Gravity.CENTER, 0, 0);

        toast.show();

    }


    @SuppressLint("NotifyDataSetChanged")

    private void onGetResult(List<ResultData> values) {

        adapter = new RincianPesananAdapter(this, values);

        adapter.notifyDataSetChanged();

        recyclerView.setAdapter(adapter);

        resultDataList = values;

    }

}
```

Script detail produk activity

```
package com.dilla.a220280030.woparepare.activity;



import static com.dilla.a220280030.woparepare.api.ApiClient.BASE_URL;



import android.annotation.SuppressLint;

import android.app.ProgressDialog;

import android.content.Intent;

import android.graphics.Color;

import android.os.Bundle;

import android.os.Handler;

import android.text.Html;

import android.view.Gravity;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.widget.LinearLayout;

import android.widget.TextView;

import android.widget.Toast;



import androidx.annotation.NonNull;

import androidx.appcompat.app.ActionBar;

import androidx.appcompat.app.AppCompatActivity;

import androidx.appcompat.widget.Toolbar;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import androidx.viewpager2.widget.ViewPager2;



import com.dilla.a220280030.woparepare.R;

import com.dilla.a220280030.woparepare.SessionManager;

import com.dilla.a220280030.woparepare.adapter.DetailProdukAdapter;

import com.dilla.a220280030.woparepare.adapter.MediaPagerAdapter;

import com.dilla.a220280030.woparepare.api.ApiClient;

import com.dilla.a220280030.woparepare.api.ApiInterface;

import com.dilla.a220280030.woparepare.model.Data;

import com.dilla.a220280030.woparepare.model.ResultData;



import org.jetbrains.annotations.NotNull;



import java.text.DecimalFormat;

import java.util.ArrayList;
```

```
import java.util.List;

import java.util.Objects;


import retrofit2.Call;

import retrofit2.Callback;

import retrofit2.Response;


public class DetailProdukActivity extends AppCompatActivity implements View.OnClickListener {


    private TextView tvharga, tvnama, tvkete, textCartItemCount;

    private ProgressDialog progress;

    private LinearLayout tambahkeranjang, belisekarng;

    String email, nohp, idkategori, nmkategori, idproduk, namaproduk, ketproduk, hargaproduk, gambarproduk, videoproduk;

    ApiInterface apiInterface;

    SessionManager sessionManager;

    private RecyclerView recyclerView;

    DetailProdukAdapter adapter;

    List<ResultData> resultDataList;

    private ViewPager2 viewPager;

    private MediaPagerAdapter mediaPagerAdapter;


    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_detail_produk);


        sessionManager = new SessionManager(DetailProdukActivity.this);

        email = sessionManager.getUserDetail().get(SessionManager.EMAIL);

        nohp = sessionManager.getUserDetail().get(SessionManager.NOHP);


        Intent get = getIntent();

        idkategori = get.getStringExtra("idkategori");

        nmkategori = get.getStringExtra("nmkategori");

        idproduk = get.getStringExtra("idproduk");

        namaproduk = get.getStringExtra("namaproduk");

        ketproduk = get.getStringExtra("ketproduk");

        hargaproduk = get.getStringExtra("hargaproduk");

        gambarproduk = get.getStringExtra("gambarproduk");

        videoproduk = get.getStringExtra("videoproduk");


        progress = new ProgressDialog(DetailProdukActivity.this);

        progress.setCancelable(false);

        progress.setMessage("Tunggu Sebentar...");


        tvharga = findViewById(R.id.tvharga);

        tvnama = findViewById(R.id.tvnama);

        tvkete = findViewById(R.id.tvkete);

        tambahkeranjang = findViewById(R.id.tambahkeranjang);

        belisekarng = findViewById(R.id.belisekarang);

        viewPager = findViewById(R.id.viewPager);


        resultDataList = new ArrayList<>();

        resultDataList.add(new ResultData(null, null, null, null, null, null, null, null, videoproduk, true));

        resultDataList.add(new ResultData(null, null, null, null, null, null, null, null, BASE_URL + "galeri/" + gambarproduk, null, false));

        mediaPagerAdapter = new MediaPagerAdapter(this, resultDataList);

        viewPager.setAdapter(mediaPagerAdapter);
```

```
        String hargaFormatted = formatRupiah(Integer.parseInt(hargaproduk));

        tvharga.setText(hargaFormatted);

        tvnama.setText(namaproduk);

        if (ketproduk.isEmpty()){

            tvkete.setVisibility(View.GONE);

        } else {

            tvkete.setVisibility(View.VISIBLE);

            tvkete.setText(Html.fromHtml(ketproduk));

        }



        recyclerView = findViewById(R.id.listviewulasan);

        recyclerView.setLayoutManager(new LinearLayoutManager(this));

        recyclerView.setHasFixedSize(true);



        ambildataulasan(idproduk);



        tambahkeranjang.setOnClickListener(this);

        belisekarng.setOnClickListener(this);



        Toolbar toolbar = findViewById(R.id.toolbar);

        setSupportActionBar(toolbar);

        ActionBar mAction = getSupportActionBar();

        assert mAction != null;

        TextView titleTextView = new TextView(getApplicationContext());

        titleTextView.setText("Detail Produk".toUpperCase());

        titleTextView.setTextSize(17);

        titleTextView.setTextColor(Color.WHITE);

        titleTextView.setGravity(Gravity.CENTER);

        TextView subtitleTextView = new TextView(getApplicationContext());

        subtitleTextView.setText(nmkategori);

        subtitleTextView.setTextSize(13);

        subtitleTextView.setTextColor(Color.WHITE);

        subtitleTextView.setGravity(Gravity.CENTER);

        LinearLayout titleLayout = new LinearLayout(getApplicationContext());


titleLayout.setOrientation(LinearLayout.VERTICAL);

        LinearLayout.LayoutParams params = new LinearLayout.LayoutParams(LinearLayout.LayoutParams.WRAP_CONTENT,
LinearLayout.LayoutParams.WRAP_CONTENT);

        params.gravity = Gravity.CENTER;

        titleLayout.addView(titleTextView, params);

        titleLayout.addView(subtitleTextView, params);


mAction.setDisplayOptions(ActionBar.DISPLAY_SHOW_CUSTOM);

        mAction.setCustomView(titleLayout, new ActionBar.LayoutParams(ActionBar.LayoutParams.WRAP_CONTENT,
ActionBar.LayoutParams.WRAP_CONTENT,
Gravity.CENTER));

        mAction.setDisplayHomeAsUpEnabled(true);

        toolbar.setNavigationOnClickListener(v -> onBackPressed());

    }



    @Override

    public void onClick(View v) {

        if (v.getId() == R.id.tambahkeranjang) {

            tambahkeranjang(email, nohp, idkategori, idproduk);

        } else if (v.getId() == R.id.belisekarang) {

            Intent it = new Intent(DetailProdukActivity.this, CheckoutActivity.class);

            it.putExtra("IDKATEGORI", idkategori);

            it.putExtra("IDPRODUK", idproduk);

            it.putExtra("NAMAKATEGORI", nmkategori);

            it.putExtra("NAMAPRODUK", namaproduk);

            it.putExtra("HARGA", hargaproduk);

            it.putExtra("JUMLAH", "1");

            it.putExtra("GAMBAR", gambarproduk);
```

```
        it.putExtra("TOTAL_JUMLAH", 1);

        it.putExtra("TOTAL_HARGA",
Integer.parseInt(hargaproduk));

        startActivity(it);

      }

   }


   @Override

   public boolean onCreateOptionsMenu(Menu menu) {

      getMenuInflater().inflate(R.menu.data_diri,
menu);

      final MenuItem menuItem =
menu.findItem(R.id.keranjang);

      View actionView = menuItem.getActionView();

      textCartItemCount =
actionView.findViewById(R.id.cart_badge);

      badgekeranjang(email);

      Handler handler = new Handler();

      handler.postDelayed(new Runnable() {

         @Override

         public void run() {

            badgekeranjang(email);

            handler.postDelayed(this, 1000);

         }

      }, 1000);

      actionView.setOnClickListener(v ->
onOptionsItemSelected(menuItem));

      MenuItem listmenu =
menu.findItem(R.id.listmenu);

      listmenu.setVisible(false);

      return true;

   }


   @SuppressLint("SetTextI18n")

   @Override

   public boolean onOptionsItemSelected(MenuItem
item) {

      if (item.getItemId() == R.id.data_diri) {

         Intent it = new Intent(DetailProdukActivity.this,
ProfilActivity.class);

         startActivity(it);

      } else if (item.getItemId() == R.id.keranjang) {

         startActivity(new
Intent(DetailProdukActivity.this,
KeranjangActivity.class));

      } else if (item.getItemId() == android.R.id.home) {

         onBackPressed();

      }

      return super.onOptionsItemSelected(item);

   }


   private void setupBadge(Integer mCartItemCount) {

      if (textCartItemCount != null) {

         if (mCartItemCount == 0) {

            if (textCartItemCount.getVisibility() !=
View.GONE) {


textCartItemCount.setVisibility(View.GONE);

            }

         } else {

            int maxLength = 2;

            String itemCountText =
String.valueOf(Math.min(mCartItemCount, 99));

            if (itemCountText.length() > maxLength) {

               itemCountText =
itemCountText.substring(0, maxLength);

            }

            if (mCartItemCount > 99) {

               itemCountText += "+";

            }

            textCartItemCount.setText(itemCountText);

            textCartItemCount.setPadding(0, 0, 0, 0);

            if (textCartItemCount.getVisibility() !=
View.VISIBLE) {
```

```
textCartItemCount.setVisibility(View.VISIBLE);
            }
        }
    }
}

private void tambahkeranjang(String email, String nohp, String idkate, String idproduk) {
    progress.show();
    apiInterface = ApiClient.getApiClient().create(ApiInterface.class);
    Call<Data> masukCall = apiInterface.tambahkeranjangRespone(idkate, idproduk, email, nohp);
    masukCall.enqueue(new Callback<Data>() {
        @Override
        public void onResponse(@NotNull Call<Data> call, @NotNull Response<Data> response) {
            progress.hide();
            if(response.body() != null && response.isSuccessful() && response.body().isStatus()) {
                showToast(response.body().getMessage(), R.layout.toast_berhasil);
            }else{
showToast(Objects.requireNonNull(response.body()).getMessage(), R.layout.toast_gagal);
            }
        }

        @SuppressLint("SetTextI18n")
        @Override
        public void onFailure(@NotNull Call<Data> call, @NotNull Throwable t) {
            progress.hide();
            showToast("Tidak ada koneksi internet, silahkan coba lagi!", R.layout.toast_gagal);
        }
    });
}

private void badgekeranjang(String email){
    apiInterface = ApiClient.getApiClient().create(ApiInterface.class);
    Call<List<ResultData>> call = apiInterface.getCountKeranjang(email);
    call.enqueue(new Callback<List<ResultData>>() {

        @Override
        public void onResponse(@NonNull Call<List<ResultData>> call, @NonNull Response<List<ResultData>> response) {
            if(response.isSuccessful() && response.body() != null){
                int jml = response.body().get(0).getIsiKeranjang();
                setupBadge(jml);
            }else{
                showToast("Gagal meload data", R.layout.toast_gagal);
            }
        }

        @Override
        public void onFailure(@NonNull Call<List<ResultData>> call, @NonNull Throwable t) {
            showToast(t.toString(), R.layout.toast_gagal);
        }
    });
}

private String formatRupiah(int number) {
    DecimalFormat formatter = new DecimalFormat("#,###,###,###");
    return "Rp" + formatter.format(number);
}
```

```
    private void showToast(String message, int layoutId)
{

        final Toast toast = new
Toast(getApplicationContext());

        toast.setDuration(Toast.LENGTH_LONG);

        @SuppressLint("InflateParams") View
custom_view = getLayoutInflater().inflate(layoutId,
null);

        TextView tvMessage =
custom_view.findViewById(R.id.message);

        tvMessage.setText(message);

        toast.setView(custom_view);

        toast.setGravity(Gravity.CENTER, 0, 0);

        toast.show();

    }



    private void ambildataulasan(String idproduk){

        apiInterface =
ApiClient.getApiClient().create(ApiInterface.class);

        Call<List<ResultData>> call =
apiInterface.getUlasanProduk(idproduk);

        call.enqueue(new Callback<List<ResultData>>() {



            @Override

            public void onResponse(@NonNull
Call<List<ResultData>> call, @NonNull
Response<List<ResultData>> response) {

                if(response.isSuccessful() &&
response.body() != null){

                    onGetResult(response.body());

                } else {

                    String errorCode;

                    switch (response.code()){

                        case 404:

                            errorCode = "404 not found";

                            break;

                        case 500:

                            errorCode = "500 server broken";

                            break;

                        default:

                            errorCode = "Unknown error";

                            break;

                    }

                    showToast("No Result\nPlease Try Again!-
"+ errorCode, R.layout.toast_gagal);

                }

            }



            @Override

            public void onFailure(@NonNull
Call<List<ResultData>> call, @NonNull Throwable t)
{

                showToast(t.toString(), R.layout.toast_gagal);

            }

        });

    }



    @SuppressLint("NotifyDataSetChanged")

    private void onGetResult(List<ResultData> values) {

        adapter = new DetailProdukAdapter(this, values);

        adapter.notifyDataSetChanged();

        recyclerView.setAdapter(adapter);

        resultDataList = values;

    }

}
```

Script transaksi pager

```
package com.dilla.a220280030.woparepare.activity;


import androidx.annotation.NonNull;

import androidx.fragment.app.Fragment;

import androidx.fragment.app.FragmentManager;

import
androidx.fragment.app.FragmentStatePagerAdapter;
```

```
import com.dilla.a220280030.woparepare.fragment.BelumBayarFragment;

import com.dilla.a220280030.woparepare.fragment.DalamProsesFragment;

import com.dilla.a220280030.woparepare.fragment.SelesaiFragment;



public class TransaksiPager extends FragmentStatePagerAdapter {

    private int tabCount;



    public TransaksiPager(@NonNull FragmentManager fm, int tabCount) {

        super(fm, BEHAVIOR_RESUME_ONLY_CURRENT_FRAGMENT);

        this.tabCount = tabCount;

    }



    @NonNull

    @Override

    public Fragment getItem(int position) {

        switch (position) {

            case 0:

                return new BelumBayarFragment();

            case 1:

                return new DalamProsesFragment();

            case 2:

                return new SelesaiFragment();

            default:

                return null;

        }

    }



    @Override

    public int getCount() {

        return tabCount;

    }

}
```

Script tranksaksi activity

```
package com.dilla.a220280030.woparepare.activity;



import android.annotation.SuppressLint;

import android.content.Intent;

import android.graphics.Color;

import android.os.Bundle;

import android.os.Handler;

import android.view.Gravity;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.widget.TextView;

import android.widget.Toast;



import androidx.annotation.NonNull;

import androidx.appcompat.app.ActionBar;

import androidx.appcompat.app.AppCompatActivity;

import androidx.appcompat.widget.Toolbar;

import androidx.viewpager.widget.ViewPager;



import com.dilla.a220280030.woparepare.R;

import com.dilla.a220280030.woparepare.SessionManager;

import com.dilla.a220280030.woparepare.api.ApiClient;

import com.dilla.a220280030.woparepare.api.ApiInterface;

import com.dilla.a220280030.woparepare.model.ResultData;

import com.google.android.material.tabs.TabLayout;



import java.util.List;
```

```
import retrofit2.Call;

import retrofit2.Callback;

import retrofit2.Response;


public class TransaksiActivity extends AppCompatActivity {


    private ViewPager viewPager;

    private TextView textCartItemCount;

    String email;

    ApiInterface apiInterface;

    SessionManager sessionManager;


    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_transaksi);


        sessionManager = new SessionManager(TransaksiActivity.this);

        email = sessionManager.getUserDetail().get(SessionManager.EMAIL);


        TabLayout tabLayout = findViewById(R.id.tabLayout);


tabLayout.addTab(tabLayout.newTab().setText("Belum Bayar"));


tabLayout.addTab(tabLayout.newTab().setText("Dalam Proses"));


tabLayout.addTab(tabLayout.newTab().setText("Selesai"));


        viewPager = findViewById(R.id.pager);


        TransaksiPager adapter = new TransaksiPager(getSupportFragmentManager(), tabLayout.getTabCount());

        viewPager.setAdapter(adapter);

        viewPager.addOnPageChangeListener(new TabLayout.TabLayoutOnPageChangeListener(tabLayout));

        tabLayout.addOnTabSelectedListener(new TabLayout.OnTabSelectedListener() {

            @Override

            public void onTabSelected(@NonNull TabLayout.Tab tab) {

                viewPager.setCurrentItem(tab.getPosition());

            }


            @Override

            public void onTabUnselected(@NonNull TabLayout.Tab tab) {


            }


            @Override

            public void onTabReselected(@NonNull TabLayout.Tab tab) {


            }

        });

        int selectedTab = getIntent().getIntExtra("SELECTED_TAB", 0);

        viewPager.setCurrentItem(selectedTab);


        Toolbar toolbar = findViewById(R.id.toolbar);

        setSupportActionBar(toolbar);

        ActionBar mAction = getSupportActionBar();

        assert mAction != null;

        TextView titleTextView = new TextView(getApplicationContext());

        titleTextView.setText("Transaksi".toUpperCase());

        titleTextView.setTextSize(17);
```

```
        titleTextView.setTextColor(Color.WHITE);

        titleTextView.setGravity(Gravity.CENTER);


mAction.setDisplayOptions(ActionBar.DISPLAY_SHOW_CUSTOM);

        mAction.setCustomView(titleTextView, new ActionBar.LayoutParams(ActionBar.LayoutParams.WRAP_CONTENT,
ActionBar.LayoutParams.WRAP_CONTENT,
Gravity.CENTER));

        mAction.setDisplayHomeAsUpEnabled(true);

        toolbar.setNavigationOnClickListener(v -> onBackPressed());

    }



    @Override

    public boolean onCreateOptionsMenu(Menu menu) {

        getMenuInflater().inflate(R.menu.data_diri, menu);

        final MenuItem menuItem = menu.findItem(R.id.keranjang);

        View actionView = menuItem.getActionView();

        textCartItemCount = actionView.findViewById(R.id.cart_badge);

        badgekeranjang(email);

        Handler handler = new Handler();

        handler.postDelayed(new Runnable() {

            @Override

            public void run() {

                badgekeranjang(email);

                handler.postDelayed(this, 1000);

            }

        }, 1000);

        actionView.setOnClickListener(v -> onOptionsItemSelected(menuItem));

        MenuItem datadiri = menu.findItem(R.id.data_diri);

        datadiri.setVisible(false);

        MenuItem listmenu = menu.findItem(R.id.listmenu);

        listmenu.setVisible(false);

        return true;

    }



    @SuppressLint("SetTextI18n")

    @Override

    public boolean onOptionsItemSelected(MenuItem item) {

        if (item.getItemId() == R.id.data_diri) {

            Intent it = new Intent(TransaksiActivity.this, ProfilActivity.class);

            startActivity(it);

        } else if (item.getItemId() == R.id.keranjang) {

            startActivity(new Intent(TransaksiActivity.this, KeranjangActivity.class));

        } else if (item.getItemId() == android.R.id.home) {

            onBackPressed();

        }

        return super.onOptionsItemSelected(item);

    }



    private void badgekeranjang(String email){

        apiInterface = ApiClient.getApiClient().create(ApiInterface.class);

        Call<List<ResultData>> call = apiInterface.getCountKeranjang(email);

        call.enqueue(new Callback<List<ResultData>>() {



            @Override

            public void onResponse(@NonNull Call<List<ResultData>> call, @NonNull Response<List<ResultData>> response) {

                if(response.isSuccessful() && response.body() != null){

                    int jml = response.body().get(0).getIsiKeranjang();

                    setupBadge(jml);

                }else{
```

```
            showToast("Gagal meload data", R.layout.toast_gagal);
            }
        }

        @Override
        public void onFailure(@NonNull Call<List<ResultData>> call, @NonNull Throwable t) {
            showToast(t.toString(), R.layout.toast_gagal);
        }
    });
  }

  private void showToast(String message, int layoutId) {
    final Toast toast = new Toast(getApplicationContext());
    toast.setDuration(Toast.LENGTH_LONG);
    @SuppressLint("InflateParams") View custom_view = getLayoutInflater().inflate(layoutId, null);
    TextView tvMessage = custom_view.findViewById(R.id.message);
    tvMessage.setText(message);
    toast.setView(custom_view);
    toast.setGravity(Gravity.CENTER, 0, 0);
    toast.show();
  }

  private void setupBadge(Integer mCartItemCount) {
    if (textCartItemCount != null) {
      if (mCartItemCount == 0) {
        if (textCartItemCount.getVisibility() != View.GONE) {
          textCartItemCount.setVisibility(View.GONE);
        }
      } else {
        int maxLength = 2;
        String itemCountText = String.valueOf(Math.min(mCartItemCount, 99));
        if (itemCountText.length() > maxLength) {
          itemCountText = itemCountText.substring(0, maxLength);
        }
        if (mCartItemCount > 99) {
          itemCountText += "+";
        }
        textCartItemCount.setText(itemCountText);
        textCartItemCount.setPadding(0, 0, 0, 0);
        if (textCartItemCount.getVisibility() != View.VISIBLE) {
          textCartItemCount.setVisibility(View.VISIBLE);
        }
      }
    }
  }
}
```

Script profil activity

```
package com.dilla.a220280030.woparepare.activity;

import android.annotation.SuppressLint;
import android.app.ProgressDialog;
import android.content.Intent;
import android.graphics.Color;
import android.graphics.Paint;
import android.os.Bundle;
import android.os.Handler;
import android.text.Editable;
import android.text.TextWatcher;
import android.text.method.HideReturnsTransformationMethod;
```

```
import
android.text.method.PasswordTransformationMethod;

import android.view.Gravity;

import android.view.LayoutInflater;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.LinearLayout;

import android.widget.TextView;

import android.widget.Toast;



import androidx.annotation.NonNull;

import androidx.appcompat.app.ActionBar;

import androidx.appcompat.app.AlertDialog;

import androidx.appcompat.app.AppCompatActivity;

import androidx.appcompat.widget.Toolbar;



import com.dilla.a220280030.woparepare.R;

import
com.dilla.a220280030.woparepare.SessionManager;

import
com.dilla.a220280030.woparepare.api.ApiClient;

import
com.dilla.a220280030.woparepare.api.ApiInterface;

import com.dilla.a220280030.woparepare.model.Data;

import
com.dilla.a220280030.woparepare.model.ResultData;

import
com.google.android.material.dialog.MaterialAlertDialo
gBuilder;

import
com.google.android.material.textfield.TextInputEditTex
t;

import
com.google.android.material.textfield.TextInputLayout;



import org.jetbrains.annotations.NotNull;

import java.util.List;

import java.util.Objects;



import retrofit2.Call;

import retrofit2.Callback;

import retrofit2.Response;



public class ProfilActivity extends AppCompatActivity
implements View.OnClickListener {



    private TextView tvnama, tvemail, tvnohp, tvalamat,
tvubahpass, tvkeluar, tvjmlbelumbayar,
tvjmldalamproses, tvjmlselesai, textCartItemCount;

    private Button btnubah;

    private ProgressDialog progress;

    private LinearLayout linearalamat, belumbayar,
dalamproses, selesai;

    private View viewalamat;

    String email, nohp, nama, alamat;

    ApiInterface apiInterface;

    SessionManager sessionManager;



    @Override

    protected void onCreate(Bundle savedInstanceState)
{

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_profil);



        sessionManager = new
SessionManager(ProfilActivity.this);

        email =
sessionManager.getUserDetail().get(SessionManager.E
MAIL);

        nohp =
sessionManager.getUserDetail().get(SessionManager.N
OHP);

        nama =
sessionManager.getUserDetail().get(SessionManager.N
AMA);
```

```
        alamat = sessionManager.getUserDetail().get(SessionManager.ALAMAT);


        progress = new ProgressDialog(ProfilActivity.this);
        progress.setCancelable(false);
        progress.setMessage("Mohon Tunggu...");


        tvjmlbelumbayar = findViewById(R.id.tvjmlbelumbayar);
        badgebelumbayar(email);
        belumbayar = findViewById(R.id.belumbayar);
        belumbayar.setOnClickListener(this);
        tvjmldalamproses = findViewById(R.id.tvjmldalamproses);
        badgedalamproses(email);
        dalamproses = findViewById(R.id.dalamproses);
        dalamproses.setOnClickListener(this);
        tvjmlselesai = findViewById(R.id.tvjmlselesai);
        badgeselesai(email);
        selesai = findViewById(R.id.selesai);
        selesai.setOnClickListener(this);
        tvnama = findViewById(R.id.tvnama);
        tvemail = findViewById(R.id.tvemail);
        tvnohp = findViewById(R.id.tvnohp);
        linearalamat = findViewById(R.id.linearalamat);
        viewalamat = findViewById(R.id.viewalamat);
        tvalamat = findViewById(R.id.tvalamat);
        updateProfileInfo(nama, email, nohp, alamat);
        btnubah = findViewById(R.id.ubahdata);
        btnubah.setOnClickListener(this);
        tvubahpass = findViewById(R.id.tvubahpass);

tvubahpass.setPaintFlags(Paint.UNDERLINE_TEXT_FLAG);
        tvubahpass.setOnClickListener(this);
        tvkeluar = findViewById(R.id.keluar);

tvkeluar.setPaintFlags(Paint.UNDERLINE_TEXT_FLAG);
        tvkeluar.setOnClickListener(this);


        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        ActionBar mAction = getSupportActionBar();
        assert mAction != null;
        TextView titleTextView = new TextView(getApplicationContext());
        titleTextView.setText("Profil".toUpperCase());
        titleTextView.setTextSize(17);
        titleTextView.setTextColor(Color.WHITE);
        titleTextView.setGravity(Gravity.CENTER);

mAction.setDisplayOptions(ActionBar.DISPLAY_SHOW_CUSTOM);
        mAction.setCustomView(titleTextView, new ActionBar.LayoutParams(ActionBar.LayoutParams.WRAP_CONTENT,
ActionBar.LayoutParams.WRAP_CONTENT,
Gravity.CENTER));
        mAction.setDisplayHomeAsUpEnabled(true);
        toolbar.setNavigationOnClickListener(v -> onBackPressed());
    }


    private void updateProfileInfo(String newName, String newEmail, String newNoHp, String newAlamat) {
        Handler handler = new Handler();
        handler.post(() -> {
            tvnama.setText(newName);
            tvemail.setText(newEmail);
            tvnohp.setText(newNoHp);
            if (newAlamat.isEmpty()){
                linearalamat.setVisibility(View.GONE);
                viewalamat.setVisibility(View.GONE);
```

```
            tvalamat.setText("");
        } else {
            linearalamat.setVisibility(View.VISIBLE);
            viewalamat.setVisibility(View.VISIBLE);
            tvalamat.setText(newAlamat);
        }
    });
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.data_diri, menu);
    final MenuItem menuItem = menu.findItem(R.id.keranjang);
    View actionView = menuItem.getActionView();
    textCartItemCount = actionView.findViewById(R.id.cart_badge);
    badgekeranjang(email);
    Handler handler = new Handler();
    handler.postDelayed(new Runnable() {
        @Override
        public void run() {
            badgekeranjang(email);
            handler.postDelayed(this, 1000);
        }
    }, 1000);
    actionView.setOnClickListener(v -> onOptionsItemSelected(menuItem));
    MenuItem datadiri = menu.findItem(R.id.data_diri);
    datadiri.setVisible(false);
    MenuItem listmenu = menu.findItem(R.id.listmenu);
    listmenu.setVisible(false);
    return true;
}

@SuppressLint("SetTextI18n")
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    if (item.getItemId() == R.id.keranjang){
        startActivity(new Intent(ProfilActivity.this, KeranjangActivity.class));
    } else if (item.getItemId() == android.R.id.home) {
        onBackPressed();
    }
    return super.onOptionsItemSelected(item);
}

@Override
public void onClick(View v) {
    if (v.getId() == R.id.ubahdata) {
        formubahdata();
    } else if (v.getId() == R.id.tvubahpass) {
        formubaspass();
    } else if (v.getId() == R.id.belumbayar) {
        Intent it = new Intent(ProfilActivity.this, TransaksiActivity.class);
        it.putExtra("SELECTED_TAB", 0);
        startActivity(it);
    } else if (v.getId() == R.id.dalamproses) {
        Intent it = new Intent(ProfilActivity.this, TransaksiActivity.class);
        it.putExtra("SELECTED_TAB", 1);
        startActivity(it);
    } else if (v.getId() == R.id.selesai) {
        Intent it = new Intent(ProfilActivity.this, TransaksiActivity.class);
        it.putExtra("SELECTED_TAB", 2);
        startActivity(it);
    } else if(v.getId() == R.id.keluar) {
```

```
        new AlertDialog.Builder(this)
                .setTitle("Konfirmasi Keluar!")
                .setMessage("Apakah Anda yakin ingin keluar?")
                .setPositiveButton("Ya", (dialog, which) -> {
                    sessionManager.logoutSession();
                    Intent intent = new Intent(ProfilActivity.this, MasukActivity.class);
intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
                    startActivity(intent);
                    finish();
                })
                .setNegativeButton("Tidak", null)
                .setIcon(R.drawable.ic_info)
                .show();
    }
  }

  private void formubahdata() {
    LayoutInflater inflater = getLayoutInflater();
    View customView = inflater.inflate(R.layout.item_dialog_ubahdata, null);
    TextView juduldialog = customView.findViewById(R.id.txtdialjudul);
    TextInputLayout tvnama = customView.findViewById(R.id.tvnama);
    TextInputLayout tvemail = customView.findViewById(R.id.tvemail);
    TextInputLayout tvnohp = customView.findViewById(R.id.tvnohp);
    TextInputLayout tvalamat = customView.findViewById(R.id.tvalamat);
    TextInputEditText edtnama = customView.findViewById(R.id.edtnama);
    TextInputEditText edtemail = customView.findViewById(R.id.edtemail);
    TextInputEditText edtnohp = customView.findViewById(R.id.edtnohp);
    TextInputEditText edtalamat = customView.findViewById(R.id.edtalamat);
    Button btnsimpan = customView.findViewById(R.id.btnsimpan);
    juduldialog.setText("Ubah Profil");
juduldialog.setTextAlignment(View.TEXT_ALIGNMENT_CENTER);
juduldialog.setTextColor(Color.parseColor("#FFFFFF"));
juduldialog.setBackgroundColor(Color.parseColor("#3C5B6F"));
    AlertDialog dialog = new MaterialAlertDialogBuilder(ProfilActivity.this, R.style.RoundedMaterialDialog).setView(customView).show();
    edtnama.addTextChangedListener(new TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence s, int start, int count, int after) {

        }

        @Override
        public void onTextChanged(CharSequence s, int start, int before, int count) {
            validnama(edtnama,tvnama);
        }

        @Override
        public void afterTextChanged(Editable s) {

        }
    });
    edtemail.addTextChangedListener(new TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence s, int start, int count, int after) {
```

```
        }


        @Override

        public void onTextChanged(CharSequence s, int
start, int before, int count) {

            validemail(edtemail,tvemail);

        }


        @Override

        public void afterTextChanged(Editable s) {


        }

    });

    edtnohp.addTextChangedListener(new
TextWatcher() {

        @Override

        public void beforeTextChanged(CharSequence
s, int start, int count, int after) {


        }


        @Override

        public void onTextChanged(CharSequence s, int
start, int before, int count) {

            validnohp(edtnohp,tvnohp);

        }


        @Override

        public void afterTextChanged(Editable s) {


        }

    });

    edtalamat.addTextChangedListener(new
TextWatcher() {

        @Override

        public void beforeTextChanged(CharSequence
s, int start, int count, int after) {


        }


        @Override

        public void onTextChanged(CharSequence s, int
start, int before, int count) {

            validalamat(edtalamat,tvalamat);

        }


        @Override

        public void afterTextChanged(Editable s) {


        }

    });

    btnsimpan.setOnClickListener(v1 -> {

        if (!validnama(edtnama,tvnama) |
!validemail(edtemail,tvemail) |
!validnohp(edtnohp,tvnohp) |
!validalamat(edtalamat,tvalamat)) {

            return;

        }


ubahdata(Objects.requireNonNull(edtemail.getText()).t
oString(),
Objects.requireNonNull(edtnohp.getText()).toString(),
Objects.requireNonNull(edtnama.getText()).toString(),
Objects.requireNonNull(edtalamat.getText()).toString(),
email, nohp);

        dialog.dismiss();

    });

    dialog.setCancelable(false);


dialog.findViewById(R.id.tutupdialog).setOnClickListe
ner(v2 -> dialog.dismiss());

}


private void formubaspass() {

    LayoutInflater inflater = getLayoutInflater();
```

```
        View customView = inflater.inflate(R.layout.item_dialog_ubahpass, null);

        TextView juduldialog = customView.findViewById(R.id.txtdialjudul);

        TextInputLayout tvpasslama = customView.findViewById(R.id.tvpasslama);

        TextInputLayout tvpassbaru = customView.findViewById(R.id.tvpassbaru);

        TextInputLayout tvulangpassbaru = customView.findViewById(R.id.tvulangpassbaru);

        TextInputEditText edtpasslama = customView.findViewById(R.id.edtpasslama);

        TextInputEditText edtpassbaru = customView.findViewById(R.id.edtpassbaru);

        TextInputEditText edtulangpassbaru = customView.findViewById(R.id.edtulangpassbaru);

        ImageView imgpasslama = customView.findViewById(R.id.showpasslama);

        ImageView imgpassbaru = customView.findViewById(R.id.showpassbaru);

        ImageView imgulangpassbaru = customView.findViewById(R.id.showulangpassbaru);

        Button btnsimpan = customView.findViewById(R.id.btnsimpan);

        juduldialog.setText("Ubah Password");


juduldialog.setTextAlignment(View.TEXT_ALIGNMENT_CENTER);


juduldialog.setTextColor(Color.parseColor("#FFFFFF"));


juduldialog.setBackgroundColor(Color.parseColor("#3C5B6F"));

        AlertDialog dialog = new MaterialAlertDialogBuilder(ProfilActivity.this, R.style.RoundedMaterialDialog).setView(customView).show();

        imgpasslama.setOnClickListener(v1 -> {

            if(v1.getId()==R.id.showpasslama){

                if (Objects.equals(edtpasslama.getTransformationMethod(), PasswordTransformationMethod.getInstance())) {


imgpasslama.setImageResource(R.drawable.hide_password);


edtpasslama.setTransformationMethod(HideReturnsTransformationMethod.getInstance());

                } else {


imgpasslama.setImageResource(R.drawable.show_password);


edtpasslama.setTransformationMethod(PasswordTransformationMethod.getInstance());

                }

            }

        });

        imgpassbaru.setOnClickListener(v2 -> {

            if(v2.getId()==R.id.showpassbaru){

                if (Objects.equals(edtpassbaru.getTransformationMethod(), PasswordTransformationMethod.getInstance())) {


imgpassbaru.setImageResource(R.drawable.hide_password);


edtpassbaru.setTransformationMethod(HideReturnsTransformationMethod.getInstance());

                } else {


imgpassbaru.setImageResource(R.drawable.show_password);


edtpassbaru.setTransformationMethod(PasswordTransformationMethod.getInstance());

                }

            }

        });

        imgulangpassbaru.setOnClickListener(v3 -> {

            if(v3.getId()==R.id.showulangpassbaru){


if(Objects.equals(edtulangpassbaru.getTransformationMethod(), PasswordTransformationMethod.getInstance())){


imgulangpassbaru.setImageResource(R.drawable.hide_password);


edtulangpassbaru.setTransformationMethod(HideReturnsTransformationMethod.getInstance());
```

```
            } else{

imgulangpassbaru.setImageResource(R.drawable.show
_password);

edtulangpassbaru.setTransformationMethod(PasswordT
ransformationMethod.getInstance());

            }

        }

    });

    edtpasslama.addTextChangedListener(new
TextWatcher() {

        @Override

        public void beforeTextChanged(CharSequence
s, int start, int count, int after) {



        }



        @Override

        public void onTextChanged(CharSequence s, int
start, int before, int count) {

            validpasslama(edtpasslama,tvpasslama);

        }



        @Override

        public void afterTextChanged(Editable s) {



        }

    });

    edtpassbaru.addTextChangedListener(new
TextWatcher() {

        @Override

        public void beforeTextChanged(CharSequence
s, int start, int count, int after) {



        }



        @Override

        public void onTextChanged(CharSequence s, int
start, int before, int count) {

            validpassbaru(edtpassbaru,tvpassbaru);

        }



        @Override

        public void afterTextChanged(Editable s) {



        }

    });

    edtulangpassbaru.addTextChangedListener(new
TextWatcher() {

        @Override

        public void beforeTextChanged(CharSequence
s, int start, int count, int after) {



        }



        @Override

        public void onTextChanged(CharSequence s, int
start, int before, int count) {

valididulangpassbaru(edtulangpassbaru,tvulangpassbaru);

        }



        @Override

        public void afterTextChanged(Editable s) {



        }

    });

    btnsimpan.setOnClickListener(v4 -> {

        if (!validpasslama(edtpasslama,tvpasslama) |
!validpassbaru(edtpassbaru,tvpassbaru) |
!validulangpassbaru(edtulangpassbaru,tvulangpassbaru)
) {

            return;

        }
```

```
ubahpass(Objects.requireNonNull(edtpasslama.getText(
)).toString(),
Objects.requireNonNull(edtpassbaru.getText()).toString
(),
Objects.requireNonNull(edtulangpassbaru.getText()).to
String(), email);

            dialog.dismiss();

        });

        dialog.setCancelable(false);


dialog.findViewById(R.id.tutupdialog).setOnClickListe
ner(v5 -> dialog.dismiss());

    }



    private void ubahdata(String email, String nohp,
String nama, String alamat, String kdeml, String
kdnohp) {

        progress.show();

        apiInterface =
ApiClient.getApiClient().create(ApiInterface.class);

        Call<Data> masukCall =
apiInterface.ubahdataRespone(email, nohp, nama,
alamat, kdeml, kdnohp);

        masukCall.enqueue(new Callback<Data>() {

            @Override

            public void onResponse(@NotNull Call<Data>
call, @NotNull Response<Data> response) {

                progress.hide();

                if(response.body() != null &&
response.isSuccessful() && response.body().isStatus())
{

                    sessionManager = new
SessionManager(ProfilActivity.this);

                    sessionManager.createLoginSession(email,
nohp, nama, alamat);

                    showToast(response.body().getMessage(),
R.layout.toast_berhasil);

                    updateProfileInfo(nama, email, nohp,
alamat);

                }else{


showToast(Objects.requireNonNull(response.body()).g
etMessage(), R.layout.toast_berhasil);

                    formubahdata();

                }

            }



            @SuppressLint("SetTextI18n")

            @Override

            public void onFailure(@NotNull Call<Data>
call, @NotNull Throwable t) {

                progress.hide();

                showToast("Tidak ada koneksi internet,
silahkan coba lagi!", R.layout.toast_gagal);

            }

        });

    }



    private void ubahpass(String passlama, String
passbaru, String ulangpassbaru, String email) {

        progress.show();

        apiInterface =
ApiClient.getApiClient().create(ApiInterface.class);

        Call<Data> masukCall =
apiInterface.ubahpassRespone(passlama, passbaru,
ulangpassbaru, email);

        masukCall.enqueue(new Callback<Data>() {

            @Override

            public void onResponse(@NotNull Call<Data>
call, @NotNull Response<Data> response) {

                progress.hide();

                if(response.body() != null &&
response.isSuccessful() && response.body().isStatus())
{

                    showToast(response.body().getMessage(),
R.layout.toast_berhasil);

                }else{


showToast(Objects.requireNonNull(response.body()).g
etMessage(), R.layout.toast_gagal);

                    formubaspass();

                }

            }
```

```
        @SuppressLint("SetTextI18n")

        @Override

        public void onFailure(@NotNull Call<Data> call, @NotNull Throwable t) {

            progress.hide();

            showToast("Tidak ada koneksi internet, silahkan coba lagi!", R.layout.toast_gagal);

        }

    });

  }



  private void badgekeranjang(String email){

    apiInterface = ApiClient.getApiClient().create(ApiInterface.class);

    Call<List<ResultData>> call = apiInterface.getCountKeranjang(email);

    call.enqueue(new Callback<List<ResultData>>() {



        @Override

        public void onResponse(@NonNull Call<List<ResultData>> call, @NonNull Response<List<ResultData>> response) {

            if(response.isSuccessful() && response.body() != null){

                int jml = response.body().get(0).getIsiKeranjang();

                setupBadge(jml);

            }else{

                showToast("Gagal meload data", R.layout.toast_gagal);

            }

        }



        @Override

        public void onFailure(@NonNull Call<List<ResultData>> call, @NonNull Throwable t) {

            showToast(t.toString(), R.layout.toast_gagal);

        }

    });

  }



  private void badgebelumbayar(String email){

    apiInterface = ApiClient.getApiClient().create(ApiInterface.class);

    Call<List<ResultData>> call = apiInterface.getCountBelumBayar(email);

    call.enqueue(new Callback<List<ResultData>>() {



        @Override

        public void onResponse(@NonNull Call<List<ResultData>> call, @NonNull Response<List<ResultData>> response) {

            if(response.isSuccessful() && response.body() != null){

                int jml = response.body().get(0).getIsiTransaksi();


tvjmlbelumbayar.setText(String.valueOf(jml));

            }else{

                showToast("Gagal meload data", R.layout.toast_gagal);

            }

        }



        @Override

        public void onFailure(@NonNull Call<List<ResultData>> call, @NonNull Throwable t) {

            showToast(t.toString(), R.layout.toast_gagal);

        }

    });

  }



  private void badgedalamproses(String email){

    apiInterface = ApiClient.getApiClient().create(ApiInterface.class);

    Call<List<ResultData>> call = apiInterface.getCountDalamProses(email);
```

```
        call.enqueue(new Callback<List<ResultData>>() {



            @Override

            public void onResponse(@NonNull Call<List<ResultData>> call, @NonNull Response<List<ResultData>> response) {

                if(response.isSuccessful() && response.body() != null){

                    int jml = response.body().get(0).getIsiTransaksi();


tvjmldalamproses.setText(String.valueOf(jml));

                }else{

                    showToast("Gagal meload data", R.layout.toast_gagal);

                }

            }



            @Override

            public void onFailure(@NonNull Call<List<ResultData>> call, @NonNull Throwable t) {

                showToast(t.toString(), R.layout.toast_gagal);

            }

        });

    }



    private void badgeselesai(String email){

        apiInterface = ApiClient.getApiClient().create(ApiInterface.class);

        Call<List<ResultData>> call = apiInterface.getCountSelesai(email);

        call.enqueue(new Callback<List<ResultData>>() {



            @Override

            public void onResponse(@NonNull Call<List<ResultData>> call, @NonNull Response<List<ResultData>> response) {

                if(response.isSuccessful() && response.body() != null){

                    int jml = response.body().get(0).getIsiTransaksi();

                    tvjmlselesai.setText(String.valueOf(jml));

                }else{

                    showToast("Gagal meload data", R.layout.toast_gagal);

                }

            }



            @Override

            public void onFailure(@NonNull Call<List<ResultData>> call, @NonNull Throwable t) {

                showToast(t.toString(), R.layout.toast_gagal);

            }

        });

    }



    private void showToast(String message, int layoutId) {

        final Toast toast = new Toast(getApplicationContext());

        toast.setDuration(Toast.LENGTH_LONG);

        @SuppressLint("InflateParams") View custom_view = getLayoutInflater().inflate(layoutId, null);

        TextView tvMessage = custom_view.findViewById(R.id.message);

        tvMessage.setText(message);

        toast.setView(custom_view);

        toast.setGravity(Gravity.CENTER, 0, 0);

        toast.show();

    }



    private void setupBadge(Integer mCartItemCount) {

        if (textCartItemCount != null) {

            if (mCartItemCount == 0) {

                if (textCartItemCount.getVisibility() != View.GONE) {
```

```
textCartItemCount.setVisibility(View.GONE);

            }

        } else {

            int maxLength = 2;

            String itemCountText =
String.valueOf(Math.min(mCartItemCount, 99));

            if (itemCountText.length() > maxLength) {

                itemCountText =
itemCountText.substring(0, maxLength);

            }

            if (mCartItemCount > 99) {

                itemCountText += "+";

            }

            textCartItemCount.setText(itemCountText);

            textCartItemCount.setPadding(0, 0, 0, 0);

            if (textCartItemCount.getVisibility() !=
View.VISIBLE) {


textCartItemCount.setVisibility(View.VISIBLE);

            }

        }

    }


    private boolean validnama(TextInputEditText
edtnama, TextInputLayout tvnama) {

        if
(Objects.requireNonNull(edtnama.getText()).toString().
trim().isEmpty()) {

            tvnama.setError("Masukkan Nama");

            return false;

        } else {

            tvnama.setError(null);

            return true;

        }

    }


    private boolean validemail(TextInputEditText
edtemail, TextInputLayout tvemail) {

        if
(Objects.requireNonNull(edtemail.getText()).toString().
trim().isEmpty()) {

            tvemail.setError("Masukkan Email");

            return false;

        } else {

            tvemail.setError(null);

            return true;

        }

    }


    private boolean validnohp(TextInputEditText
edtnohp, TextInputLayout tvnohp) {

        if
(Objects.requireNonNull(edtnohp.getText()).toString().t
rim().isEmpty()) {

            tvnohp.setError("Masukkan No. HP");

            return false;

        } else {

            tvnohp.setError(null);

            return true;

        }

    }


    private boolean validalamat(TextInputEditText
edtalamat, TextInputLayout tvalamat) {

        if
(Objects.requireNonNull(edtalamat.getText()).toString(
).trim().isEmpty()) {

            tvalamat.setError("Masukkan Alamat");

            return false;

        } else {

            tvalamat.setError(null);

            return true;

        }

    }
```

```
    private boolean validpasslama(TextInputEditText edtpasslama, TextInputLayout tvpasslama){

        if (Objects.requireNonNull(edtpasslama.getText()).toString().trim().isEmpty()) {

            tvpasslama.setError("Masukkan Password Lama");

            return false;

        } else {

            tvpasslama.setError(null);

            return true;

        }

    }



    private boolean validpassbaru(TextInputEditText edtpassbaru, TextInputLayout tvpassbaru){

        if (Objects.requireNonNull(edtpassbaru.getText()).toString().trim().isEmpty()) {

            tvpassbaru.setError("Masukkan Password Baru");

            return false;

        } else {

            tvpassbaru.setError(null);

            return true;

        }

    }



    private boolean validulangpassbaru(TextInputEditText edtpassbaru, TextInputLayout tvpassbaru){

        if (Objects.requireNonNull(edtpassbaru.getText()).toString().trim().isEmpty()) {

            tvpassbaru.setError("Masukkan Konfirmasi Password Baru");

            return false;

        } else {

            tvpassbaru.setError(null);

            return true;

        }

    }

}
```

Script menu daftar activity

```
package com.dilla.a220280030.woparepare.activity;


import android.annotation.SuppressLint;

import android.content.Intent;

import android.graphics.Color;

import android.os.Bundle;

import android.os.Handler;

import android.view.Gravity;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.RelativeLayout;

import android.widget.TextView;

import android.widget.Toast;


import androidx.annotation.NonNull;

import androidx.appcompat.app.ActionBar;

import androidx.appcompat.app.AppCompatActivity;

import androidx.appcompat.widget.Toolbar;

import androidx.recyclerview.widget.GridLayoutManager;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import androidx.swiperefreshlayout.widget.SwipeRefreshLayout;


import com.dilla.a220280030.woparepare.R;
```

```
import com.dilla.a220280030.woparepare.SessionManager;

import com.dilla.a220280030.woparepare.adapter.MenuDaftarAdapter;

import com.dilla.a220280030.woparepare.api.ApiClient;

import com.dilla.a220280030.woparepare.api.ApiInterface;

import com.dilla.a220280030.woparepare.model.ResultData;



import java.util.List;



import retrofit2.Call;

import retrofit2.Callback;

import retrofit2.Response;



public class MenuDaftarActivity extends AppCompatActivity {



    private RecyclerView recyclerView;

    private SwipeRefreshLayout mSwipeRefreshLayout;

    private RelativeLayout errorlayout;

    private ImageView imgerror;

    private TextView errortitle, errormessage, textCartItemCount;

    private Button btnretry;

    String idkategori, nmkategori, email;

    MenuDaftarAdapter adapter;

    MenuDaftarAdapter.ItemClickListener itemClickListener;

    List<ResultData> resultDataList;

    SessionManager sessionManager;

    ApiInterface apiInterface;



    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_menu_daftar);



        sessionManager = new SessionManager(MenuDaftarActivity.this);

        email = sessionManager.getUserDetail().get(SessionManager.EMAIL);



        Intent get = getIntent();

        idkategori = get.getStringExtra("idkate");

        nmkategori = get.getStringExtra("nmkate");



        errorlayout = findViewById(R.id.rel_error);

        imgerror = findViewById(R.id.imgerror);

        errortitle = findViewById(R.id.tverror);

        errormessage = findViewById(R.id.tvsuberror);

        btnretry = findViewById(R.id.btnretry);



        mSwipeRefreshLayout = findViewById(R.id.swaplistmenudaftar);

        recyclerView = findViewById(R.id.listviewmenudaftar);

        recyclerView.setLayoutManager(new LinearLayoutManager(this));

        recyclerView.setHasFixedSize(true);

        recyclerView.setLayoutManager(new GridLayoutManager(this, 2));



        ambilmenudaftar(idkategori);



mSwipeRefreshLayout.setColorSchemeResources(R.color.colorPrimary);

        mSwipeRefreshLayout.setOnRefreshListener(() -> ambilmenudaftar(idkategori));



        itemClickListener = ((view, position) -> {
```

```
        String idproduk = resultDataList.get(position).getIdProduk();

        String namaproduk = resultDataList.get(position).getNamaProduk();

        String ketproduk = resultDataList.get(position).getKetProduk();

        String hargaproduk = resultDataList.get(position).getHargaProduk();

        String gambarproduk = resultDataList.get(position).getGambarProduk();

        String videoproduk = resultDataList.get(position).getVideoProduk();

        Intent it = new Intent(MenuDaftarActivity.this, DetailProdukActivity.class);

        it.putExtra("idkategori", idkategori);

        it.putExtra("nmkategori", nmkategori);

        it.putExtra("idproduk", idproduk);

        it.putExtra("namaproduk", namaproduk);

        it.putExtra("ketproduk", ketproduk);

        it.putExtra("hargaproduk", hargaproduk);

        it.putExtra("gambarproduk", gambarproduk);

        it.putExtra("videoproduk", videoproduk);

        startActivity(it);

      });


      Toolbar toolbar = findViewById(R.id.toolbar);

      setSupportActionBar(toolbar);

      ActionBar mAction = getSupportActionBar();

      assert mAction != null;

      TextView titleTextView = new TextView(getApplicationContext());

      titleTextView.setText(nmkategori.toUpperCase());

      titleTextView.setTextSize(17);

      titleTextView.setTextColor(Color.WHITE);

      titleTextView.setGravity(Gravity.CENTER);


mAction.setDisplayOptions(ActionBar.DISPLAY_SHOW_CUSTOM);

      mAction.setCustomView(titleTextView, new ActionBar.LayoutParams(ActionBar.LayoutParams.WRAP_CONTENT,
ActionBar.LayoutParams.WRAP_CONTENT,
Gravity.CENTER));

      mAction.setDisplayHomeAsUpEnabled(true);

      toolbar.setNavigationOnClickListener(v -> onBackPressed());

   }


   @Override

   public boolean onCreateOptionsMenu(Menu menu) {

      getMenuInflater().inflate(R.menu.data_diri, menu);

      final MenuItem menuItem = menu.findItem(R.id.keranjang);

      View actionView = menuItem.getActionView();

      textCartItemCount = actionView.findViewById(R.id.cart_badge);

      badgekeranjang(email);

      Handler handler = new Handler();

      handler.postDelayed(new Runnable() {

         @Override

         public void run() {

            badgekeranjang(email);

            handler.postDelayed(this, 1000);

         }

      }, 1000);

      actionView.setOnClickListener(v -> onOptionsItemSelected(menuItem));

      MenuItem listmenu = menu.findItem(R.id.listmenu);

      listmenu.setVisible(false);

      return true;

   }


   @SuppressLint("SetTextI18n")

   @Override
```

```
    public boolean onOptionsItemSelected(MenuItem item) {

        if (item.getItemId() == R.id.data_diri) {

            Intent it = new Intent(MenuDaftarActivity.this, ProfilActivity.class);

            startActivity(it);

        } else if (item.getItemId() == R.id.keranjang) {

            startActivity(new Intent(MenuDaftarActivity.this, KeranjangActivity.class));

        } else if (item.getItemId() == android.R.id.home) {

            onBackPressed();

        }

        return super.onOptionsItemSelected(item);

    }



    private void ambilmenudaftar(String idkategori){

        errorlayout.setVisibility(View.GONE);

        showLoading();

        apiInterface = ApiClient.getApiClient().create(ApiInterface.class);

        Call<List<ResultData>> call = apiInterface.getDaftarMenu(idkategori);

        call.enqueue(new Callback<List<ResultData>>() {



            @Override

            public void onResponse(@NonNull Call<List<ResultData>> call, @NonNull Response<List<ResultData>> response) {

                hideLoading();

                if(response.isSuccessful() && response.body() != null){

                    onGetResult(response.body());

                }else{

                    String errorCode;

                    switch (response.code()){

                        case 404:

                            errorCode = "404 not found";

                            break;

                        case 500:

                            errorCode = "500 server broken";

                            break;

                        default:

                            errorCode = "Unknown error";

                            break;

                    }

                    showErrorMessage(R.drawable.no_result, "No Result", "Please Try Again!-"+ errorCode);

                }

            }



            @Override

            public void onFailure(@NonNull Call<List<ResultData>> call, @NonNull Throwable t) {

                hideLoading();

                onErrorLoading(t.toString());

            }

        });

    }

    private void badgekeranjang(String email){

        apiInterface = ApiClient.getApiClient().create(ApiInterface.class);

        Call<List<ResultData>> call = apiInterface.getCountKeranjang(email);

        call.enqueue(new Callback<List<ResultData>>() {



            @Override

            public void onResponse(@NonNull Call<List<ResultData>> call, @NonNull Response<List<ResultData>> response) {

                if(response.isSuccessful() && response.body() != null){

                    int jml = response.body().get(0).getIsiKeranjang();

                    setupBadge(jml);
```

```
            }else{

                showToast("Gagal meload data", R.layout.toast_gagal);

            }

        }



        @Override

        public void onFailure(@NonNull Call<List<ResultData>> call, @NonNull Throwable t) {

            showToast(t.toString(), R.layout.toast_gagal);

        }

    });

  }



  private void showToast(String message, int layoutId) {

    final Toast toast = new Toast(getApplicationContext());

    toast.setDuration(Toast.LENGTH_LONG);

    @SuppressLint("InflateParams") View custom_view = getLayoutInflater().inflate(layoutId, null);

    TextView tvMessage = custom_view.findViewById(R.id.message);

    tvMessage.setText(message);

    toast.setView(custom_view);

    toast.setGravity(Gravity.CENTER, 0, 0);

    toast.show();

  }



  private void setupBadge(Integer mCartItemCount) {

    if (textCartItemCount != null) {

      if (mCartItemCount == 0) {

        if (textCartItemCount.getVisibility() != View.GONE) {


textCartItemCount.setVisibility(View.GONE);

        }

      } else {

        int maxLength = 2;

        String itemCountText = String.valueOf(Math.min(mCartItemCount, 99));

        if (itemCountText.length() > maxLength) {

          itemCountText = itemCountText.substring(0, maxLength);

        }

        if (mCartItemCount > 99) {

          itemCountText += "+";

        }

        textCartItemCount.setText(itemCountText);

        textCartItemCount.setPadding(0, 0, 0, 0);

        if (textCartItemCount.getVisibility() != View.VISIBLE) {


textCartItemCount.setVisibility(View.VISIBLE);

        }

      }

    }

  }



  private void showLoading() {

    mSwipeRefreshLayout.setRefreshing(true);

  }



  private void hideLoading() {

    mSwipeRefreshLayout.setRefreshing(false);

  }



  @SuppressLint("NotifyDataSetChanged")

  private void onGetResult(List<ResultData> values) {

    adapter = new MenuDaftarAdapter(this, values, itemClickListener);

    adapter.notifyDataSetChanged();

    recyclerView.setAdapter(adapter);
```

```
        resultDataList = values;

    }



    private void onErrorLoading(String message) {

        showErrorMessage(R.drawable.no_result,
"Oops...", "Network failure, Please Try Again!.-"+
message);

    }



    private void showErrorMessage(int imageView,
String title, String message){



        if(errorlayout.getVisibility() == View.GONE){

            errorlayout.setVisibility(View.VISIBLE);

        }



        imgerror.setImageResource(imageView);

        errortitle.setText(title);

        errormessage.setText(message);



        btnretry.setOnClickListener(view ->
ambilmenudaftar(idkategori));

    }

}
```

Script belum bayar

```
package com.dilla.a220280030.woparepare.fragment;



import static
com.dilla.a220280030.woparepare.api.ApiClient.BASE
_URL;



import android.annotation.SuppressLint;

import android.content.Intent;

import android.graphics.Color;

import android.os.Bundle;



import androidx.annotation.NonNull;

import androidx.appcompat.app.AlertDialog;

import androidx.fragment.app.Fragment;

import
androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import
androidx.swiperefreshlayout.widget.SwipeRefreshLayo
ut;



import android.view.LayoutInflater;

import android.view.View;



import android.view.ViewGroup;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.RelativeLayout;

import android.widget.TextView;



import com.bumptech.glide.Glide;

import
com.bumptech.glide.load.engine.DiskCacheStrategy;

import com.bumptech.glide.request.RequestOptions;

import com.dilla.a220280030.woparepare.R;

import
com.dilla.a220280030.woparepare.SessionManager;

import
com.dilla.a220280030.woparepare.activity.Konfirmasi
BayarActivity;

import
com.dilla.a220280030.woparepare.activity.RincianPesa
nanActivity;

import
com.dilla.a220280030.woparepare.adapter.BelumBayar
Adapter;

import
com.dilla.a220280030.woparepare.api.ApiClient;

import
com.dilla.a220280030.woparepare.api.ApiInterface;

import
com.dilla.a220280030.woparepare.model.ResultData;

import com.github.chrisbanes.photoview.PhotoView;
```

```
import com.google.android.material.dialog.MaterialAlertDialogBuilder;


import java.util.List;


import retrofit2.Call;

import retrofit2.Callback;

import retrofit2.Response;


public class BelumBayarFragment extends Fragment {


    private RecyclerView recyclerView;

    private SwipeRefreshLayout mSwipeRefreshLayout;

    private RelativeLayout errorlayout;

    private ImageView imgerror;

    private TextView errortitle, errormessage;

    private Button btnretry;

    String email, nohp;

    BelumBayarAdapter adapter;

    BelumBayarAdapter.ItemClickListener itemClickListener;

    List<ResultData> resultDataList;

    ApiInterface apiInterface;

    SessionManager sessionManager;



    @Override

    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {

        View rootView = inflater.inflate(R.layout.fragment_belum_bayar, container, false);



        sessionManager = new SessionManager(getContext());

        email = sessionManager.getUserDetail().get(SessionManager.EMAIL);

        nohp = sessionManager.getUserDetail().get(SessionManager.NOHP);



        errorlayout = rootView.findViewById(R.id.rel_error);

        imgerror = rootView.findViewById(R.id.imgerror);

        errortitle = rootView.findViewById(R.id.tverror);

        errormessage = rootView.findViewById(R.id.tvsuberror);

        btnretry = rootView.findViewById(R.id.btnretry);



        mSwipeRefreshLayout = rootView.findViewById(R.id.swaplistbelumbayar);

        recyclerView = rootView.findViewById(R.id.listviewbelumbayar);

        recyclerView.setLayoutManager(new LinearLayoutManager(getContext()));

        recyclerView.setHasFixedSize(true);



        ambilbelumbayar(email);



mSwipeRefreshLayout.setColorSchemeResources(R.color.colorPrimary);

        mSwipeRefreshLayout.setOnRefreshListener(() -> ambilbelumbayar(email));



        itemClickListener = new BelumBayarAdapter.ItemClickListener() {

            @Override

            public void OnItemClickProdukLebih(View view, int position) {

                String idTransaksi = resultDataList.get(position).getIdTransaksi();

                String statusByr = resultDataList.get(position).getStatusBayar();

                String statusBrng = resultDataList.get(position).getStatusBarang();

                String totjml = resultDataList.get(position).getJumlahData();
```

```
            String tothrg = resultDataList.get(position).getTotalHarga();


            Intent intent = new Intent(getContext(), RincianPesananActivity.class);

            intent.putExtra("IDTRANSAKSI", idTransaksi);

            intent.putExtra("STATUSBAYAR", statusByr);

            intent.putExtra("STATUSBARANG", statusBrng);

            intent.putExtra("TOTAL_JUMLAH", totjml);

            intent.putExtra("TOTAL_HARGA", tothrg);

            startActivity(intent);

        }


        @Override

        public void OnItemClickBayarSekarang(View view, int position) {

            String idTransaksi = resultDataList.get(position).getIdTransaksi();

            String totalHarga = resultDataList.get(position).getTotalHarga();

            String metoteBayar = resultDataList.get(position).getMetodeBayar();

            Intent it = new Intent(getContext(), KonfirmasiBayarActivity.class);

            it.putExtra("IdTransaksi", idTransaksi);

            it.putExtra("TotalHarga", totalHarga);

            it.putExtra("MetodeBayar", metoteBayar);

            startActivity(it);

        }


        @Override

        public void OnItemClickBuktiTransfer(View view, int position) {

            String idTransaksi = resultDataList.get(position).getIdTransaksi();

            buktitf(idTransaksi);

        }
    };


    setHasOptionsMenu(true);

    return rootView;

}


private void buktitf(String idtransaksi) {

    LayoutInflater inflater = getLayoutInflater();

    View customView = inflater.inflate(R.layout.item_dialog_bukti, null);

    TextView juduldialog = customView.findViewById(R.id.txtdialjudul);

    PhotoView imgbuktrfunik = customView.findViewById(R.id.imgbuktitrf);

    juduldialog.setText("Bukti Transfer");


juduldialog.setTextAlignment(View.TEXT_ALIGNMENT_CENTER);


juduldialog.setTextColor(Color.parseColor("#FFFFFF"));


juduldialog.setBackgroundColor(Color.parseColor("#3C5B6F"));

    AlertDialog dialog = new MaterialAlertDialogBuilder(getContext(), R.style.RoundedMaterialDialog).setView(customView).show();

    Glide.with(this)

            .load(BASE_URL + "bukti/" + idtransaksi +".jpg")


.apply(RequestOptions.skipMemoryCacheOf(true))


.apply(RequestOptions.diskCacheStrategyOf(DiskCacheStrategy.NONE))

            .placeholder(R.drawable.noimage)

            .into(imgbuktrfunik);

    dialog.setCancelable(false);


dialog.findViewById(R.id.tutupdialog).setOnClickListener(v -> dialog.dismiss());

}
```

```
    private void ambilbelumbayar(String email) {

        errorlayout.setVisibility(View.GONE);

        showLoading();

        apiInterface = ApiClient.getApiClient().create(ApiInterface.class);

        Call<List<ResultData>> call = apiInterface.getTampilBelumBayar(email);

        call.enqueue(new Callback<List<ResultData>>() {


            @Override

            public void onResponse(@NonNull Call<List<ResultData>> call, @NonNull Response<List<ResultData>> response) {

                hideLoading();

                if(response.isSuccessful() && response.body() != null){

                    onGetResult(response.body());

                }else{

                    String errorCode;

                    switch (response.code()){

                        case 404:

                            errorCode = "404 not found";

                            break;

                        case 500:

                            errorCode = "500 server broken";

                            break;

                        default:

                            errorCode = "Unknown error";

                            break;

                    }

                    showErrorMessage(R.drawable.no_result, "No Result", "Please Try Again!-"+ errorCode);

                }

            }


            @Override

            public void onFailure(@NonNull Call<List<ResultData>> call, @NonNull Throwable t) {

                hideLoading();

                onErrorLoading(t.toString());

            }

        });

    }


    private void showLoading() {

        mSwipeRefreshLayout.setRefreshing(true);

    }


    private void hideLoading() {

        mSwipeRefreshLayout.setRefreshing(false);

    }


    @SuppressLint("NotifyDataSetChanged")

    private void onGetResult(List<ResultData> results) {

        adapter = new BelumBayarAdapter(getContext(), results, itemClickListener);

        adapter.notifyDataSetChanged();

        recyclerView.setAdapter(adapter);

        resultDataList = results;

    }


    private void onErrorLoading(String message) {

        showErrorMessage(R.drawable.no_result, "Oops...", "Network failure, Please Try Again!.\n"+ message);

    }


    private void showErrorMessage(int imageView, String title, String message){


        if(errorlayout.getVisibility() == View.GONE){

            errorlayout.setVisibility(View.VISIBLE);
```

```
        }


        imgerror.setImageResource(imageView);

        errortitle.setText(title);

        errormessage.setText(message);


        btnretry.setOnClickListener(view -> ambilbelumbayar(email));


    }

}
```

Script checkout activity

```
package com.dilla.a220280030.woparepare.activity;


import android.annotation.SuppressLint;

import android.app.DatePickerDialog;

import android.app.ProgressDialog;

import android.content.Intent;

import android.graphics.Color;

import android.icu.util.Calendar;

import android.os.Bundle;

import android.view.Gravity;

import android.view.MenuItem;

import android.view.View;

import android.widget.LinearLayout;

import android.widget.TextView;

import android.widget.Toast;


import androidx.appcompat.app.ActionBar;

import androidx.appcompat.app.AppCompatActivity;

import androidx.appcompat.widget.Toolbar;

import androidx.core.content.ContextCompat;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import com.dilla.a220280030.woparepare.R;

import com.dilla.a220280030.woparepare.SessionManager;

import com.dilla.a220280030.woparepare.adapter.CheckoutAdapter;

import com.dilla.a220280030.woparepare.api.ApiClient;

import com.dilla.a220280030.woparepare.api.ApiInterface;

import com.dilla.a220280030.woparepare.model.Data;

import com.dilla.a220280030.woparepare.model.ResultData;

import com.google.android.material.bottomsheet.BottomSheetDialog;


import org.jetbrains.annotations.NotNull;


import java.text.DecimalFormat;

import java.util.ArrayList;

import java.util.List;

import java.util.Objects;


import retrofit2.Call;

import retrofit2.Callback;

import retrofit2.Response;


public class CheckoutActivity extends AppCompatActivity {


    SessionManager sessionManager;

    String email, nohp, nama, alamat, idKategori, idProduk, namaKategori, namaProduk, harga, jumlah, gambar;

    String[] idKategoriArray, idProdukArray, namaKategoriArray, namaProdukArray, hargaArray, jumlahArray, gambarArray;

    int totalJumlah, totalHarga;
```

```
    ApiInterface apiInterface;
    private ProgressDialog progress;

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

        sessionManager = new
SessionManager(CheckoutActivity.this);
        email =
sessionManager.getUserDetail().get(SessionManager.E
MAIL);
        nohp =
sessionManager.getUserDetail().get(SessionManager.N
OHP);
        nama =
sessionManager.getUserDetail().get(SessionManager.N
AMA);
        alamat =
sessionManager.getUserDetail().get(SessionManager.A
LAMAT);

        Intent intent = getIntent();
        idKategori =
intent.getStringExtra("IDKATEGORI");
        idProduk = intent.getStringExtra("IDPRODUK");
        namaKategori =
intent.getStringExtra("NAMAKATEGORI");
        namaProduk =
intent.getStringExtra("NAMAPRODUK");
        harga = intent.getStringExtra("HARGA");
        jumlah = intent.getStringExtra("JUMLAH");
        gambar = intent.getStringExtra("GAMBAR");
        totalJumlah =
intent.getIntExtra("TOTAL_JUMLAH", 0);
        totalHarga =
intent.getIntExtra("TOTAL_HARGA", 0);

        idKategoriArray = idKategori.split(", ");
        idProdukArray = idProduk.split(", ");
        namaKategoriArray = namaKategori.split(", ");
        namaProdukArray = namaProduk.split(", ");
        hargaArray = harga.split(", ");
        jumlahArray = jumlah.split(", ");
        gambarArray = gambar.split(", ");

        List<ResultData> resultDataList = new
ArrayList<>();
        for (int i = 0; i < namaProdukArray.length; i++) {
            resultDataList.add(new ResultData(
                    idKategoriArray[i],
                    idProdukArray[i],
                    namaKategoriArray[i],
                    namaProdukArray[i],

formatRupiah(Integer.parseInt(hargaArray[i])),
                    jumlahArray[i],
                    null,
                    gambarArray[i],
                    null,
                    true
            ));
        }

        RecyclerView recyclerView =
findViewById(R.id.listviewcheckout);
        recyclerView.setLayoutManager(new
LinearLayoutManager(this));
        CheckoutAdapter adapter = new
CheckoutAdapter(this, resultDataList);
        recyclerView.setAdapter(adapter);

        progress = new
ProgressDialog(CheckoutActivity.this);
        progress.setCancelable(false);
        progress.setMessage("Tunggu Sebentar...");
```

```
        TextView tvalamat = findViewById(R.id.tvDetailAlamat);

        TextView tvtglacara = findViewById(R.id.texttglacara);

        TextView tvlamaacara = findViewById(R.id.textlamaacara);

        TextView tvTotalJml = findViewById(R.id.tvtotaljml);

        TextView tvTotalByr = findViewById(R.id.tvtotalbyr);

        LinearLayout metodebyr = findViewById(R.id.metodebyr);

        TextView tvpilihmetode = findViewById(R.id.tvpilihmetode);

        tvpilihmetode.setText("Pilih Metode Pembayaran");

        tvpilihmetode.setTextColor(Color.RED);

        LinearLayout checkout = findViewById(R.id.checkout);

        TextView tvbuatpesanan = findViewById(R.id.tvcheckout);



        tvalamat.setText(nama + "\n" + email + " | " + nohp + "\n" + alamat);


findViewById(R.id.btntglacara).setOnClickListener(v -> {

            final Calendar c = Calendar.getInstance();

            int tahun = c.get(Calendar.YEAR);

            int bulan = c.get(Calendar.MONTH);

            int tanggal = c.get(Calendar.DAY_OF_MONTH);

            DatePickerDialog datePickerDialog = new DatePickerDialog(CheckoutActivity.this, R.style.CustomDatePickerDialog, (view, year, month, dayOfMonth) -> {

                String formattedDate = String.format("%02d-%02d-%d", dayOfMonth, month + 1, year);

                tvtglacara.setText(formattedDate);

            }, tahun, bulan, tanggal);

            int colorAccent = ContextCompat.getColor(CheckoutActivity.this, R.color.colorAccent);

            datePickerDialog.setOnShowListener(dialog -> {


datePickerDialog.getButton(DatePickerDialog.BUTTON_POSITIVE).setTextColor(colorAccent);


datePickerDialog.getButton(DatePickerDialog.BUTTON_NEGATIVE).setTextColor(colorAccent);

            });

            datePickerDialog.show();

        });


findViewById(R.id.btnlamaacara).setOnClickListener(v -> {

            BottomSheetDialog bottomSheetDialog = new BottomSheetDialog(CheckoutActivity.this);


bottomSheetDialog.setContentView(R.layout.lama_sewa);



            TextView satuhari = bottomSheetDialog.findViewById(R.id.satuhari);

            TextView duahari = bottomSheetDialog.findViewById(R.id.duahari);

            TextView tigahari = bottomSheetDialog.findViewById(R.id.tigahari);

            TextView empathari = bottomSheetDialog.findViewById(R.id.empathari);

            TextView limahari = bottomSheetDialog.findViewById(R.id.limahari);



            final TextView[] lastSelected = {null};

            View.OnClickListener onClickListener = view -> {

                TextView selectedTextView = (TextView) view;

                String selectedMethod = selectedTextView.getText().toString();

                tvlamaacara.setText(selectedMethod);

                lastSelected[0] = selectedTextView;

                bottomSheetDialog.dismiss();

            };
```

```
        satuhari.setOnClickListener(onClickListener);

        duahari.setOnClickListener(onClickListener);

        tigahari.setOnClickListener(onClickListener);

        empathari.setOnClickListener(onClickListener);

        limahari.setOnClickListener(onClickListener);

        bottomSheetDialog.show();

    });

    tvTotalJml.setText("Total Pembayaran (" + totalJumlah + " Produk):");

    tvTotalByr.setText(formatRupiah(totalHarga));

    metodebyr.setOnClickListener(v -> {

        BottomSheetDialog bottomSheetDialog = new BottomSheetDialog(CheckoutActivity.this);


bottomSheetDialog.setContentView(R.layout.metode_bayar);



//                TextView paymentMethodCOD = bottomSheetDialog.findViewById(R.id.payment_method_cod);

        TextView paymentBSI = bottomSheetDialog.findViewById(R.id.payment_method_bsi);

        TextView paymentBRI = bottomSheetDialog.findViewById(R.id.payment_method_bri);

        TextView paymentBCA = bottomSheetDialog.findViewById(R.id.payment_method_bca);

        TextView paymentBNI = bottomSheetDialog.findViewById(R.id.payment_method_bni);

        TextView paymentMandiri = bottomSheetDialog.findViewById(R.id.payment_method_mandiri);



        final TextView[] lastSelected = {null};

        View.OnClickListener onClickListener = view -> {

            TextView selectedTextView = (TextView) view;

            String selectedMethod = selectedTextView.getText().toString();


//                if (selectedTextView == paymentMethodCOD) {

//                    tvpilihmetode.setText(selectedMethod);

//                    tvpilihmetode.setTextColor(Color.BLACK);

//                } else {

                    tvpilihmetode.setText("Transfer Bank - " + selectedMethod);


tvpilihmetode.setTextColor(Color.BLACK);

//                }

            lastSelected[0] = selectedTextView;

            bottomSheetDialog.dismiss();

        };



//        paymentMethodCOD.setOnClickListener(onClickListener);


paymentBSI.setOnClickListener(onClickListener);


paymentBRI.setOnClickListener(onClickListener);


paymentBCA.setOnClickListener(onClickListener);


paymentBNI.setOnClickListener(onClickListener);


paymentMandiri.setOnClickListener(onClickListener);

        bottomSheetDialog.show();

    });

//    for (String id : idKategoriArray) {

//        Log.d("CheckoutActivity", "ID Kategori: " + id);

//    }

    boolean isIdKategori = true;

    for (String id : idKategoriArray) {

        if (!id.equals("2")) {

            isIdKategori = false;

            break;

        }
```

```
        }

        if (isIdKategori) {

findViewById(R.id.btnlamaacara).setVisibility(View.GONE);

            tvlamaacara.setVisibility(View.GONE);

            tvbuatpesanan.setText("Buat Pesanan");

            checkout.setOnClickListener(v -> {

                String selectedPaymentMethod = tvpilihmetode.getText().toString();

                if (tvtglacara.getText().equals("")){

                    showToast("Pilih Tanggal Acara", R.layout.toast_info);

                } else if (selectedPaymentMethod.equals("Pilih Metode Pembayaran")){

                    showToast("Pilih Metode Pembayaran", R.layout.toast_info);

                } else {

                    String tglnya = tvtglacara.getText().toString();

                    String potonghari = tvlamaacara.getText().toString();

                    String lamanya = potonghari.replace(" Hari", "");

                    String metodenya = selectedPaymentMethod.replace("Transfer Bank - ", "");

                    simpantransaksi(email, nohp, idKategori, idProduk, jumlah, tglnya, lamanya, metodenya);

                }

            });

        } else {

findViewById(R.id.btnlamaacara).setVisibility(View.VISIBLE);

            tvlamaacara.setVisibility(View.VISIBLE);

            tvbuatpesanan.setText("Buat Pesanan");

            checkout.setOnClickListener(v -> {

                String selectedPaymentMethod = tvpilihmetode.getText().toString();

                if (tvtglacara.getText().equals("")){

                    showToast("Pilih Tanggal Acara", R.layout.toast_info);

                } else if (tvlamaacara.getText().equals("")){

                    showToast("Pilih Lama Sewa", R.layout.toast_info);

                } else if (selectedPaymentMethod.equals("Pilih Metode Pembayaran")){

                    showToast("Pilih Metode Pembayaran", R.layout.toast_info);

                } else {

                    String tglnya = tvtglacara.getText().toString();

                    String potonghari = tvlamaacara.getText().toString();

                    String lamanya = potonghari.replace(" Hari", "");

                    String metodenya = selectedPaymentMethod.replace("Transfer Bank - ", "");

                    simpantransaksi(email, nohp, idKategori, idProduk, jumlah, tglnya, lamanya, metodenya);

                }

            });

        }


        Toolbar toolbar = findViewById(R.id.toolbar);

        setSupportActionBar(toolbar);

        ActionBar mAction = getSupportActionBar();

        assert mAction != null;

        TextView titleTextView = new TextView(getApplicationContext());

        titleTextView.setText("Checkout".toUpperCase());

        titleTextView.setTextSize(17);

        titleTextView.setTextColor(Color.WHITE);

        titleTextView.setGravity(Gravity.CENTER);


mAction.setDisplayOptions(ActionBar.DISPLAY_SHOW_CUSTOM);

        mAction.setCustomView(titleTextView, new ActionBar.LayoutParams(ActionBar.LayoutParams.WRAP_CONTENT,
```

```
ActionBar.LayoutParams.WRAP_CONTENT,
Gravity.CENTER));

    mAction.setDisplayHomeAsUpEnabled(true);

    toolbar.setNavigationOnClickListener(v ->
onBackPressed());

  }



  @SuppressLint("SetTextI18n")

  @Override

  public boolean onOptionsItemSelected(MenuItem
item) {

    if (item.getItemId() == android.R.id.home) {

      onBackPressed();

    }

    return super.onOptionsItemSelected(item);

  }



  private void simpantransaksi(String email, String
nohp, String idkate, String idproduk, String jumlah,
String tglacara, String lamaacara, String metode) {

    progress.show();

    apiInterface =
ApiClient.getApiClient().create(ApiInterface.class);

    Call<Data> masukCall =
apiInterface.simpantransaksiRespone(email, nohp,
idkate, idproduk, jumlah, tglacara, lamaacara, metode);

    masukCall.enqueue(new Callback<Data>() {

      @Override

      public void onResponse(@NotNull Call<Data>
call, @NotNull Response<Data> response) {

        progress.hide();

        if(response.body() != null &&
response.isSuccessful() && response.body().isStatus())
{

          showToast(response.body().getMessage(),
R.layout.toast_berhasil);

          Intent it = new
Intent(CheckoutActivity.this, TransaksiActivity.class);

          it.putExtra("SELECTED_TAB", 0);

          startActivity(it);

          finish();

        }else{

showToast(Objects.requireNonNull(response.body()).g
etMessage(), R.layout.toast_gagal);

        }

      }



      @SuppressLint("SetTextI18n")

      @Override

      public void onFailure(@NotNull Call<Data>
call, @NotNull Throwable t) {

        progress.hide();

        showToast("Tidak ada koneksi internet,
silahkan coba lagi!", R.layout.toast_gagal);

      }

    });

  }



  private String formatRupiah(int number) {

    DecimalFormat formatter = new
DecimalFormat("#,###,###,###");

    return "Rp" + formatter.format(number);

  }



  private void showToast(String message, int layoutId)
{

    final Toast toast = new
Toast(getApplicationContext());

    toast.setDuration(Toast.LENGTH_LONG);

    @SuppressLint("InflateParams") View
custom_view = getLayoutInflater().inflate(layoutId,
null);

    TextView tvMessage =
custom_view.findViewById(R.id.message);

    tvMessage.setText(message);

    toast.setView(custom_view);
```

```
        toast.setGravity(Gravity.CENTER, 0, 0);
        toast.show();
    }
}
```

Script dalam proses

```
package com.dilla.a220280030.woparepare.fragment;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;

import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import androidx.swiperefreshlayout.widget.SwipeRefreshLayout;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.RelativeLayout;
import android.widget.TextView;

import com.dilla.a220280030.woparepare.R;
import com.dilla.a220280030.woparepare.SessionManager;
import com.dilla.a220280030.woparepare.activity.RincianPesananActivity;
import com.dilla.a220280030.woparepare.adapter.DalamProsesAdapter;
import com.dilla.a220280030.woparepare.api.ApiClient;
import com.dilla.a220280030.woparepare.api.ApiInterface;
import com.dilla.a220280030.woparepare.model.ResultData;

import java.util.List;

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

public class DalamProsesFragment extends Fragment {

    private RecyclerView recyclerView;
    private SwipeRefreshLayout mSwipeRefreshLayout;
    private RelativeLayout errorlayout;
    private ImageView imgerror;
    private TextView errortitle, errormessage;
    private Button btnretry;
    String email, nohp;
    DalamProsesAdapter adapter;
    DalamProsesAdapter.ItemClickListener itemClickListener;
    List<ResultData> resultDataList;
    ApiInterface apiInterface;
    SessionManager sessionManager;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        View rootView = inflater.inflate(R.layout.fragment_dalam_proses, container, false);

        sessionManager = new SessionManager(getContext());
```

```
        email = sessionManager.getUserDetail().get(SessionManager.EMAIL);

        nohp = sessionManager.getUserDetail().get(SessionManager.NOHP);



        errorlayout = rootView.findViewById(R.id.rel_error);

        imgerror = rootView.findViewById(R.id.imgerror);

        errortitle = rootView.findViewById(R.id.tverror);

        errormessage = rootView.findViewById(R.id.tvsuberror);

        btnretry = rootView.findViewById(R.id.btnretry);



        mSwipeRefreshLayout = rootView.findViewById(R.id.swaplistdalamproses);

        recyclerView = rootView.findViewById(R.id.listviewdalamproses);

        recyclerView.setLayoutManager(new LinearLayoutManager(getContext()));

        recyclerView.setHasFixedSize(true);



        ambildalamproses(email);



mSwipeRefreshLayout.setColorSchemeResources(R.color.colorPrimary);

        mSwipeRefreshLayout.setOnRefreshListener(() -> ambildalamproses(email));



        itemClickListener = (view, position) -> {

            String idTransaksi = resultDataList.get(position).getIdTransaksi();

            String statusByr = resultDataList.get(position).getStatusBayar();

            String statusBrng = resultDataList.get(position).getStatusBarang();

            String totjml = resultDataList.get(position).getJumlahData();

            String tothrg = resultDataList.get(position).getTotalHarga();

            Intent intent = new Intent(getContext(), RincianPesananActivity.class);

            intent.putExtra("IDTRANSAKSI", idTransaksi);

            intent.putExtra("STATUSBAYAR", statusByr);

            intent.putExtra("STATUSBARANG", statusBrng);

            intent.putExtra("TOTAL_JUMLAH", totjml);

            intent.putExtra("TOTAL_HARGA", tothrg);

            startActivity(intent);

        };



        setHasOptionsMenu(true);

        return rootView;

    }



    private void ambildalamproses(String email) {

        errorlayout.setVisibility(View.GONE);

        showLoading();

        apiInterface = ApiClient.getApiClient().create(ApiInterface.class);

        Call<List<ResultData>> call = apiInterface.getTampilDalamProses(email);

        call.enqueue(new Callback<List<ResultData>>() {



            @Override

            public void onResponse(@NonNull Call<List<ResultData>> call, @NonNull Response<List<ResultData>> response) {

                hideLoading();

                if(response.isSuccessful() && response.body() != null){

                    onGetResult(response.body());

                }else{

                    String errorCode;

                    switch (response.code()){

                        case 404:
```

```
                    errorCode = "404 not found";
                    break;
                case 500:
                    errorCode = "500 server broken";
                    break;
                default:
                    errorCode = "Unknown error";
                    break;
            }
            showErrorMessage(R.drawable.no_result,
"No Result", "Please Try Again!-"+ errorCode);
        }
    }

    @Override
    public void onFailure(@NonNull
Call<List<ResultData>> call, @NonNull Throwable t)
{
        hideLoading();
        onErrorLoading(t.toString());
    }
  });
}

private void showLoading() {
  mSwipeRefreshLayout.setRefreshing(true);
}

private void hideLoading() {
  mSwipeRefreshLayout.setRefreshing(false);
}

@SuppressLint("NotifyDataSetChanged")
private void onGetResult(List<ResultData> results) {
  adapter = new DalamProsesAdapter(getContext(),
results, itemClickListener);
  adapter.notifyDataSetChanged();
  recyclerView.setAdapter(adapter);
  resultDataList = results;
}

private void onErrorLoading(String message) {
  showErrorMessage(R.drawable.no_result,
"Oops...", "Network failure, Please Try Again!.\n"+
message);
}

private void showErrorMessage(int imageView,
String title, String message){

  if(errorlayout.getVisibility() == View.GONE){
    errorlayout.setVisibility(View.VISIBLE);
  }

  imgerror.setImageResource(imageView);
  errortitle.setText(title);
  errormessage.setText(message);

  btnretry.setOnClickListener(view ->
ambildalamproses(email));

}
}
```

Script selesai

```
package com.dilla.a220280030.woparepare.fragment;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.graphics.Color;
import android.os.Bundle;

import androidx.annotation.NonNull;
```

```
import androidx.appcompat.app.AlertDialog;

import androidx.fragment.app.Fragment;

import
androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import
androidx.swiperefreshlayout.widget.SwipeRefreshLayo
ut;



import android.text.Editable;

import android.text.TextWatcher;

import android.view.Gravity;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.RelativeLayout;

import android.widget.TextView;

import android.widget.Toast;



import com.dilla.a220280030.woparepare.R;

import
com.dilla.a220280030.woparepare.SessionManager;

import
com.dilla.a220280030.woparepare.activity.RincianPesa
nanActivity;

import
com.dilla.a220280030.woparepare.adapter.SelesaiAdap
ter;

import
com.dilla.a220280030.woparepare.api.ApiClient;

import
com.dilla.a220280030.woparepare.api.ApiInterface;

import com.dilla.a220280030.woparepare.model.Data;

import
com.dilla.a220280030.woparepare.model.ResultData;

import
com.google.android.material.dialog.MaterialAlertDialo
gBuilder;

import
com.google.android.material.textfield.TextInputEditTex
t;

import
com.google.android.material.textfield.TextInputLayout;



import org.jetbrains.annotations.NotNull;



import java.util.List;

import java.util.Objects;



import retrofit2.Call;

import retrofit2.Callback;

import retrofit2.Response;



public class SelesaiFragment extends Fragment {



    private RecyclerView recyclerView;

    private SwipeRefreshLayout mSwipeRefreshLayout;

    private RelativeLayout errorlayout;

    private ImageView imgerror;

    private TextView errortitle, errormessage;

    private Button btnretry;

    String email, nohp;

    SelesaiAdapter adapter;

    SelesaiAdapter.ItemClickListener itemClickListener;

    List<ResultData> resultDataList;

    ApiInterface apiInterface;

    SessionManager sessionManager;



    @Override

    public View onCreateView(LayoutInflater inflater,
ViewGroup container, Bundle savedInstanceState) {

        View rootView =
inflater.inflate(R.layout.fragment_selesai, container,
false);
```

```
        sessionManager = new SessionManager(getContext());

        email = sessionManager.getUserDetail().get(SessionManager.EMAIL);

        nohp = sessionManager.getUserDetail().get(SessionManager.NOHP);



        errorlayout = rootView.findViewById(R.id.rel_error);

        imgerror = rootView.findViewById(R.id.imgerror);

        errortitle = rootView.findViewById(R.id.tverror);

        errormessage = rootView.findViewById(R.id.tvsuberror);

        btnretry = rootView.findViewById(R.id.btnretry);



        mSwipeRefreshLayout = rootView.findViewById(R.id.swaplistselesai);

        recyclerView = rootView.findViewById(R.id.listviewselesai);

        recyclerView.setLayoutManager(new LinearLayoutManager(getContext()));

        recyclerView.setHasFixedSize(true);



        ambilselesai(email);



mSwipeRefreshLayout.setColorSchemeResources(R.color.colorPrimary);

        mSwipeRefreshLayout.setOnRefreshListener(() -> ambilselesai(email));



        itemClickListener = new SelesaiAdapter.ItemClickListener() {

            @Override

            public void OnItemClickProdukLebih(View view, int position) {

                String idTransaksi = resultDataList.get(position).getIdTransaksi();

                String statusByr = resultDataList.get(position).getStatusBayar();

                String statusBrng = resultDataList.get(position).getStatusBarang();

                String totjml = resultDataList.get(position).getJumlahData();

                String tothrg = resultDataList.get(position).getTotalHarga();



                Intent intent = new Intent(getContext(), RincianPesananActivity.class);

                intent.putExtra("IDTRANSAKSI", idTransaksi);

                intent.putExtra("STATUSBAYAR", statusByr);

                intent.putExtra("STATUSBARANG", statusBrng);

                intent.putExtra("TOTAL_JUMLAH", totjml);

                intent.putExtra("TOTAL_HARGA", tothrg);

                startActivity(intent);

            }



            @Override

            public void OnItemClickBeriUlasan(View view, int position) {

                String idTransaksi = resultDataList.get(position).getIdTransaksi();

                formberiulasan(idTransaksi);

            }

        };



        setHasOptionsMenu(true);

        return rootView;

    }



    private void formberiulasan(String idnya) {

        LayoutInflater inflater = getLayoutInflater();

        View customView = inflater.inflate(R.layout.item_dialog_ulasan, null);

        TextView judduldialog = customView.findViewById(R.id.txtdialjudul);
```

```
        TextInputLayout tvulasan = customView.findViewById(R.id.tvulasan);

        TextInputEditText edtulasan = customView.findViewById(R.id.edtulasan);

        Button btnkirim = customView.findViewById(R.id.btnkirim);

        judulldialog.setText("Beri Ulasan");


judulldialog.setTextAlignment(View.TEXT_ALIGNMENT_CENTER);


judulldialog.setTextColor(Color.parseColor("#FFFFFF"));


judulldialog.setBackgroundColor(Color.parseColor("#3C5B6F"));

        AlertDialog dialog = new MaterialAlertDialogBuilder(getContext(), R.style.RoundedMaterialDialog).setView(customView).show();

        edtulasan.addTextChangedListener(new TextWatcher() {

            @Override

            public void beforeTextChanged(CharSequence s, int start, int count, int after) {



            }



            @Override

            public void onTextChanged(CharSequence s, int start, int before, int count) {

                validulasan(edtulasan,tvulasan);

            }



            @Override

            public void afterTextChanged(Editable s) {



            }

        });

        btnkirim.setOnClickListener(v -> {

            if (!validulasan(edtulasan,tvulasan)) {

                return;

            }


berikanulasan(Objects.requireNonNull(edtulasan.getText()).toString(), idnya, email);

            dialog.dismiss();

        });

        dialog.setCancelable(false);


dialog.findViewById(R.id.tutupdialog).setOnClickListener(v -> dialog.dismiss());

    }



    private boolean validulasan(TextInputEditText edtulasan, TextInputLayout tvulasan) {

        if (Objects.requireNonNull(edtulasan.getText()).toString().trim().isEmpty()) {

            tvulasan.setError("Berikan Ulasan Anda");

            return false;

        } else {

            tvulasan.setError(null);

            return true;

        }

    }



    private void berikanulasan(String ulasan, String idtransaksi, String email) {

        apiInterface = ApiClient.getApiClient().create(ApiInterface.class);

        Call<Data> masukCall = apiInterface.beriulasanRespone(ulasan, idtransaksi, email);

        masukCall.enqueue(new Callback<Data>() {

            @Override

            public void onResponse(@NotNull Call<Data> call, @NotNull Response<Data> response) {

                if(response.body() != null && response.isSuccessful() && response.body().isStatus()) {
```

```
                showToast(response.body().getMessage(), R.layout.toast_berhasil);

                ambilselesai(email);

            }else{

                assert response.body() != null;

                showToast(response.body().getMessage(), R.layout.toast_gagal);

            }

        }


        @SuppressLint("SetTextI18n")

        @Override

        public void onFailure(@NotNull Call<Data> call, @NotNull Throwable t) {

            showToast("Tidak ada koneksi internet, silahkan coba lagi!", R.layout.toast_gagal);

        }

    });

  }


  private void ambilselesai(String email) {

    errorlayout.setVisibility(View.GONE);

    showLoading();

    apiInterface = ApiClient.getApiClient().create(ApiInterface.class);

    Call<List<ResultData>> call = apiInterface.getTampilSelesai(email);

    call.enqueue(new Callback<List<ResultData>>() {


        @Override

        public void onResponse(@NonNull Call<List<ResultData>> call, @NonNull Response<List<ResultData>> response) {

            hideLoading();

            if(response.isSuccessful() && response.body() != null){

                onGetResult(response.body());

            }else{

                String errorCode;

                switch (response.code()){

                    case 404:

                        errorCode = "404 not found";

                        break;

                    case 500:

                        errorCode = "500 server broken";

                        break;

                    default:

                        errorCode = "Unknown error";

                        break;

                }

                showErrorMessage(R.drawable.no_result, "No Result", "Please Try Again!-"+ errorCode);

            }

        }


        @Override

        public void onFailure(@NonNull Call<List<ResultData>> call, @NonNull Throwable t) {

            hideLoading();

            onErrorLoading(t.toString());

        }

    });

  }


  private void showToast(String message, int layoutId) {

    final Toast toast = new Toast(getContext());

    toast.setDuration(Toast.LENGTH_LONG);

    @SuppressLint("InflateParams") View custom_view = getLayoutInflater().inflate(layoutId, null);

    TextView tvMessage = custom_view.findViewById(R.id.message);

    tvMessage.setText(message);
```

```
        toast.setView(custom_view);
        toast.setGravity(Gravity.CENTER, 0, 0);
        toast.show();
    }

    private void showLoading() {
        mSwipeRefreshLayout.setRefreshing(true);
    }

    private void hideLoading() {
        mSwipeRefreshLayout.setRefreshing(false);
    }

    @SuppressLint("NotifyDataSetChanged")
    private void onGetResult(List<ResultData> results) {
        adapter = new SelesaiAdapter(getContext(), results, itemClickListener);
        adapter.notifyDataSetChanged();
        recyclerView.setAdapter(adapter);
        resultDataList = results;
    }

    private void onErrorLoading(String message) {
        showErrorMessage(R.drawable.no_result, "Oops...", "Network failure, Please Try Again!.\n"+ message);
    }

    private void showErrorMessage(int imageView, String title, String message){

        if(errorlayout.getVisibility() == View.GONE){
            errorlayout.setVisibility(View.VISIBLE);
        }

        imgerror.setImageResource(imageView);
        errortitle.setText(title);
        errormessage.setText(message);

        btnretry.setOnClickListener(view -> ambilselesai(email));

    }
```

SERTIFIKAT

Nomor : 737/PRK/LB-TI/II.3.AU/D/2023

Diberikan Kepada :

Nurfadilah

NIM.220280030

Telah Menyelesaikan

PRAKTIKUM V

Laboratorium Teknik Informatika
Program Studi Teknik Informatika Fakultas Teknik
Universitas Muhammadiyah Parepare

Pada tanggal 18 September - 16 Desember 2023

Parepare, 03 Januari 2024

KETUA PRODI
TEKNIK INOFORMATIKA

Marlina, S.kom., M.kom
NBM. 1162 680

KEPALA
LABORATORIUM

Ir. Untung Suwardoyo, S.Kom., MT., IPP.
NBM. 1288 973

KARTU MONITORING BIMBINGAN

MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

HASIL

Mahasiswa : NURFADILAH	**Pembimbing I : Marlina, S.Kom.,M.Kom.**
NIM : 220280030	**Pembimbing II : Ir.Untung Suwardoyo, S.Kom.,M.T.,IPP**
Judul Skripsi : APLIKASI WEDDING ORGANIZER BERBASIS ANDROID	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 Perbaiki format penulisan		Konsultasi 1 Paket sewa	
Konsultasi 2		Konsultasi 2 Perjelas antara Paket dan Item	
Konsultasi 3		Konsultasi 3 Lengkapi data	
Konsultasi 4 Acc ujian Hasil		Konsultasi 4 Lengkapi datanya	
Konsultasi 5		Konsultasi 5 +Flowchart =) kompulan dan abstrak	

Lanjut ke halaman sebelah...

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton berwarna hijau muda dan dicetak timbal balik

SERTIFIKAT

Nomor :119/PRK/LB-TI/II.3.AU/D/2022

diberikan kepada :

NURFADILAH

NIM.220280030

telah menyelesaikan

PRAKTIKUM IV

Laboratorium Teknik Informatika
Program Studi Teknik Informatika Fakultas Teknik
Universitas Muhammadiyah Parepare

Pada tanggal 14 September 2022 - 25 Desember 2022

Parepare, 02 Januari 2023

KETUA PRODI
TEKNIK INFORMATIKA

MARLINA. S.Kom., M.Kom

KEPALA
LABORATORIUM

Ir. UNTUNG SUWARDOYO, S.Kom., MT.,IPP.

SERTIFIKAT

Nomor : 301/PRK/LB-TI/II.3.AU/D/2023

Diberikan Kepada :

Nurfadilah Thamrin

NIM.220280030

Telah Menyelesaikan

PRAKTIKUM III

Laboratorium Teknik Informatika
Program Studi Teknik Informatika Fakultas Teknik
Universitas Muhammadiyah Parepare

Pada tanggal 16 Januari 2023 - 29 Maret 2023

Parepare, 15 Juni 2023

KETUA PRODI
TEKNIK INOFORMATIKA

Marlina. S.kom., M.kom
NBM. 1162 680

KEPALA
LABORATORIUM

Ir. Untung Suwardoyo, S.Kom., MT., IPP.
NBM. 1288 973