Lampiran 1. Daftar Riwayat Hidup

1. Data Pribadi

Nama	: Sandri Alfarisi
Tempat/Tanggal Lahir	: Campang Tiga, 24 Juli 1998
Jenis Kelamin	: Laki-laki
Alamat Rumah/Kode Pos	: Perumahan Graha Cibadak blok A2/24 RT. 06/01 Desa Pasir Nangka Kec. Tigaraksa, Tangerang – Banten 15720
Agama	: Islam
Kebangsaan	: Indonesia
Status	: Belum Menikah
Telepon	: 081296595591
Email	: sandrialfarisi@gmail.com

2. Riwayat Pendidikan

Pendidikan Formal

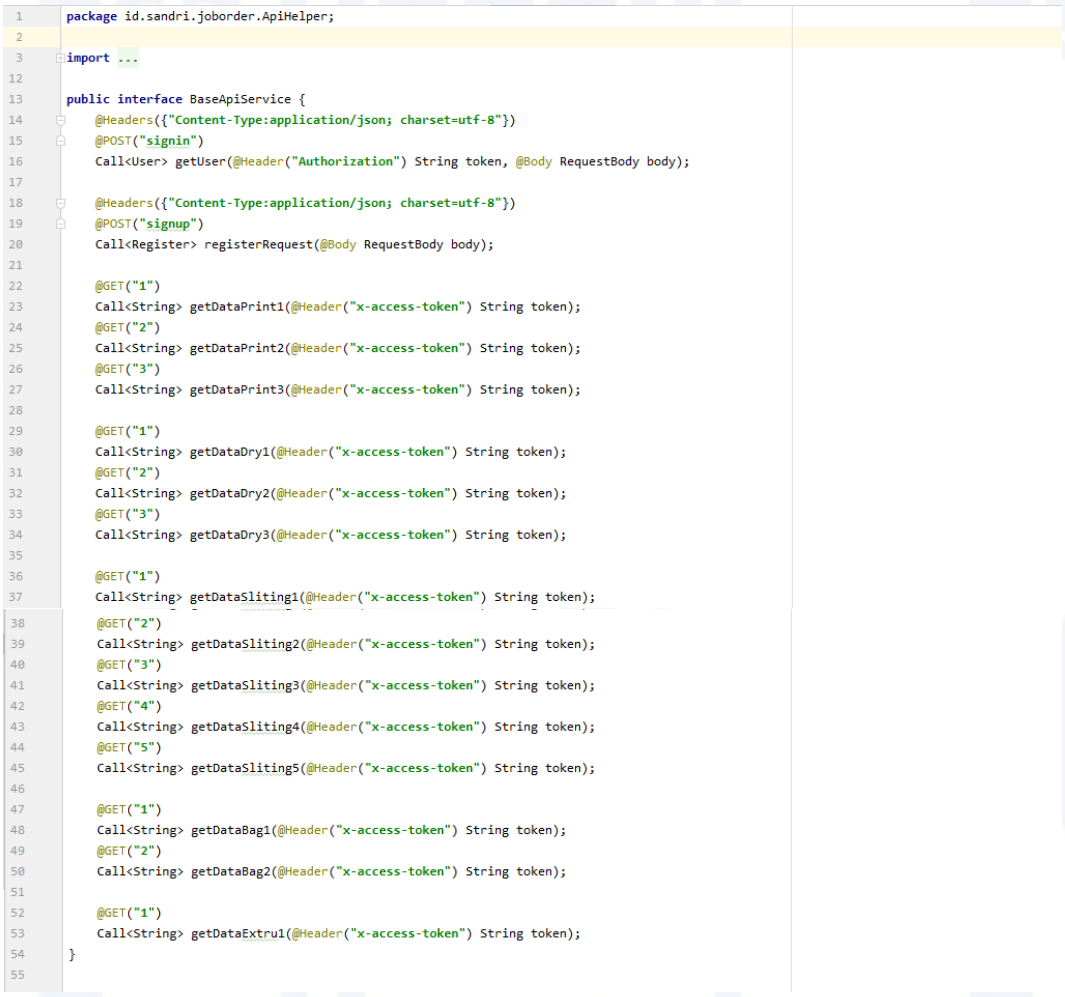
2004 – 2010	: SDN Sukanagara
2010 – 2013	: SMPN 2 Tigaraksa
2013 – 2016	: SMK Al-Fattah Tigaraksa
2016 – 2020	: Universitas Esa Unggul Jakarta (S1), Fakultas Ilmu Komputer, Program Studi Teknik Informatika

Lampiran 2. Hasil Wawancara

No	Pertanyaan	Jawaban	Tanggal
1.	Bagaimana proses bisnis yang berjalan saat ini, untuk melihat data jadwal produksi?	Masih menggunakan *file* Excel dan hasil *print out*	30/01/2019
2.	Aplikasi *Mobile* Android seperti apa yang diharapkan?	Yang dapat memberikan informasi secara cepat	30/01/2019
3.	Tampilan aplikasi yang seperti yang diharapkan?	Memiliki tampilan yang menarik dan enak dilihat	30/01/2019
4.	Fitur apa saja yang diperlukan pada aplikasi untuk saat ini?	Untuk saat ini fitur yang cukup dibutuhkan adalah Form Login, menu daftar mesin, daftar data jadwal produksi, dan detail data produksi saat jadwal produksi dipilih	30/01/2019

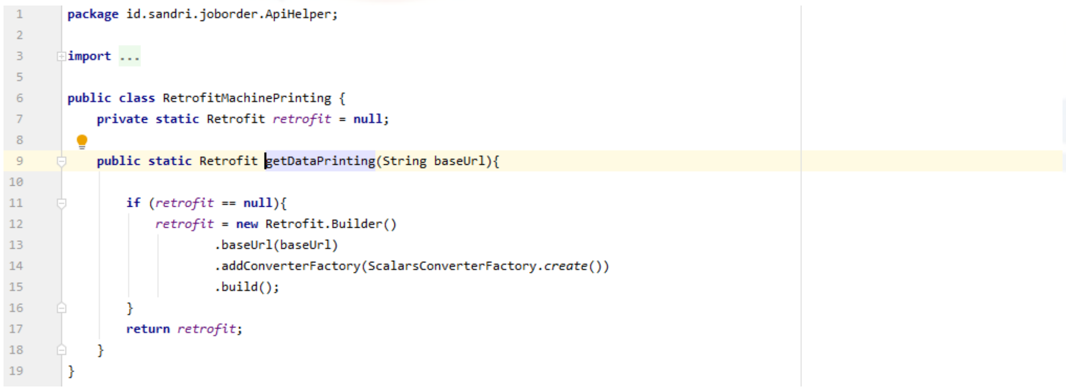
Lampiran 3. Source Code

1. API Helper
   BaseApiService.class



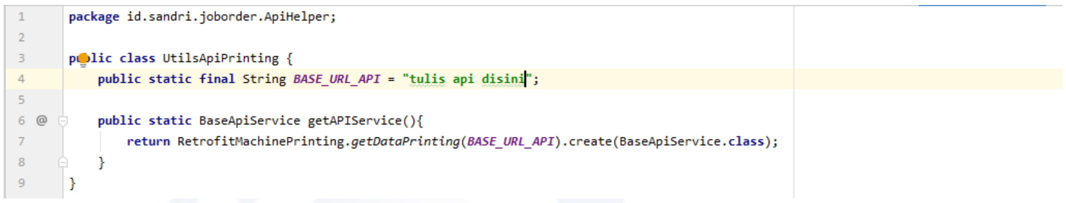
```java
package id.sandri.joborder.ApiHelper;

import ...

public interface BaseApiService {
    @Headers({"Content-Type:application/json; charset=utf-8"})
    @POST("signin")
    Call<User> getUser(@Header("Authorization") String token, @Body RequestBody body);

    @Headers({"Content-Type:application/json; charset=utf-8"})
    @POST("signup")
    Call<Register> registerRequest(@Body RequestBody body);

    @GET("1")
    Call<String> getDataPrint1(@Header("x-access-token") String token);
    @GET("2")
    Call<String> getDataPrint2(@Header("x-access-token") String token);
    @GET("3")
    Call<String> getDataPrint3(@Header("x-access-token") String token);

    @GET("1")
    Call<String> getDataDry1(@Header("x-access-token") String token);
    @GET("2")
    Call<String> getDataDry2(@Header("x-access-token") String token);
    @GET("3")
    Call<String> getDataDry3(@Header("x-access-token") String token);

    @GET("1")
    Call<String> getDataSliting1(@Header("x-access-token") String token);
    @GET("2")
    Call<String> getDataSliting2(@Header("x-access-token") String token);
    @GET("3")
    Call<String> getDataSliting3(@Header("x-access-token") String token);
    @GET("4")
    Call<String> getDataSliting4(@Header("x-access-token") String token);
    @GET("5")
    Call<String> getDataSliting5(@Header("x-access-token") String token);

    @GET("1")
    Call<String> getDataBag1(@Header("x-access-token") String token);
    @GET("2")
    Call<String> getDataBag2(@Header("x-access-token") String token);

    @GET("1")
    Call<String> getDataExtru1(@Header("x-access-token") String token);
}
```

Retrofit Machine



```java
package id.sandri.joborder.ApiHelper;

import ...

public class RetrofitMachinePrinting {
    private static Retrofit retrofit = null;

    public static Retrofit getDataPrinting(String baseUrl){

        if (retrofit == null){
            retrofit = new Retrofit.Builder()
                    .baseUrl(baseUrl)
                    .addConverterFactory(ScalarsConverterFactory.create())
                    .build();
        }
        return retrofit;
    }
}
```

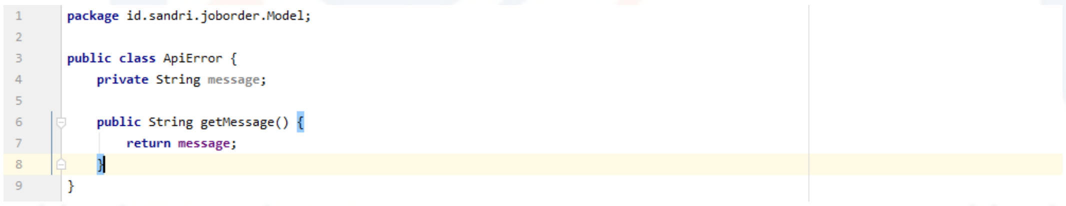
Utils Api



```java
package id.sandri.joborder.ApiHelper;

public class UtilsApiPrinting {
    public static final String BASE_URL_API = "tulis api disini";

    public static BaseApiService getAPIService(){
        return RetrofitMachinePrinting.getDataPrinting(BASE_URL_API).create(BaseApiService.class);
    }
}
```

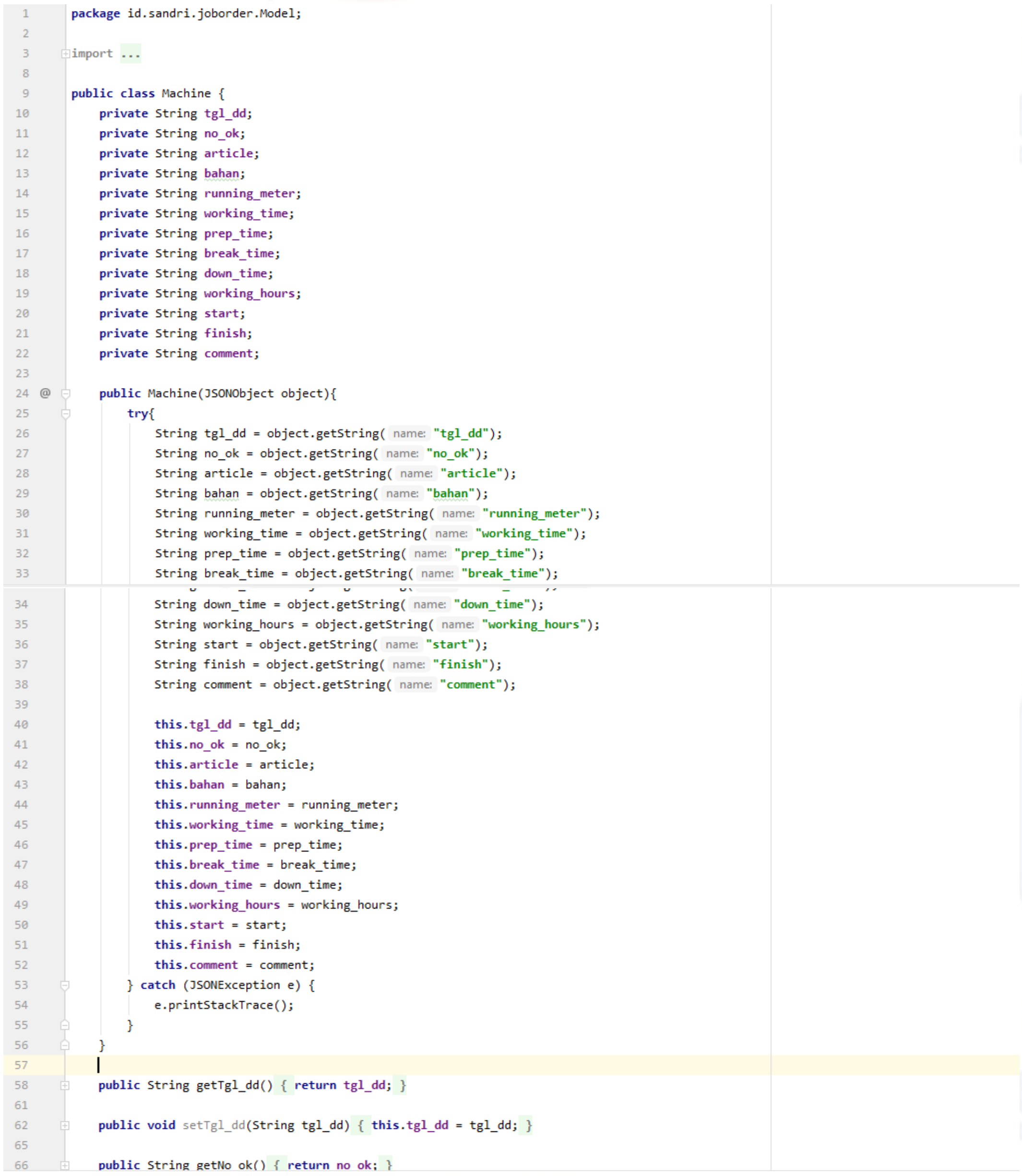
2. Class Model

Api Error

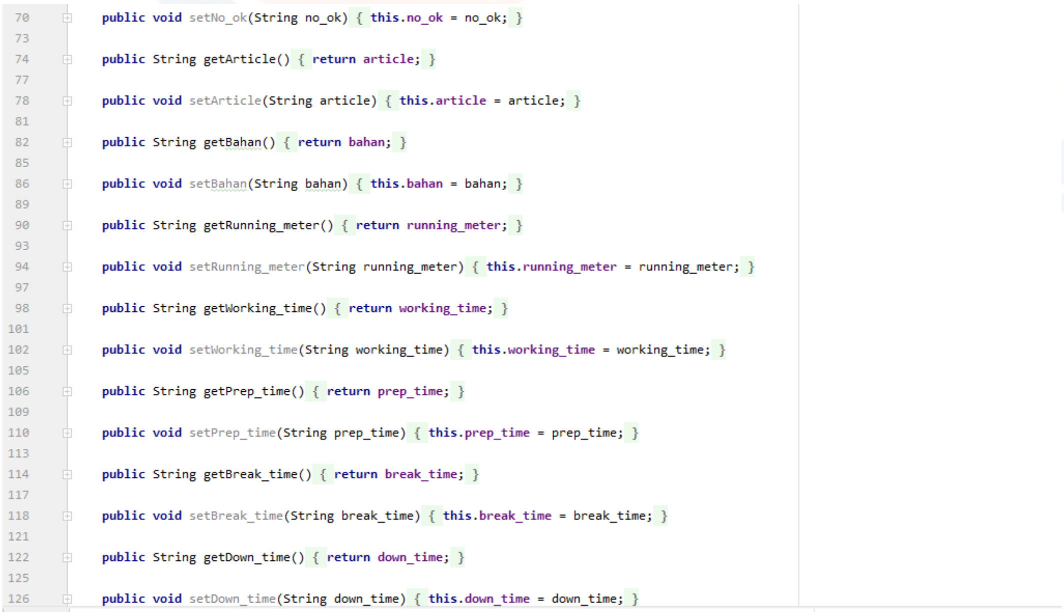


```java
package id.sandri.joborder.Model;

public class ApiError {
    private String message;

    public String getMessage() {
        return message;
    }
}
```

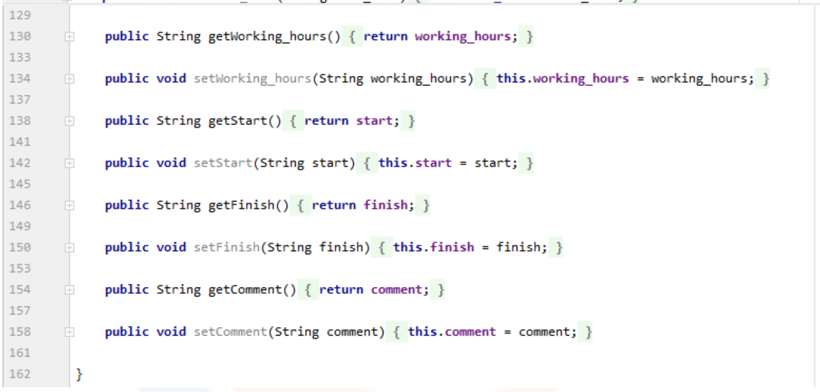
Machine



```java
package id.sandri.joborder.Model;

import ...

public class Machine {
    private String tgl_dd;
    private String no_ok;
    private String article;
    private String bahan;
    private String running_meter;
    private String working_time;
    private String prep_time;
    private String break_time;
    private String down_time;
    private String working_hours;
    private String start;
    private String finish;
    private String comment;

    public Machine(JSONObject object){
        try{
            String tgl_dd = object.getString( name: "tgl_dd");
            String no_ok = object.getString( name: "no_ok");
            String article = object.getString( name: "article");
            String bahan = object.getString( name: "bahan");
            String running_meter = object.getString( name: "running_meter");
            String working_time = object.getString( name: "working_time");
            String prep_time = object.getString( name: "prep_time");
            String break_time = object.getString( name: "break_time");
            String down_time = object.getString( name: "down_time");
            String working_hours = object.getString( name: "working_hours");
            String start = object.getString( name: "start");
            String finish = object.getString( name: "finish");
            String comment = object.getString( name: "comment");

            this.tgl_dd = tgl_dd;
            this.no_ok = no_ok;
            this.article = article;
            this.bahan = bahan;
            this.running_meter = running_meter;
            this.working_time = working_time;
            this.prep_time = prep_time;
            this.break_time = break_time;
            this.down_time = down_time;
            this.working_hours = working_hours;
            this.start = start;
            this.finish = finish;
            this.comment = comment;
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }

    public String getTgl_dd() { return tgl_dd; }

    public void setTgl_dd(String tgl_dd) { this.tgl_dd = tgl_dd; }

    public String getNo_ok() { return no_ok; }
```

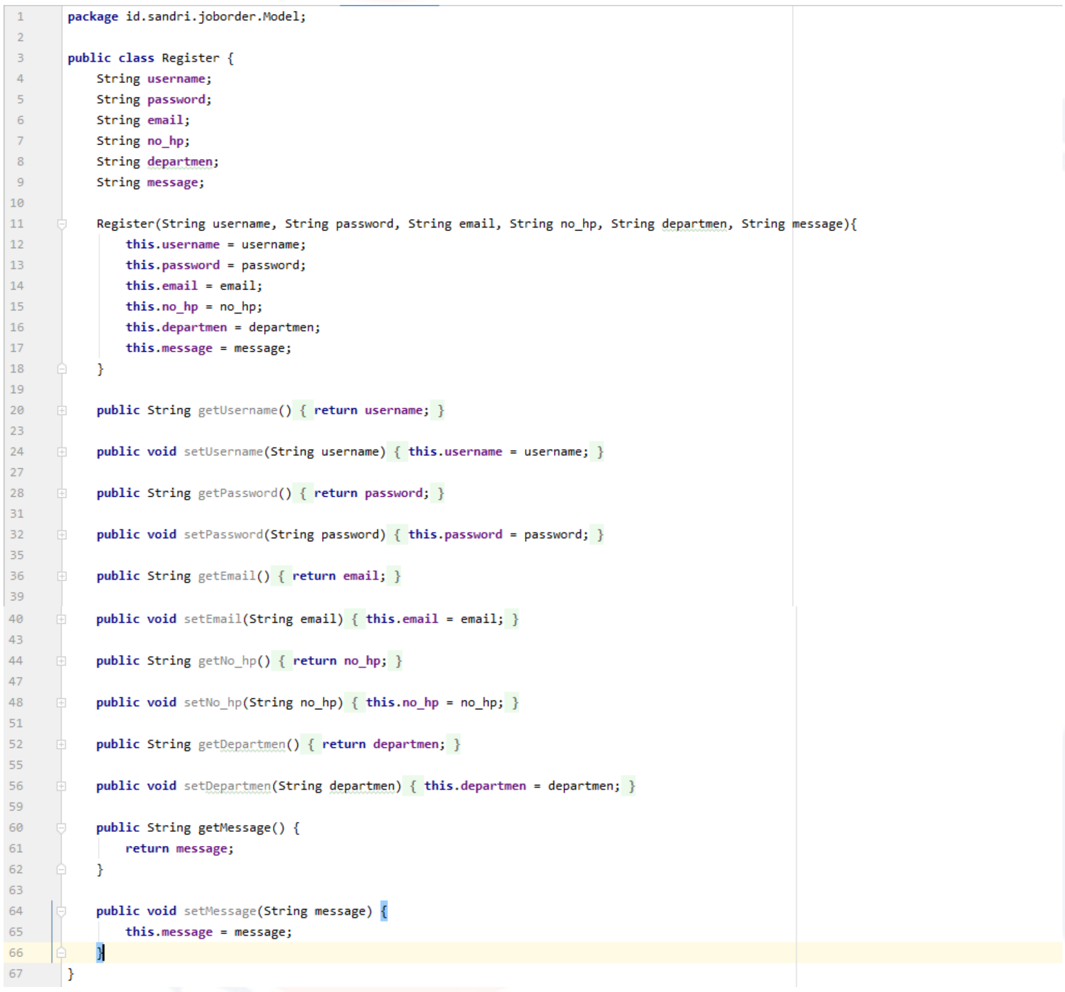


```java
    public void setNo_ok(String no_ok) { this.no_ok = no_ok; }

    public String getArticle() { return article; }

    public void setArticle(String article) { this.article = article; }

    public String getBahan() { return bahan; }

    public void setBahan(String bahan) { this.bahan = bahan; }

    public String getRunning_meter() { return running_meter; }

    public void setRunning_meter(String running_meter) { this.running_meter = running_meter; }

    public String getWorking_time() { return working_time; }

    public void setWorking_time(String working_time) { this.working_time = working_time; }

    public String getPrep_time() { return prep_time; }

    public void setPrep_time(String prep_time) { this.prep_time = prep_time; }

    public String getBreak_time() { return break_time; }

    public void setBreak_time(String break_time) { this.break_time = break_time; }

    public String getDown_time() { return down_time; }

    public void setDown_time(String down_time) { this.down_time = down_time; }
```



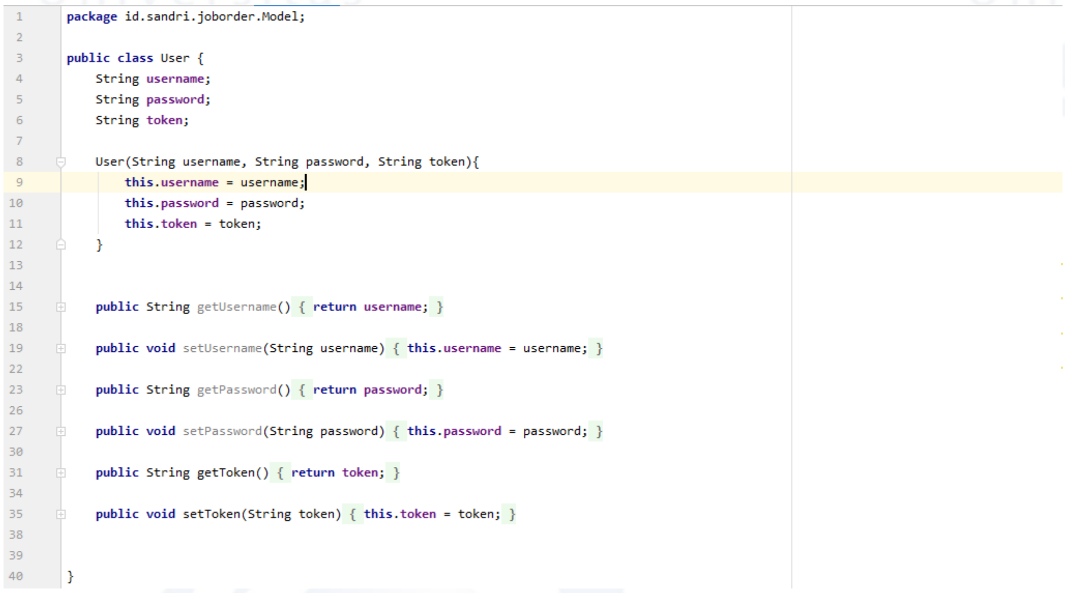
```java

    public String getWorking_hours() { return working_hours; }

    public void setWorking_hours(String working_hours) { this.working_hours = working_hours; }

    public String getStart() { return start; }

    public void setStart(String start) { this.start = start; }

    public String getFinish() { return finish; }

    public void setFinish(String finish) { this.finish = finish; }

    public String getComment() { return comment; }

    public void setComment(String comment) { this.comment = comment; }

}
```

Register



```java
package id.sandri.joborder.Model;

public class Register {
    String username;
    String password;
    String email;
    String no_hp;
    String departmen;
    String message;

    Register(String username, String password, String email, String no_hp, String departmen, String message){
        this.username = username;
        this.password = password;
        this.email = email;
        this.no_hp = no_hp;
        this.departmen = departmen;
        this.message = message;
    }

    public String getUsername() { return username; }

    public void setUsername(String username) { this.username = username; }

    public String getPassword() { return password; }

    public void setPassword(String password) { this.password = password; }

    public String getEmail() { return email; }

    public void setEmail(String email) { this.email = email; }

    public String getNo_hp() { return no_hp; }

    public void setNo_hp(String no_hp) { this.no_hp = no_hp; }

    public String getDepartmen() { return departmen; }

    public void setDepartmen(String departmen) { this.departmen = departmen; }

    public String getMessage() {
        return message;
    }

    public void setMessage(String message) {
        this.message = message;
    }
}
```

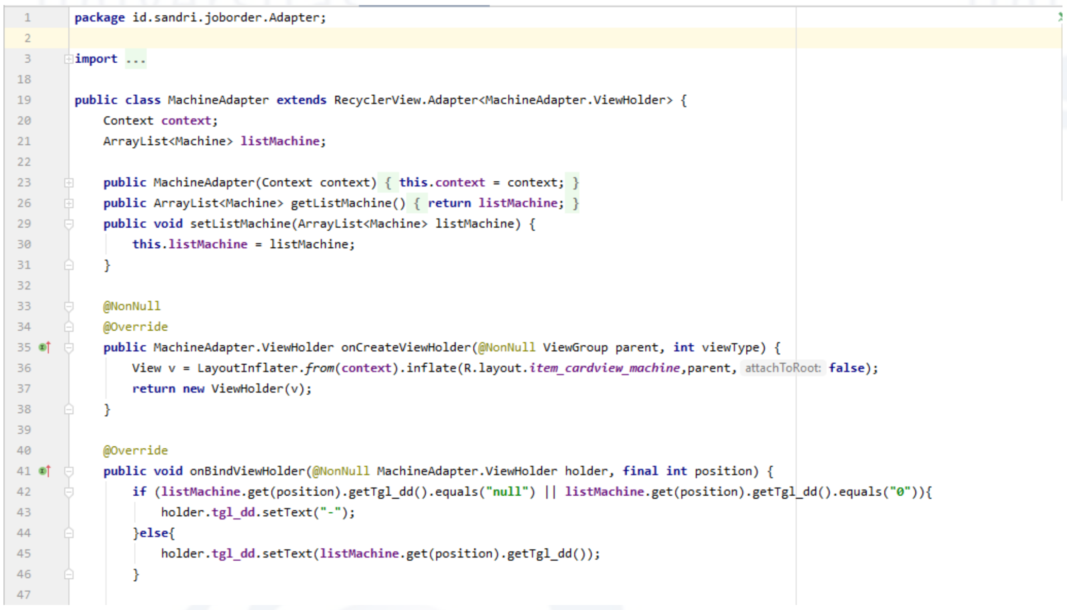
User



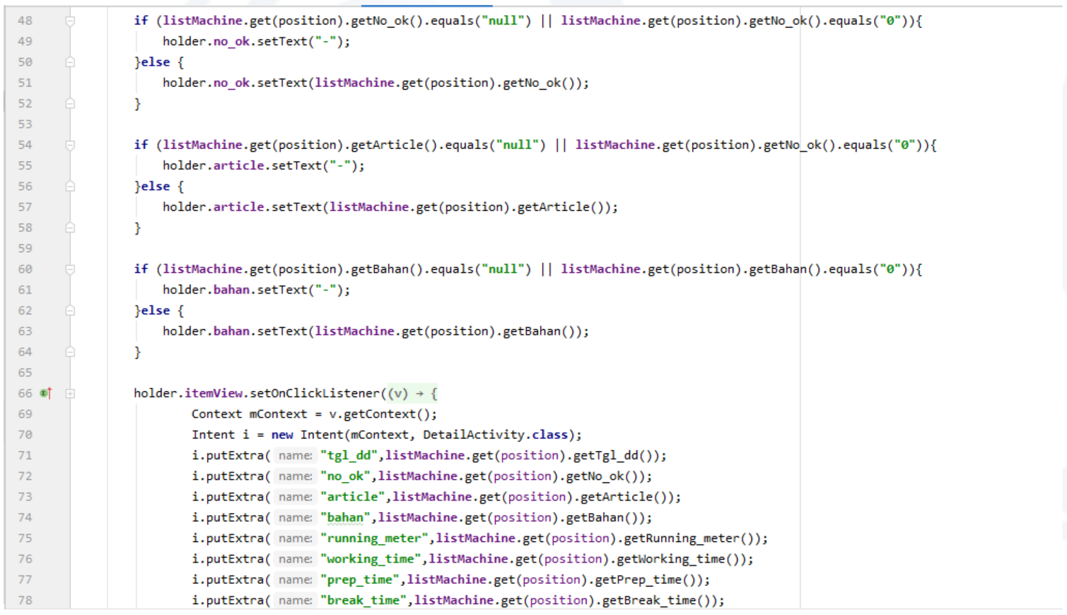
```java
package id.sandri.joborder.Model;

public class User {
    String username;
    String password;
    String token;

    User(String username, String password, String token){
        this.username = username;
        this.password = password;
        this.token = token;
    }


    public String getUsername() { return username; }

    public void setUsername(String username) { this.username = username; }

    public String getPassword() { return password; }

    public void setPassword(String password) { this.password = password; }

    public String getToken() { return token; }

    public void setToken(String token) { this.token = token; }


}
```

3. Adapter Class

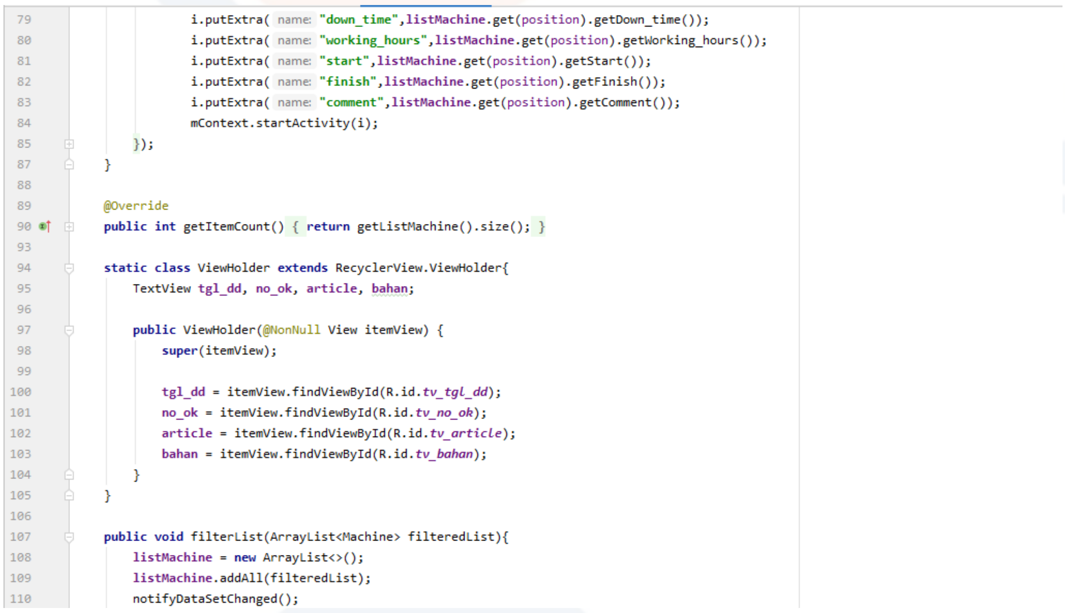
   Machine Adapter



```java
package id.sandri.joborder.Adapter;

import ...

public class MachineAdapter extends RecyclerView.Adapter<MachineAdapter.ViewHolder> {
    Context context;
    ArrayList<Machine> listMachine;

    public MachineAdapter(Context context) { this.context = context; }
    public ArrayList<Machine> getListMachine() { return listMachine; }
    public void setListMachine(ArrayList<Machine> listMachine) {
        this.listMachine = listMachine;
    }

    @NonNull
    @Override
    public MachineAdapter.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View v = LayoutInflater.from(context).inflate(R.layout.item_cardview_machine,parent, attachToRoot: false);
        return new ViewHolder(v);
    }

    @Override
    public void onBindViewHolder(@NonNull MachineAdapter.ViewHolder holder, final int position) {
        if (listMachine.get(position).getTgl_dd().equals("null") || listMachine.get(position).getTgl_dd().equals("0")){
            holder.tgl_dd.setText("-");
        }else{
            holder.tgl_dd.setText(listMachine.get(position).getTgl_dd());
        }

```

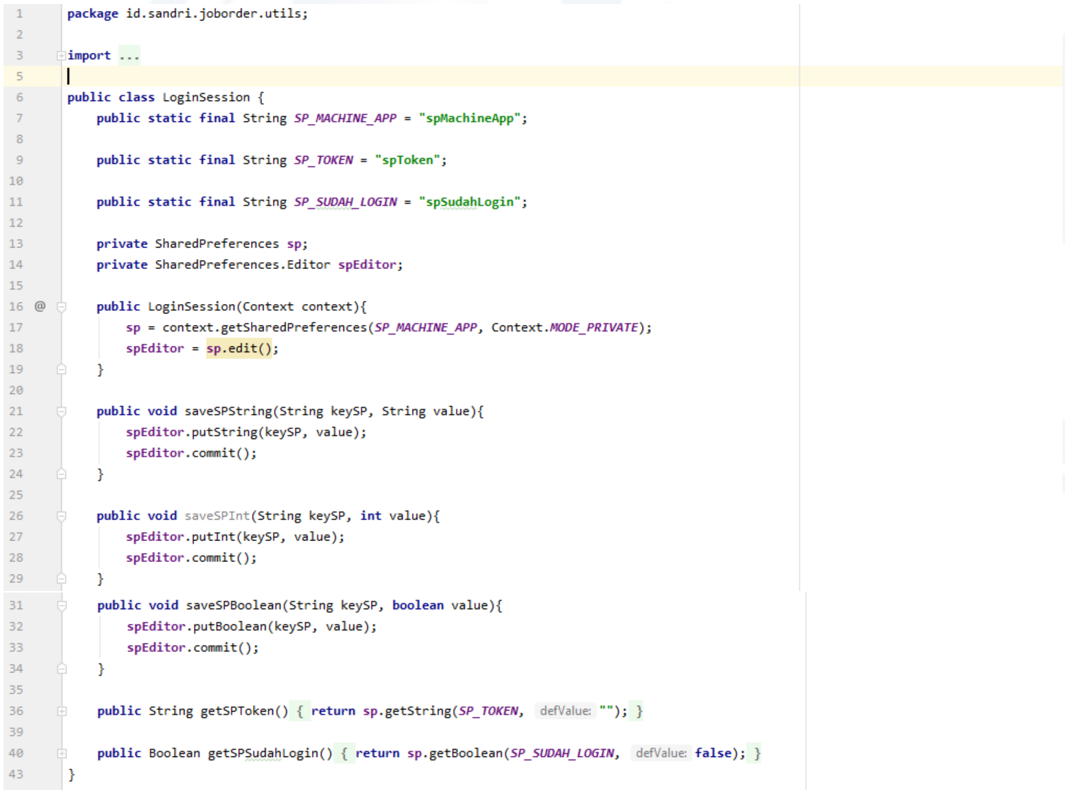


```java
        if (listMachine.get(position).getNo_ok().equals("null") || listMachine.get(position).getNo_ok().equals("0")){
            holder.no_ok.setText("-");
        }else {
            holder.no_ok.setText(listMachine.get(position).getNo_ok());
        }

        if (listMachine.get(position).getArticle().equals("null") || listMachine.get(position).getNo_ok().equals("0")){
            holder.article.setText("-");
        }else {
            holder.article.setText(listMachine.get(position).getArticle());
        }

        if (listMachine.get(position).getBahan().equals("null") || listMachine.get(position).getBahan().equals("0")){
            holder.bahan.setText("-");
        }else {
            holder.bahan.setText(listMachine.get(position).getBahan());
        }

        holder.itemView.setOnClickListener((v) → {
                Context mContext = v.getContext();
                Intent i = new Intent(mContext, DetailActivity.class);
                i.putExtra( name: "tgl_dd",listMachine.get(position).getTgl_dd());
                i.putExtra( name: "no_ok",listMachine.get(position).getNo_ok());
                i.putExtra( name: "article",listMachine.get(position).getArticle());
                i.putExtra( name: "bahan",listMachine.get(position).getBahan());
                i.putExtra( name: "running_meter",listMachine.get(position).getRunning_meter());
                i.putExtra( name: "working_time",listMachine.get(position).getWorking_time());
                i.putExtra( name: "prep_time",listMachine.get(position).getPrep_time());
                i.putExtra( name: "break_time",listMachine.get(position).getBreak_time());
```



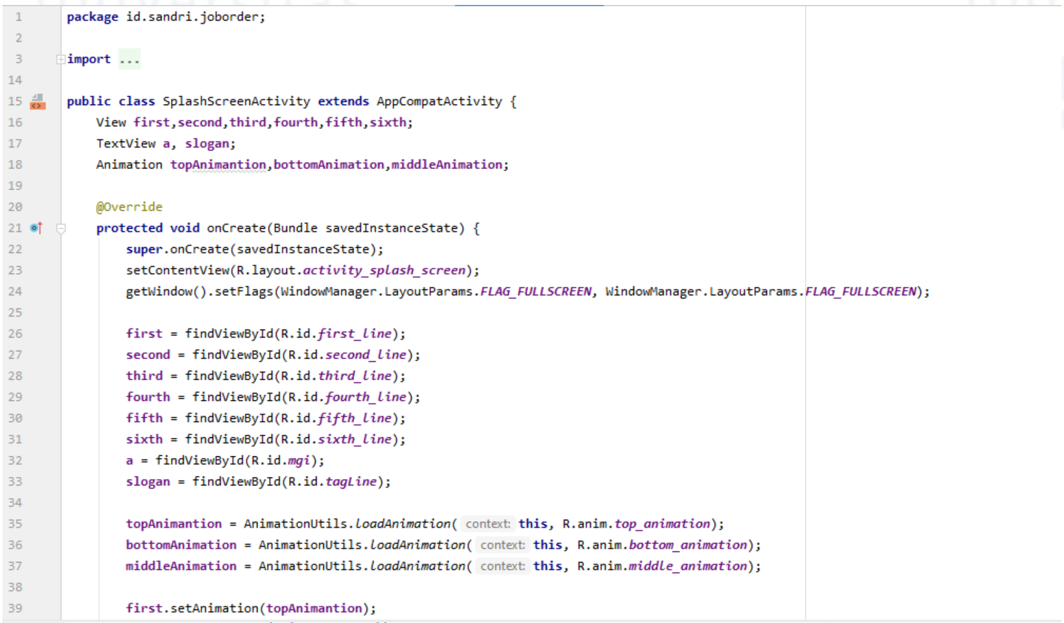
```java
                i.putExtra( name: "down_time",listMachine.get(position).getDown_time());
                i.putExtra( name: "working_hours",listMachine.get(position).getWorking_hours());
                i.putExtra( name: "start",listMachine.get(position).getStart());
                i.putExtra( name: "finish",listMachine.get(position).getFinish());
                i.putExtra( name: "comment",listMachine.get(position).getComment());
                mContext.startActivity(i);
        });
    }

    @Override
    public int getItemCount() { return getListMachine().size(); }

    static class ViewHolder extends RecyclerView.ViewHolder{
        TextView tgl_dd, no_ok, article, bahan;

        public ViewHolder(@NonNull View itemView) {
            super(itemView);

            tgl_dd = itemView.findViewById(R.id.tv_tgl_dd);
            no_ok = itemView.findViewById(R.id.tv_no_ok);
            article = itemView.findViewById(R.id.tv_article);
            bahan = itemView.findViewById(R.id.tv_bahan);
        }
    }

    public void filterList(ArrayList<Machine> filteredList){
        listMachine = new ArrayList<>();
        listMachine.addAll(filteredList);
        notifyDataSetChanged();
```

4. Login Session Utils
   Login Session

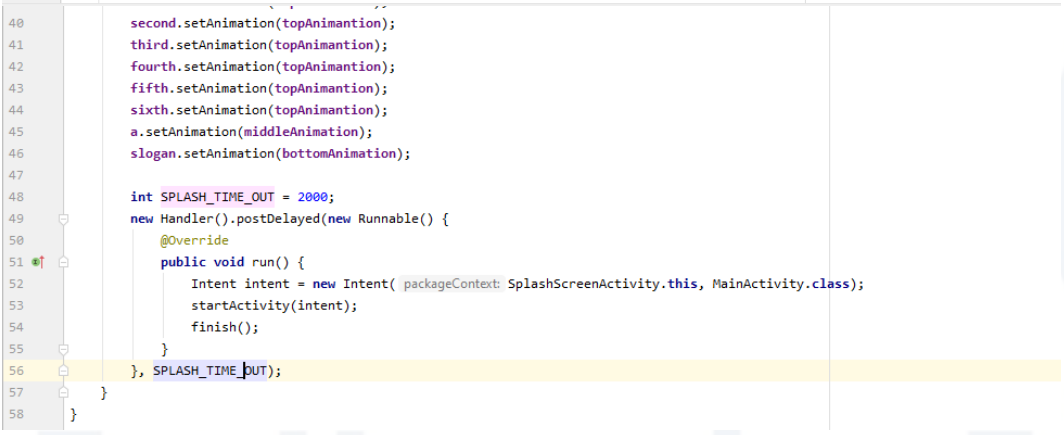


```java
package id.sandri.joborder.utils;

import ...

public class LoginSession {
    public static final String SP_MACHINE_APP = "spMachineApp";

    public static final String SP_TOKEN = "spToken";

    public static final String SP_SUDAH_LOGIN = "spSudahLogin";

    private SharedPreferences sp;
    private SharedPreferences.Editor spEditor;

    public LoginSession(Context context){
        sp = context.getSharedPreferences(SP_MACHINE_APP, Context.MODE_PRIVATE);
        spEditor = sp.edit();
    }

    public void saveSPString(String keySP, String value){
        spEditor.putString(keySP, value);
        spEditor.commit();
    }

    public void saveSPInt(String keySP, int value){
        spEditor.putInt(keySP, value);
        spEditor.commit();
    }
    public void saveSPBoolean(String keySP, boolean value){
        spEditor.putBoolean(keySP, value);
        spEditor.commit();
    }

    public String getSPToken() { return sp.getString(SP_TOKEN, defValue: ""); }

    public Boolean getSPSudahLogin() { return sp.getBoolean(SP_SUDAH_LOGIN, defValue: false); }
}
```

5. Splash Screen
   Splash Screen Activity

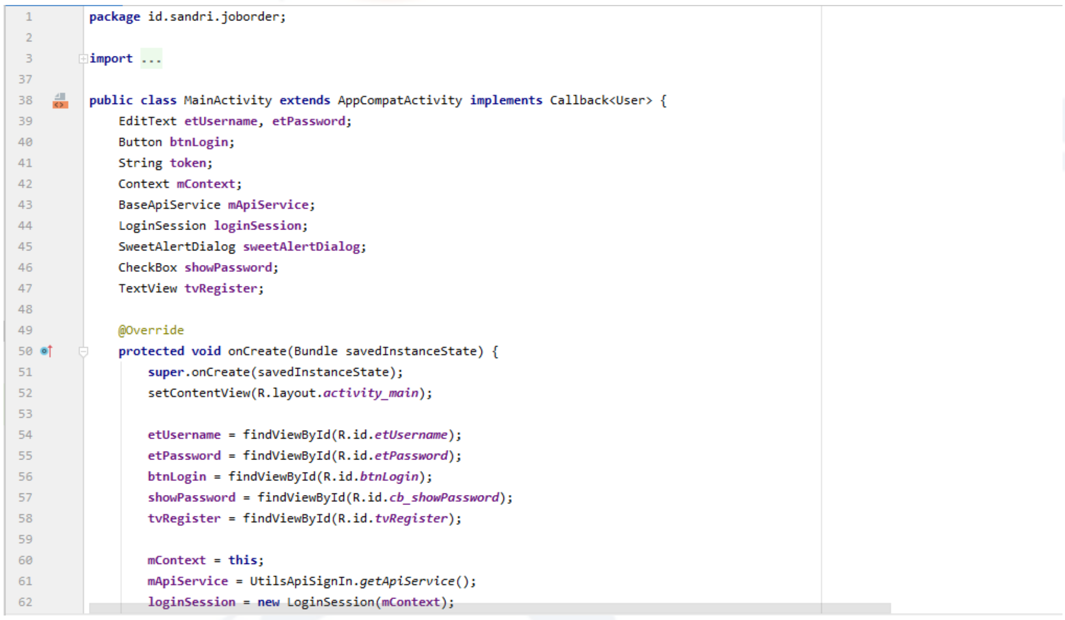


```java
package id.sandri.joborder;

import ...

public class SplashScreenActivity extends AppCompatActivity {
    View first,second,third,fourth,fifth,sixth;
    TextView a, slogan;
    Animation topAnimantion,bottomAnimation,middleAnimation;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash_screen);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);

        first = findViewById(R.id.first_line);
        second = findViewById(R.id.second_line);
        third = findViewById(R.id.third_line);
        fourth = findViewById(R.id.fourth_line);
        fifth = findViewById(R.id.fifth_line);
        sixth = findViewById(R.id.sixth_line);
        a = findViewById(R.id.mgi);
        slogan = findViewById(R.id.tagLine);

        topAnimantion = AnimationUtils.loadAnimation( context: this, R.anim.top_animation);
        bottomAnimation = AnimationUtils.loadAnimation( context: this, R.anim.bottom_animation);
        middleAnimation = AnimationUtils.loadAnimation( context: this, R.anim.middle_animation);

        first.setAnimation(topAnimantion);
```

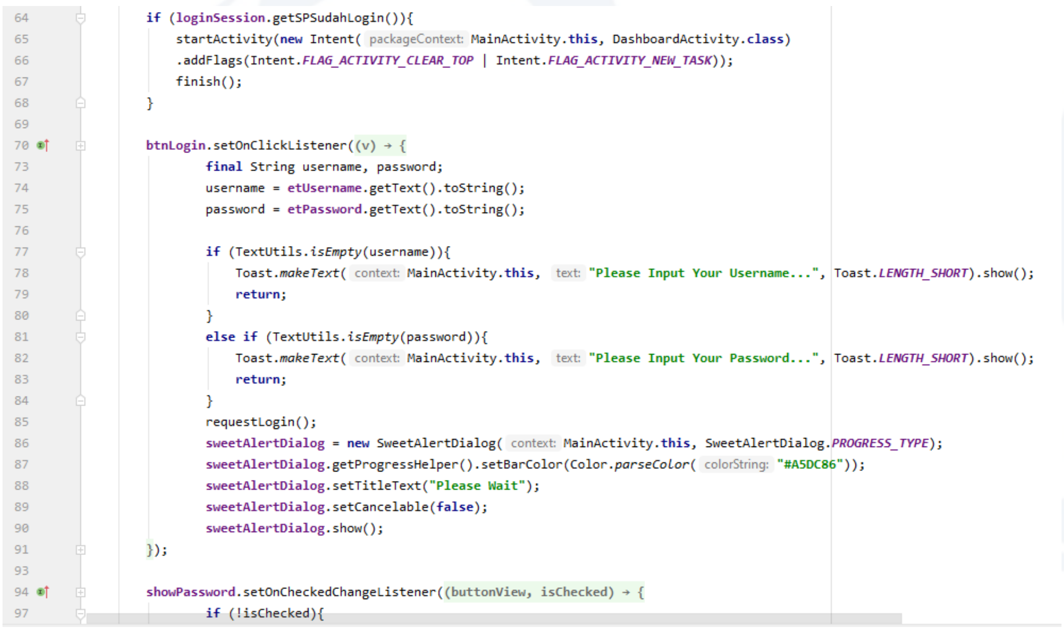


```java
        second.setAnimation(topAnimantion);
        third.setAnimation(topAnimantion);
        fourth.setAnimation(topAnimantion);
        fifth.setAnimation(topAnimantion);
        sixth.setAnimation(topAnimantion);
        a.setAnimation(middleAnimation);
        slogan.setAnimation(bottomAnimation);

        int SPLASH_TIME_OUT = 2000;
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                Intent intent = new Intent( packageContext: SplashScreenActivity.this, MainActivity.class);
                startActivity(intent);
                finish();
            }
        }, SPLASH_TIME_OUT);
    }
}
```

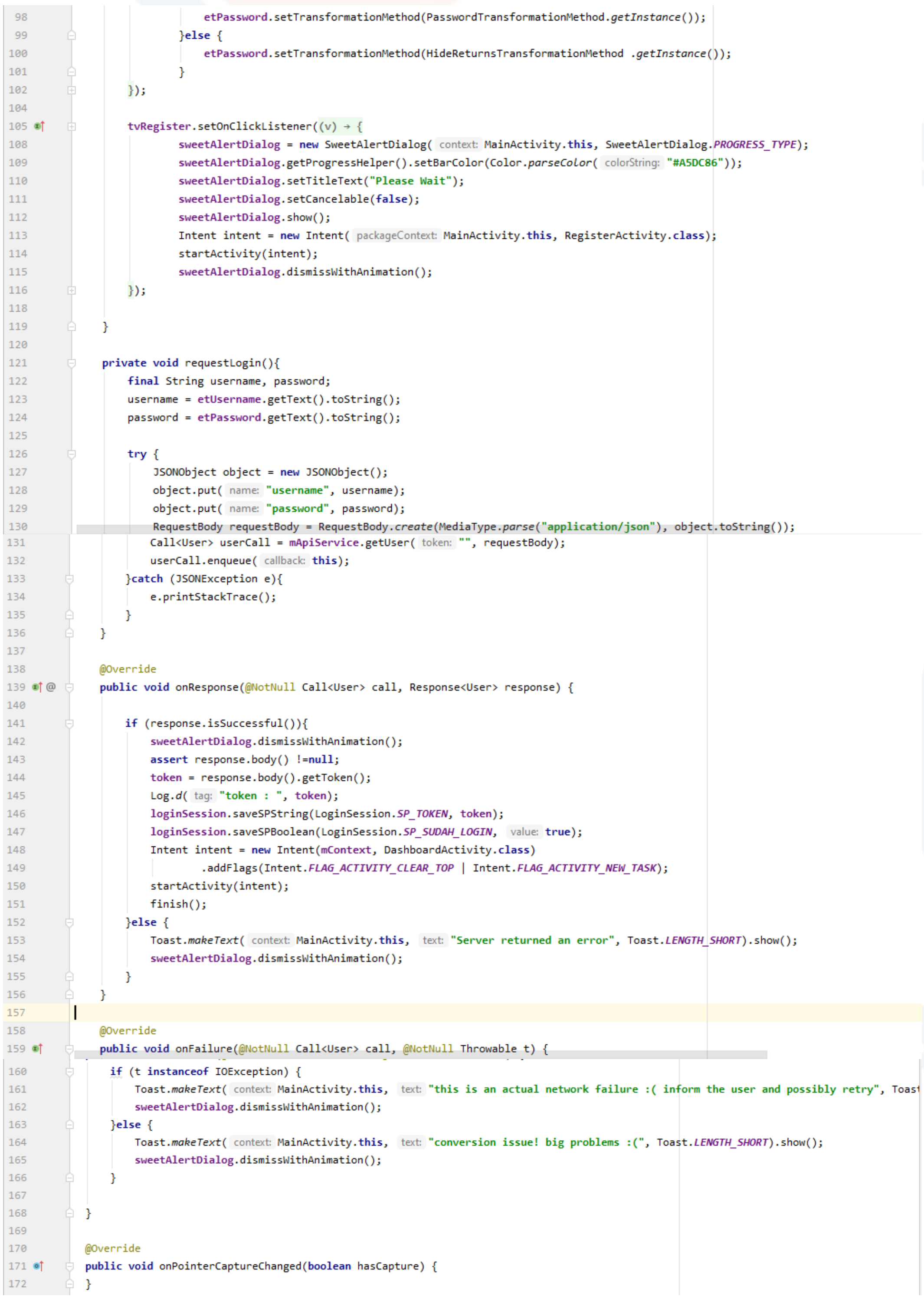
6. Login Activity



```java
package id.sandri.joborder;

import ...

public class MainActivity extends AppCompatActivity implements Callback<User> {
    EditText etUsername, etPassword;
    Button btnLogin;
    String token;
    Context mContext;
    BaseApiService mApiService;
    LoginSession loginSession;
    SweetAlertDialog sweetAlertDialog;
    CheckBox showPassword;
    TextView tvRegister;

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

        etUsername = findViewById(R.id.etUsername);
        etPassword = findViewById(R.id.etPassword);
        btnLogin = findViewById(R.id.btnLogin);
        showPassword = findViewById(R.id.cb_showPassword);
        tvRegister = findViewById(R.id.tvRegister);

        mContext = this;
        mApiService = UtilsApiSignIn.getApiService();
        loginSession = new LoginSession(mContext);
```

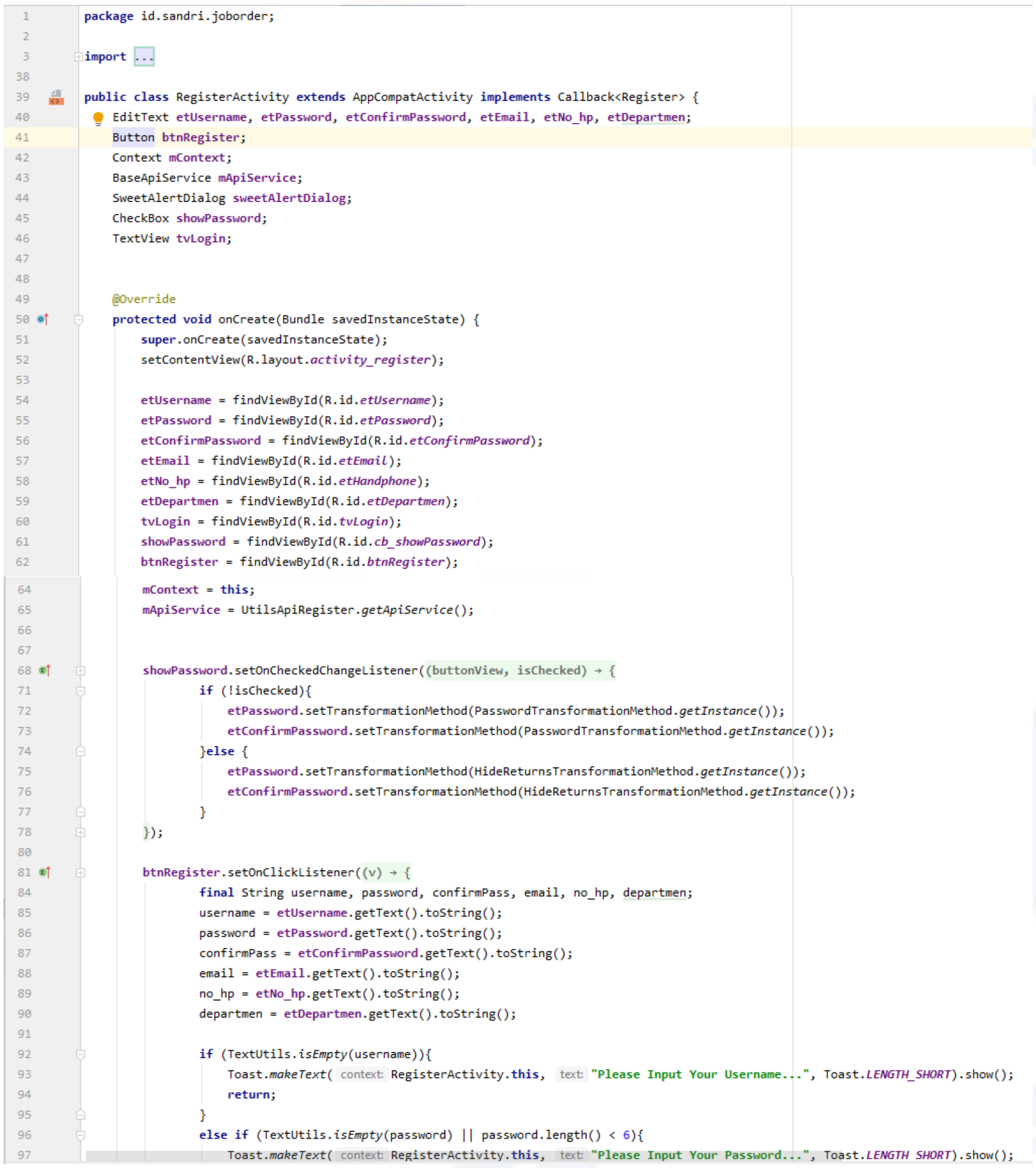


```java
        if (loginSession.getSPSudahLogin()){
            startActivity(new Intent( packageContext: MainActivity.this, DashboardActivity.class)
            .addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP | Intent.FLAG_ACTIVITY_NEW_TASK));
            finish();
        }

        btnLogin.setOnClickListener((v) → {
                final String username, password;
                username = etUsername.getText().toString();
                password = etPassword.getText().toString();

                if (TextUtils.isEmpty(username)){
                    Toast.makeText( context: MainActivity.this,  text: "Please Input Your Username...", Toast.LENGTH_SHORT).show();
                    return;
                }
                else if (TextUtils.isEmpty(password)){
                    Toast.makeText( context: MainActivity.this,  text: "Please Input Your Password...", Toast.LENGTH_SHORT).show();
                    return;
                }
                requestLogin();
                sweetAlertDialog = new SweetAlertDialog( context: MainActivity.this, SweetAlertDialog.PROGRESS_TYPE);
                sweetAlertDialog.getProgressHelper().setBarColor(Color.parseColor( colorString: "#A5DC86"));
                sweetAlertDialog.setTitleText("Please Wait");
                sweetAlertDialog.setCancelable(false);
                sweetAlertDialog.show();
        });

        showPassword.setOnCheckedChangeListener((buttonView, isChecked) → {
                if (!isChecked){
```

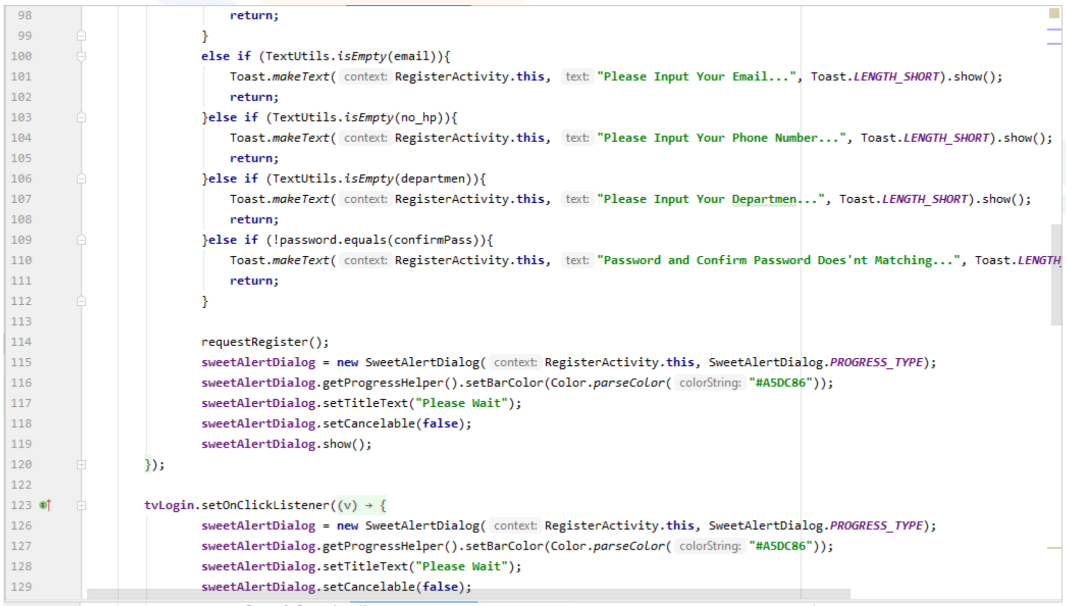


```java
                etPassword.setTransformationMethod(PasswordTransformationMethod.getInstance());
            }else {
                etPassword.setTransformationMethod(HideReturnsTransformationMethod .getInstance());
            }
        });

        tvRegister.setOnClickListener((v) → {
                sweetAlertDialog = new SweetAlertDialog( context: MainActivity.this, SweetAlertDialog.PROGRESS_TYPE);
                sweetAlertDialog.getProgressHelper().setBarColor(Color.parseColor( colorString: "#A5DC86"));
                sweetAlertDialog.setTitleText("Please Wait");
                sweetAlertDialog.setCancelable(false);
                sweetAlertDialog.show();
                Intent intent = new Intent( packageContext: MainActivity.this, RegisterActivity.class);
                startActivity(intent);
                sweetAlertDialog.dismissWithAnimation();
        });

    }

    private void requestLogin(){
        final String username, password;
        username = etUsername.getText().toString();
        password = etPassword.getText().toString();

        try {
            JSONObject object = new JSONObject();
            object.put( name: "username", username);
            object.put( name: "password", password);
            RequestBody requestBody = RequestBody.create(MediaType.parse("application/json"), object.toString());
            Call<User> userCall = mApiService.getUser( token: "", requestBody);
            userCall.enqueue( callback: this);
        }catch (JSONException e){
            e.printStackTrace();
        }
    }

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

        if (response.isSuccessful()){
            sweetAlertDialog.dismissWithAnimation();
            assert response.body() !=null;
            token = response.body().getToken();
            Log.d( tag: "token : ", token);
            loginSession.saveSPString(LoginSession.SP_TOKEN, token);
            loginSession.saveSPBoolean(LoginSession.SP_SUDAH_LOGIN,  value: true);
            Intent intent = new Intent(mContext, DashboardActivity.class)
                    .addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP | Intent.FLAG_ACTIVITY_NEW_TASK);
            startActivity(intent);
            finish();
        }else {
            Toast.makeText( context: MainActivity.this,  text: "Server returned an error", Toast.LENGTH_SHORT).show();
            sweetAlertDialog.dismissWithAnimation();
        }
    }

    @Override
    public void onFailure(@NotNull Call<User> call, @NotNull Throwable t) {
        if (t instanceof IOException) {
            Toast.makeText( context: MainActivity.this,  text: "this is an actual network failure :( inform the user and possibly retry", Toast
            sweetAlertDialog.dismissWithAnimation();
        }else {
            Toast.makeText( context: MainActivity.this,  text: "conversion issue! big problems :(", Toast.LENGTH_SHORT).show();
            sweetAlertDialog.dismissWithAnimation();
        }

    }

    @Override
    public void onPointerCaptureChanged(boolean hasCapture) {
    }
```

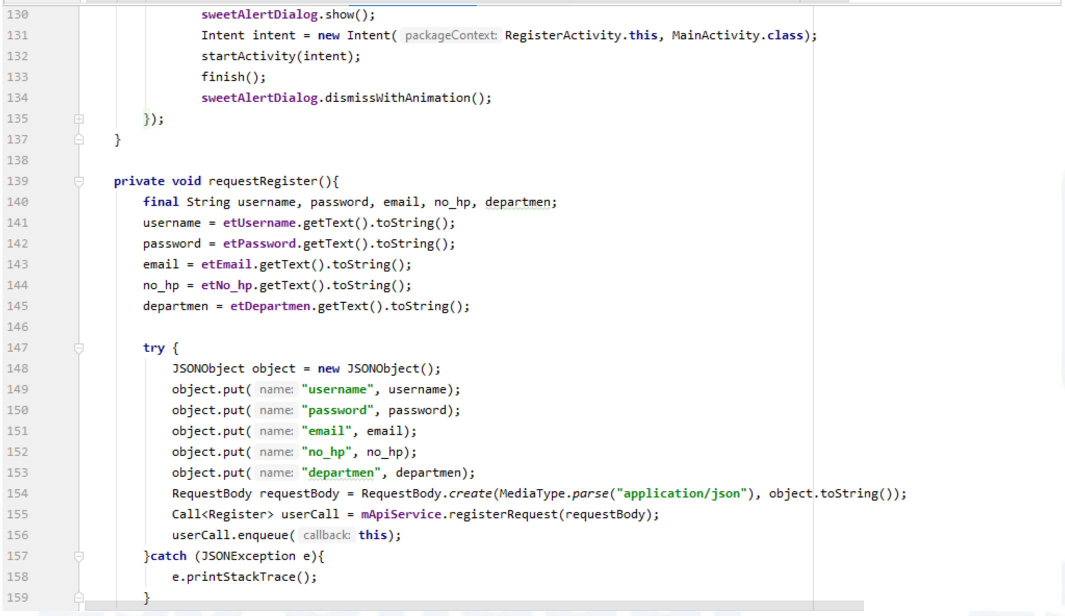
7. Register Activity



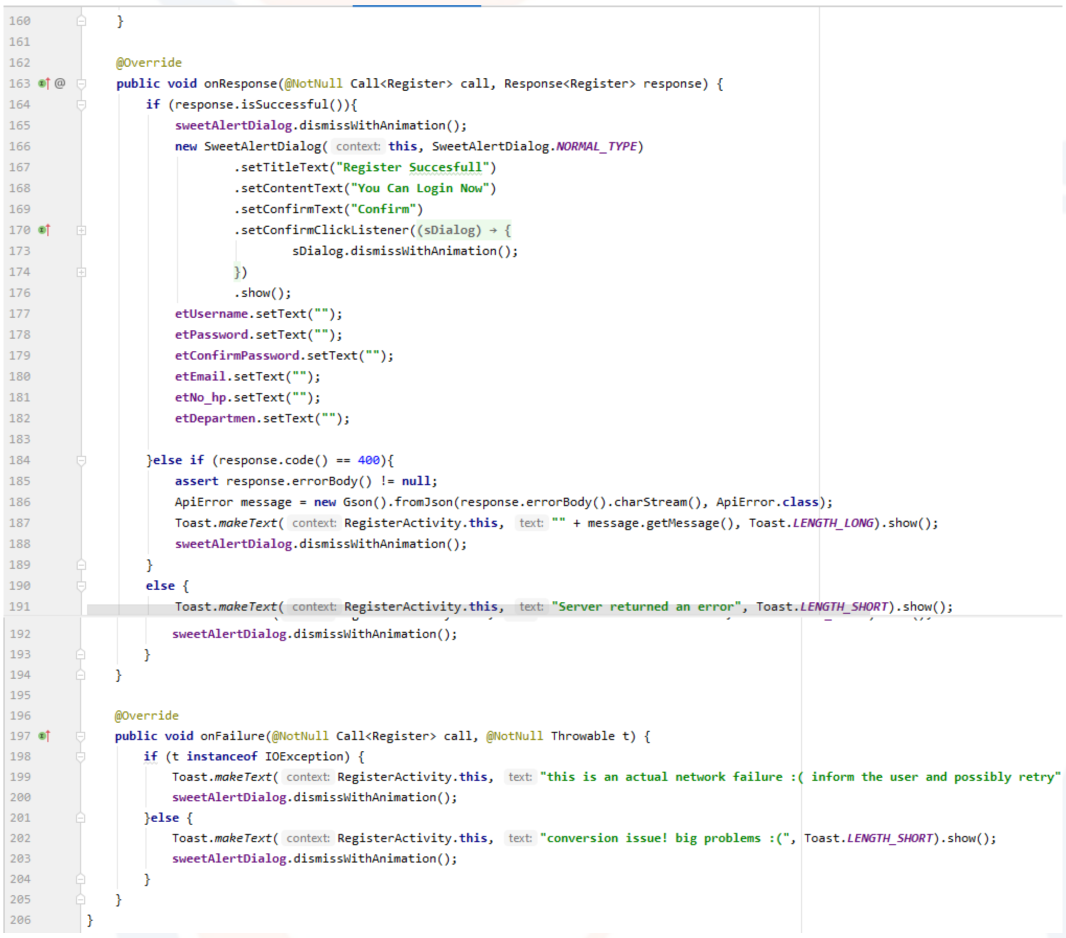
```
package id.sandri.joborder;

import ...

public class RegisterActivity extends AppCompatActivity implements Callback<Register> {
    EditText etUsername, etPassword, etConfirmPassword, etEmail, etNo_hp, etDepartmen;
    Button btnRegister;
    Context mContext;
    BaseApiService mApiService;
    SweetAlertDialog sweetAlertDialog;
    CheckBox showPassword;
    TextView tvLogin;


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

        etUsername = findViewById(R.id.etUsername);
        etPassword = findViewById(R.id.etPassword);
        etConfirmPassword = findViewById(R.id.etConfirmPassword);
        etEmail = findViewById(R.id.etEmail);
        etNo_hp = findViewById(R.id.etHandphone);
        etDepartmen = findViewById(R.id.etDepartmen);
        tvLogin = findViewById(R.id.tvLogin);
        showPassword = findViewById(R.id.cb_showPassword);
        btnRegister = findViewById(R.id.btnRegister);
        mContext = this;
        mApiService = UtilsApiRegister.getApiService();


        showPassword.setOnCheckedChangeListener((buttonView, isChecked) → {
                if (!isChecked){
                    etPassword.setTransformationMethod(PasswordTransformationMethod.getInstance());
                    etConfirmPassword.setTransformationMethod(PasswordTransformationMethod.getInstance());
                }else {
                    etPassword.setTransformationMethod(HideReturnsTransformationMethod.getInstance());
                    etConfirmPassword.setTransformationMethod(HideReturnsTransformationMethod.getInstance());
                }
        });

        btnRegister.setOnClickListener((v) → {
                final String username, password, confirmPass, email, no_hp, departmen;
                username = etUsername.getText().toString();
                password = etPassword.getText().toString();
                confirmPass = etConfirmPassword.getText().toString();
                email = etEmail.getText().toString();
                no_hp = etNo_hp.getText().toString();
                departmen = etDepartmen.getText().toString();

                if (TextUtils.isEmpty(username)){
                    Toast.makeText( context: RegisterActivity.this,  text: "Please Input Your Username...", Toast.LENGTH_SHORT).show();
                    return;
                }
                else if (TextUtils.isEmpty(password) || password.length() < 6){
                    Toast.makeText( context: RegisterActivity.this,  text: "Please Input Your Password...", Toast.LENGTH_SHORT).show();
```



```
                    return;
                }
                else if (TextUtils.isEmpty(email)){
                    Toast.makeText( context: RegisterActivity.this,  text: "Please Input Your Email...", Toast.LENGTH_SHORT).show();
                    return;
                }else if (TextUtils.isEmpty(no_hp)){
                    Toast.makeText( context: RegisterActivity.this,  text: "Please Input Your Phone Number...", Toast.LENGTH_SHORT).show();
                    return;
                }else if (TextUtils.isEmpty(departmen)){
                    Toast.makeText( context: RegisterActivity.this,  text: "Please Input Your Departmen...", Toast.LENGTH_SHORT).show();
                    return;
                }else if (!password.equals(confirmPass)){
                    Toast.makeText( context: RegisterActivity.this,  text: "Password and Confirm Password Does'nt Matching...", Toast.LENGTH
                    return;
                }

                requestRegister();
                sweetAlertDialog = new SweetAlertDialog( context: RegisterActivity.this, SweetAlertDialog.PROGRESS_TYPE);
                sweetAlertDialog.getProgressHelper().setBarColor(Color.parseColor( colorString: "#A5DC86"));
                sweetAlertDialog.setTitleText("Please Wait");
                sweetAlertDialog.setCancelable(false);
                sweetAlertDialog.show();
        });

        tvLogin.setOnClickListener((v) → {
                sweetAlertDialog = new SweetAlertDialog( context: RegisterActivity.this, SweetAlertDialog.PROGRESS_TYPE);
                sweetAlertDialog.getProgressHelper().setBarColor(Color.parseColor( colorString: "#A5DC86"));
                sweetAlertDialog.setTitleText("Please Wait");
                sweetAlertDialog.setCancelable(false);
```

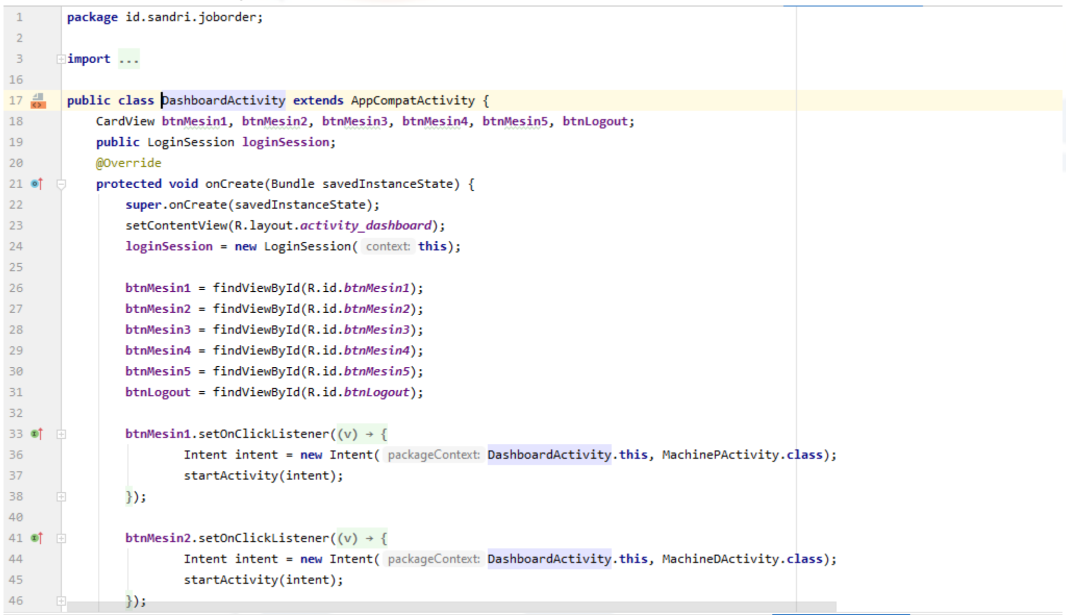


```
                sweetAlertDialog.show();
                Intent intent = new Intent( packageContext: RegisterActivity.this, MainActivity.class);
                startActivity(intent);
                finish();
                sweetAlertDialog.dismissWithAnimation();
        });
    }

    private void requestRegister(){
        final String username, password, email, no_hp, departmen;
        username = etUsername.getText().toString();
        password = etPassword.getText().toString();
        email = etEmail.getText().toString();
        no_hp = etNo_hp.getText().toString();
        departmen = etDepartmen.getText().toString();

        try {
            JSONObject object = new JSONObject();
            object.put( name: "username", username);
            object.put( name: "password", password);
            object.put( name: "email", email);
            object.put( name: "no_hp", no_hp);
            object.put( name: "departmen", departmen);
            RequestBody requestBody = RequestBody.create(MediaType.parse("application/json"), object.toString());
            Call<Register> userCall = mApiService.registerRequest(requestBody);
            userCall.enqueue( callback: this);
        }catch (JSONException e){
            e.printStackTrace();
        }
```

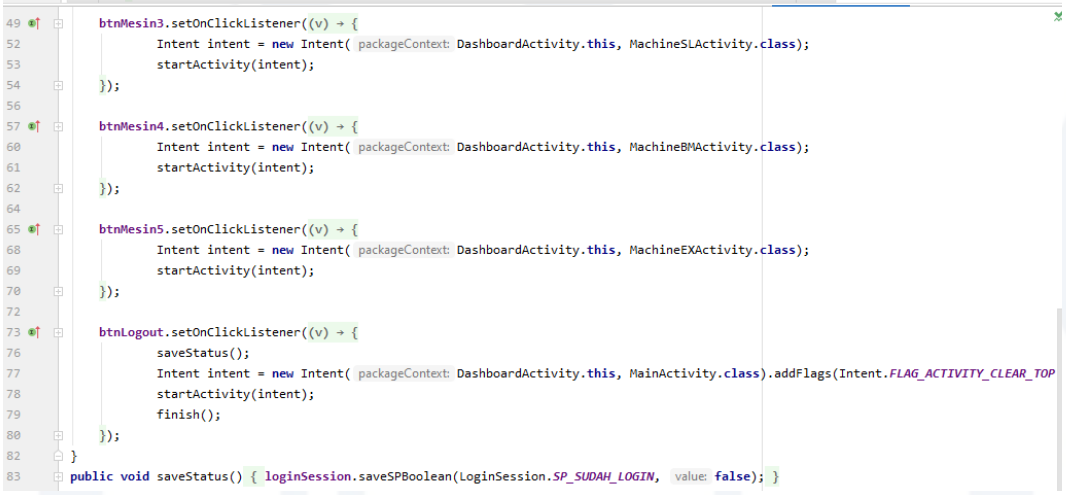


```
    }

    @Override
    public void onResponse(@NotNull Call<Register> call, Response<Register> response) {
        if (response.isSuccessful()){
            sweetAlertDialog.dismissWithAnimation();
            new SweetAlertDialog( context: this, SweetAlertDialog.NORMAL_TYPE)
                    .setTitleText("Register Succesfull")
                    .setContentText("You Can Login Now")
                    .setConfirmText("Confirm")
                    .setConfirmClickListener((sDialog) → {
                            sDialog.dismissWithAnimation();
                    })
                    .show();
            etUsername.setText("");
            etPassword.setText("");
            etConfirmPassword.setText("");
            etEmail.setText("");
            etNo_hp.setText("");
            etDepartmen.setText("");

        }else if (response.code() == 400){
            assert response.errorBody() != null;
            ApiError message = new Gson().fromJson(response.errorBody().charStream(), ApiError.class);
            Toast.makeText( context: RegisterActivity.this,  text: "" + message.getMessage(), Toast.LENGTH_LONG).show();
            sweetAlertDialog.dismissWithAnimation();
        }
        else {
            Toast.makeText( context: RegisterActivity.this,  text: "Server returned an error", Toast.LENGTH_SHORT).show();
            sweetAlertDialog.dismissWithAnimation();
        }
    }

    @Override
    public void onFailure(@NotNull Call<Register> call, @NotNull Throwable t) {
        if (t instanceof IOException) {
            Toast.makeText( context: RegisterActivity.this,  text: "this is an actual network failure :( inform the user and possibly retry"
            sweetAlertDialog.dismissWithAnimation();
        }else {
            Toast.makeText( context: RegisterActivity.this,  text: "conversion issue! big problems :(", Toast.LENGTH_SHORT).show();
            sweetAlertDialog.dismissWithAnimation();
        }
    }
}
```

8. List Mesin Activity



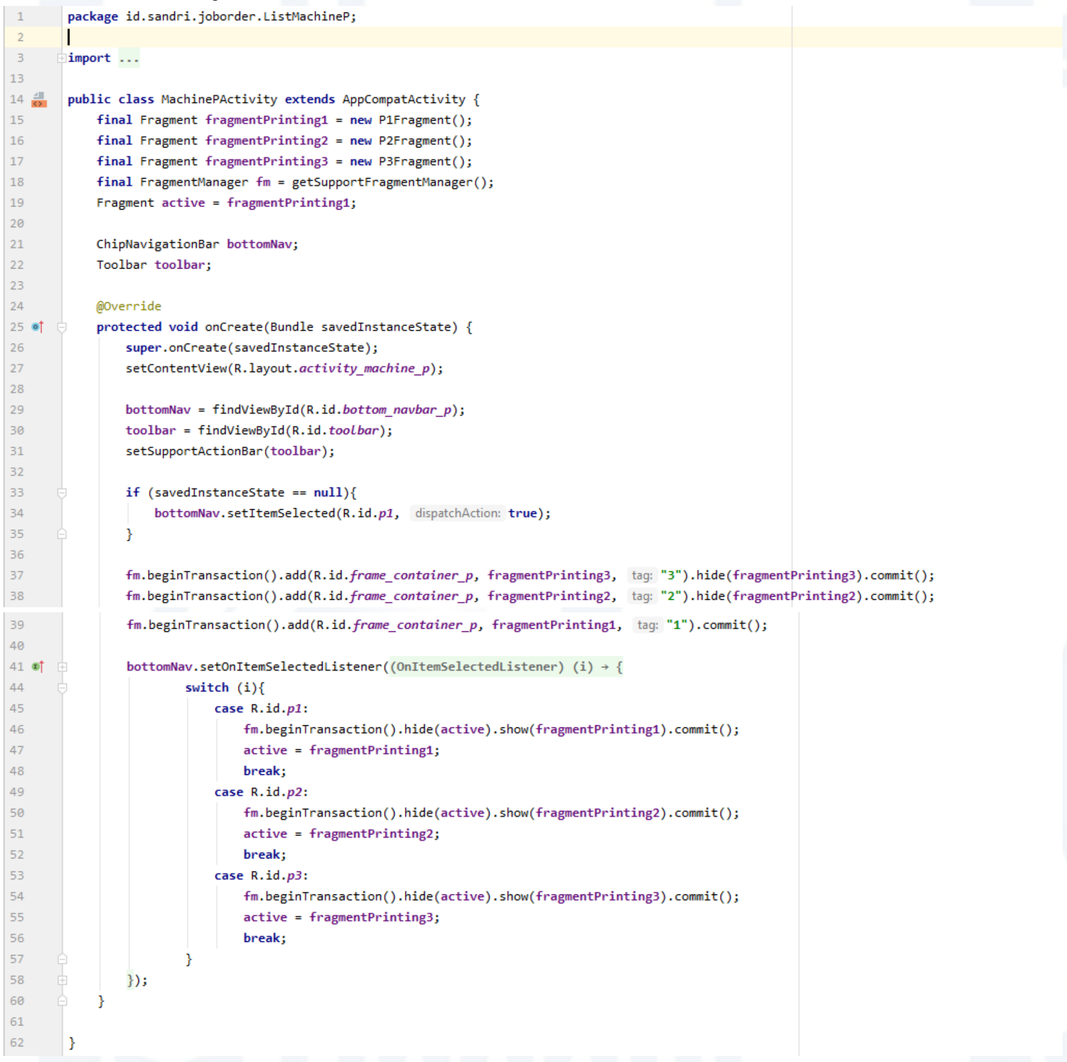
```
package id.sandri.joborder;

import ...

public class DashboardActivity extends AppCompatActivity {
    CardView btnMesin1, btnMesin2, btnMesin3, btnMesin4, btnMesin5, btnLogout;
    public LoginSession loginSession;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_dashboard);
        loginSession = new LoginSession( context: this);

        btnMesin1 = findViewById(R.id.btnMesin1);
        btnMesin2 = findViewById(R.id.btnMesin2);
        btnMesin3 = findViewById(R.id.btnMesin3);
        btnMesin4 = findViewById(R.id.btnMesin4);
        btnMesin5 = findViewById(R.id.btnMesin5);
        btnLogout = findViewById(R.id.btnLogout);

        btnMesin1.setOnClickListener((v) → {
                Intent intent = new Intent( packageContext: DashboardActivity.this, MachinePActivity.class);
                startActivity(intent);
        });

        btnMesin2.setOnClickListener((v) → {
                Intent intent = new Intent( packageContext: DashboardActivity.this, MachineDActivity.class);
                startActivity(intent);
        });
```



```
        btnMesin3.setOnClickListener((v) → {
                Intent intent = new Intent( packageContext: DashboardActivity.this, MachineSLActivity.class);
                startActivity(intent);
        });

        btnMesin4.setOnClickListener((v) → {
                Intent intent = new Intent( packageContext: DashboardActivity.this, MachineBMActivity.class);
                startActivity(intent);
        });

        btnMesin5.setOnClickListener((v) → {
                Intent intent = new Intent( packageContext: DashboardActivity.this, MachineEXActivity.class);
                startActivity(intent);
        });

        btnLogout.setOnClickListener((v) → {
                saveStatus();
                Intent intent = new Intent( packageContext: DashboardActivity.this, MainActivity.class).addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP
                startActivity(intent);
                finish();
        });
    }
    public void saveStatus() { loginSession.saveSPBoolean(LoginSession.SP_SUDAH_LOGIN,  value: false); }
```

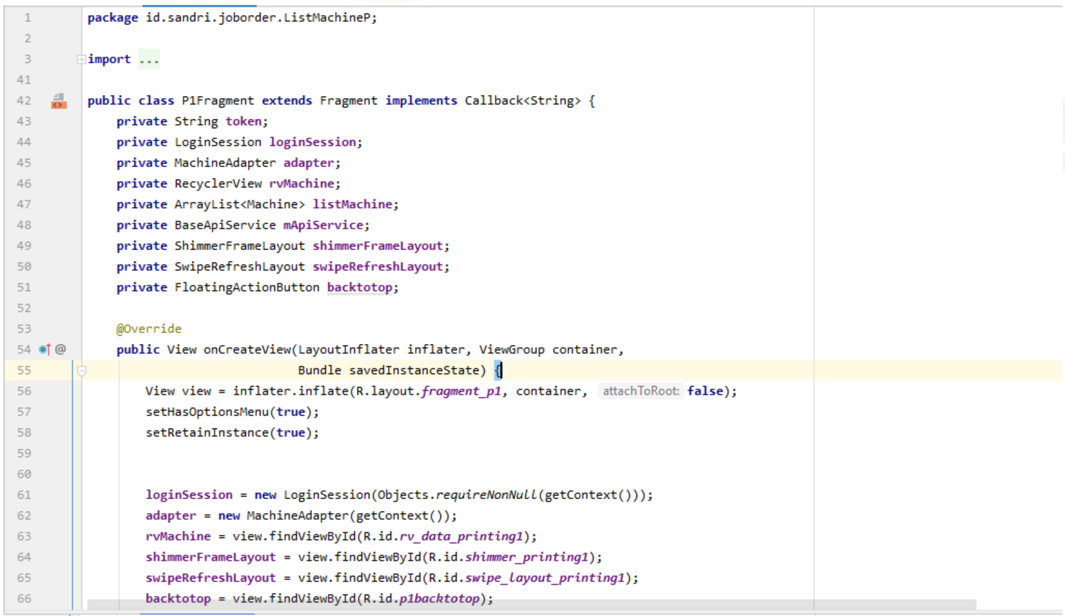
9. Mesin Activity dan Fragment

Machine Activity

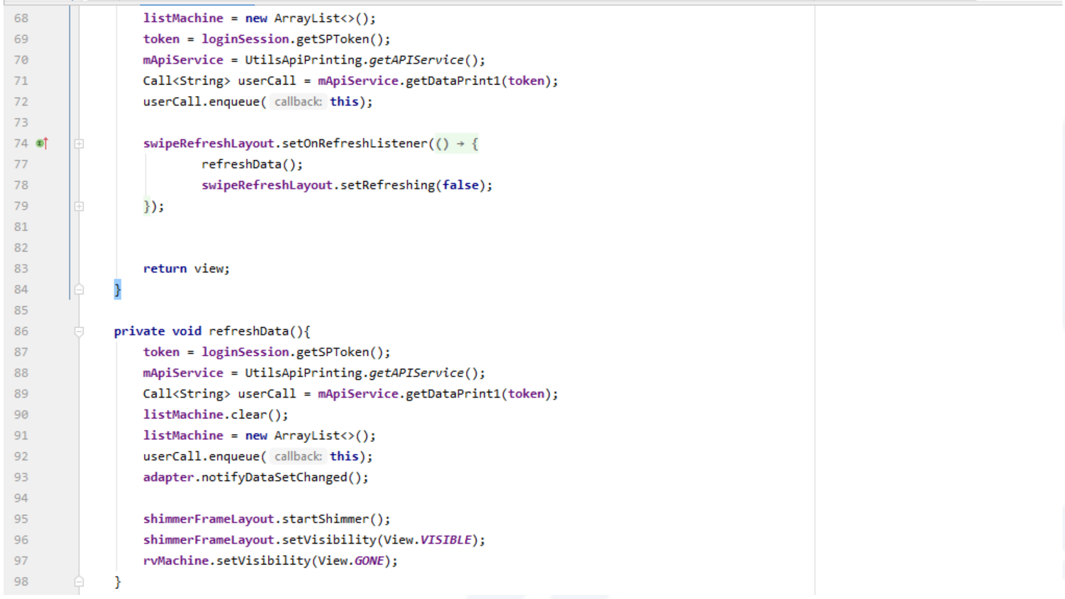


```java
package id.sandri.joborder.ListMachineP;

import ...

public class MachinePActivity extends AppCompatActivity {
    final Fragment fragmentPrinting1 = new P1Fragment();
    final Fragment fragmentPrinting2 = new P2Fragment();
    final Fragment fragmentPrinting3 = new P3Fragment();
    final FragmentManager fm = getSupportFragmentManager();
    Fragment active = fragmentPrinting1;

    ChipNavigationBar bottomNav;
    Toolbar toolbar;

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

        bottomNav = findViewById(R.id.bottom_navbar_p);
        toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        if (savedInstanceState == null){
            bottomNav.setItemSelected(R.id.p1,  dispatchAction: true);
        }

        fm.beginTransaction().add(R.id.frame_container_p, fragmentPrinting3,  tag: "3").hide(fragmentPrinting3).commit();
        fm.beginTransaction().add(R.id.frame_container_p, fragmentPrinting2,  tag: "2").hide(fragmentPrinting2).commit();
        fm.beginTransaction().add(R.id.frame_container_p, fragmentPrinting1,  tag: "1").commit();

        bottomNav.setOnItemSelectedListener((OnItemSelectedListener) (i) → {
                switch (i){
                    case R.id.p1:
                        fm.beginTransaction().hide(active).show(fragmentPrinting1).commit();
                        active = fragmentPrinting1;
                        break;
                    case R.id.p2:
                        fm.beginTransaction().hide(active).show(fragmentPrinting2).commit();
                        active = fragmentPrinting2;
                        break;
                    case R.id.p3:
                        fm.beginTransaction().hide(active).show(fragmentPrinting3).commit();
                        active = fragmentPrinting3;
                        break;
                }
        });
    }

}
```

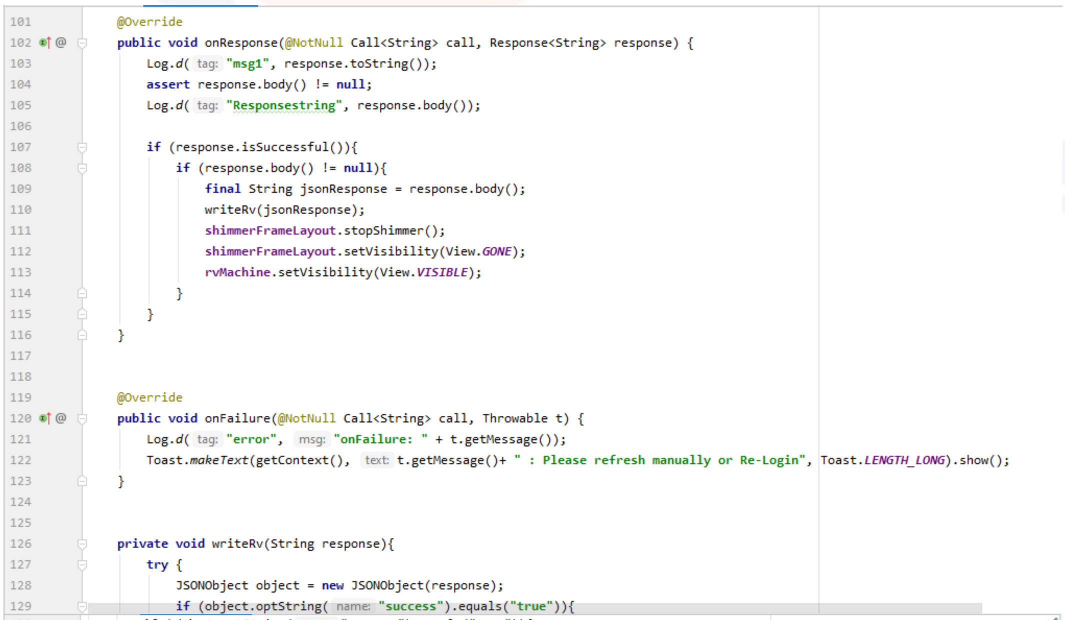
Machine Fragment



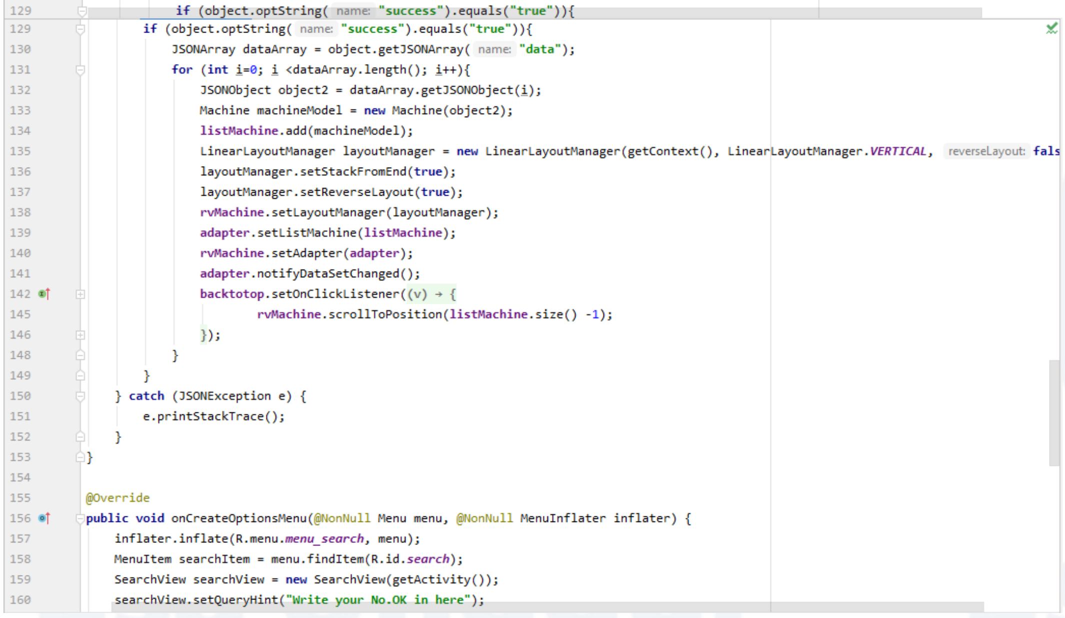
```
package id.sandri.joborder.ListMachineP;

import ...

public class P1Fragment extends Fragment implements Callback<String> {
    private String token;
    private LoginSession loginSession;
    private MachineAdapter adapter;
    private RecyclerView rvMachine;
    private ArrayList<Machine> listMachine;
    private BaseApiService mApiService;
    private ShimmerFrameLayout shimmerFrameLayout;
    private SwipeRefreshLayout swipeRefreshLayout;
    private FloatingActionButton backtotop;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_p1, container,  attachToRoot: false);
        setHasOptionsMenu(true);
        setRetainInstance(true);


        loginSession = new LoginSession(Objects.requireNonNull(getContext()));
        adapter = new MachineAdapter(getContext());
        rvMachine = view.findViewById(R.id.rv_data_printing1);
        shimmerFrameLayout = view.findViewById(R.id.shimmer_printing1);
        swipeRefreshLayout = view.findViewById(R.id.swipe_layout_printing1);
        backtotop = view.findViewById(R.id.p1backtotop);
```



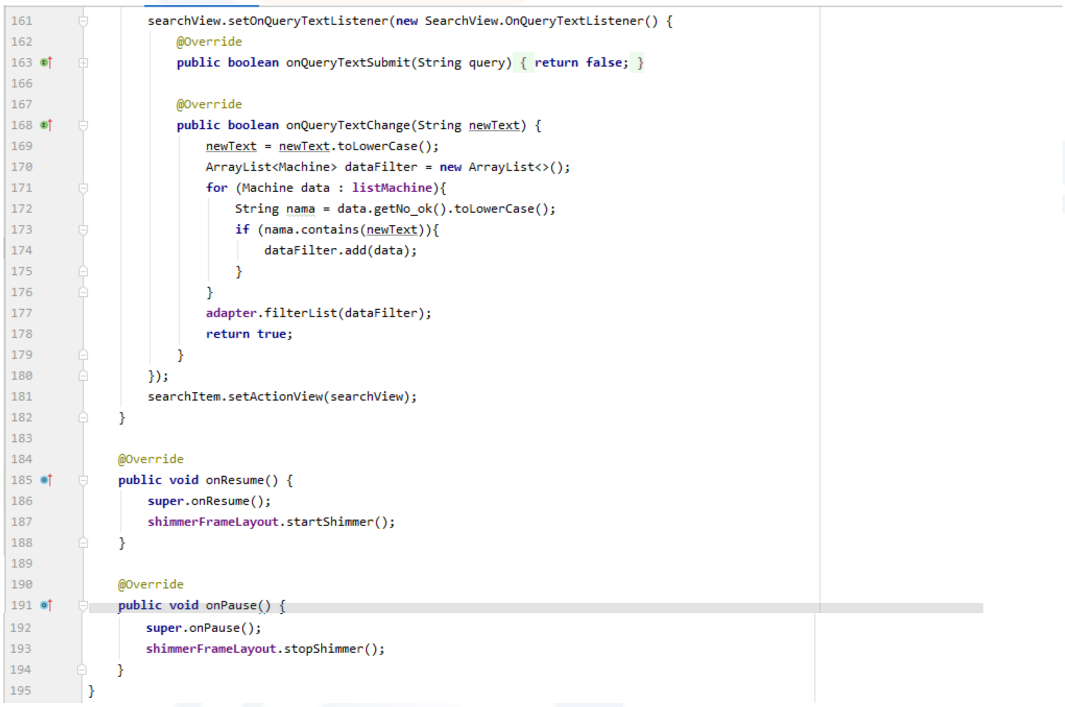
```
        listMachine = new ArrayList<>();
        token = loginSession.getSPToken();
        mApiService = UtilsApiPrinting.getAPIService();
        Call<String> userCall = mApiService.getDataPrint1(token);
        userCall.enqueue( callback: this);

        swipeRefreshLayout.setOnRefreshListener(() → {
                refreshData();
                swipeRefreshLayout.setRefreshing(false);
        });


        return view;
    }

    private void refreshData(){
        token = loginSession.getSPToken();
        mApiService = UtilsApiPrinting.getAPIService();
        Call<String> userCall = mApiService.getDataPrint1(token);
        listMachine.clear();
        listMachine = new ArrayList<>();
        userCall.enqueue( callback: this);
        adapter.notifyDataSetChanged();

        shimmerFrameLayout.startShimmer();
        shimmerFrameLayout.setVisibility(View.VISIBLE);
        rvMachine.setVisibility(View.GONE);
    }
```



```java
    @Override
    public void onResponse(@NotNull Call<String> call, Response<String> response) {
        Log.d( tag: "msg1", response.toString());
        assert response.body() != null;
        Log.d( tag: "Responsestring", response.body());

        if (response.isSuccessful()){
            if (response.body() != null){
                final String jsonResponse = response.body();
                writeRv(jsonResponse);
                shimmerFrameLayout.stopShimmer();
                shimmerFrameLayout.setVisibility(View.GONE);
                rvMachine.setVisibility(View.VISIBLE);
            }
        }
    }


    @Override
    public void onFailure(@NotNull Call<String> call, Throwable t) {
        Log.d( tag: "error",  msg: "onFailure: " + t.getMessage());
        Toast.makeText(getContext(),  text: t.getMessage()+ " : Please refresh manually or Re-Login", Toast.LENGTH_LONG).show();
    }


    private void writeRv(String response){
        try {
            JSONObject object = new JSONObject(response);
            if (object.optString( name: "success").equals("true")){
```

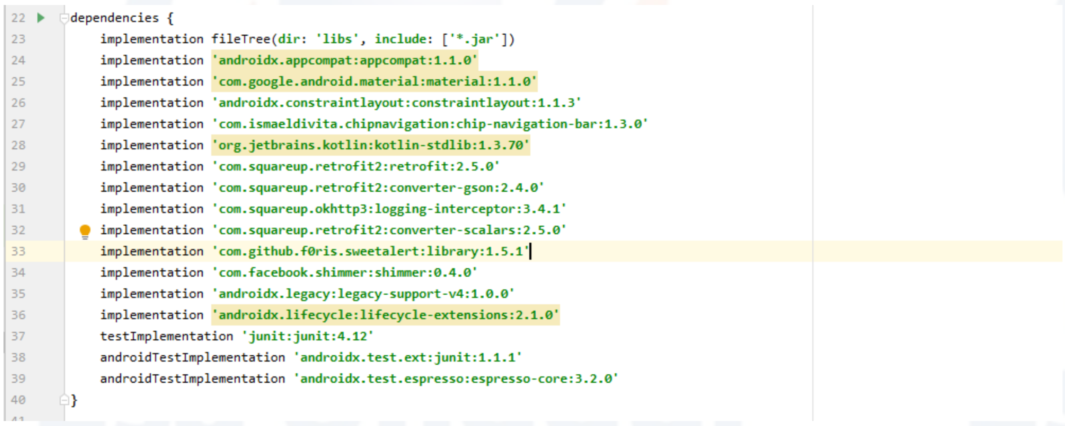


```java
        if (object.optString( name: "success").equals("true")){
            JSONArray dataArray = object.getJSONArray( name: "data");
            for (int i=0; i <dataArray.length(); i++){
                JSONObject object2 = dataArray.getJSONObject(i);
                Machine machineModel = new Machine(object2);
                listMachine.add(machineModel);
                LinearLayoutManager layoutManager = new LinearLayoutManager(getContext(), LinearLayoutManager.VERTICAL,  reverseLayout: fals
                layoutManager.setStackFromEnd(true);
                layoutManager.setReverseLayout(true);
                rvMachine.setLayoutManager(layoutManager);
                adapter.setListMachine(listMachine);
                rvMachine.setAdapter(adapter);
                adapter.notifyDataSetChanged();
                backtotop.setOnClickListener((v) → {
                        rvMachine.scrollToPosition(listMachine.size() -1);
                });
            }
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
}

@Override
public void onCreateOptionsMenu(@NonNull Menu menu, @NonNull MenuInflater inflater) {
    inflater.inflate(R.menu.menu_search, menu);
    MenuItem searchItem = menu.findItem(R.id.search);
    SearchView searchView = new SearchView(getActivity());
    searchView.setQueryHint("Write your No.OK in here");
```



```
        searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
            @Override
            public boolean onQueryTextSubmit(String query) { return false; }

            @Override
            public boolean onQueryTextChange(String newText) {
                newText = newText.toLowerCase();
                ArrayList<Machine> dataFilter = new ArrayList<>();
                for (Machine data : listMachine){
                    String nama = data.getNo_ok().toLowerCase();
                    if (nama.contains(newText)){
                        dataFilter.add(data);
                    }
                }
                adapter.filterList(dataFilter);
                return true;
            }
        });
        searchItem.setActionView(searchView);
    }

    @Override
    public void onResume() {
        super.onResume();
        shimmerFrameLayout.startShimmer();
    }

    @Override
    public void onPause() {
        super.onPause();
        shimmerFrameLayout.stopShimmer();
    }
}
```

10. Android Dependency



```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'androidx.appcompat:appcompat:1.1.0'
    implementation 'com.google.android.material:material:1.1.0'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    implementation 'com.ismaeldivita.chipnavigation:chip-navigation-bar:1.3.0'
    implementation 'org.jetbrains.kotlin:kotlin-stdlib:1.3.70'
    implementation 'com.squareup.retrofit2:retrofit:2.5.0'
    implementation 'com.squareup.retrofit2:converter-gson:2.4.0'
    implementation 'com.squareup.okhttp3:logging-interceptor:3.4.1'
    implementation 'com.squareup.retrofit2:converter-scalars:2.5.0'
    implementation 'com.github.f0ris.sweetalert:library:1.5.1'
    implementation 'com.facebook.shimmer:shimmer:0.4.0'
    implementation 'androidx.legacy:legacy-support-v4:1.0.0'
    implementation 'androidx.lifecycle:lifecycle-extensions:2.1.0'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test.ext:junit:1.1.1'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'
}
```