

**LAPORAN UJIAN AKHIR SEMESTER PROYEK  
PEMROGRAMAN PLATFORM KHUSUS ANDROID STUDIO  
DENGAN IMPLEMENTASI API**



Oleh  
Gholidho Herda Prilasakly  
222112074

Dosen Pembimbing  
Ibnu Santoso

Politeknik Statistika STIS  
Tahun ajaran 2023/2024

## A. Aplikasi

Aplikasi kali ini adalah Pelaporan Kerusakan Inventaris Kelas, yang digunakan untuk melakukan monitoring dan pelaporan apa saja yang rusak.

## B. Tujuan

Tujuan pembuatan aplikasi kali ini untuk memudahkan mahasiswa melakukan pelaporan jika ada inventaris yang rusak seperti spidol habis, ac rusak, kursi patah, dan sebagainya serta sebagai alat monitoring oleh pihak administrasi kampus terhadap semua inventaris di kelas yang ada di Politeknik Statistika STIS.

## C. Dokumentasi

Dokumen akan dilakukan per-*activity* untuk memudahkan dilihat dan dipahami dan dijelaskan tentunya.

### 1. List Dependency yang diperlukan

```
dependencies {
    implementation("androidx.appcompat:appcompat:1.6.1")
    implementation("com.google.android.material:material:1.10.0")
    implementation("com.android.volley:volley:1.2.0")

    testImplementation("junit:junit:4.13.2")
    androidTestImplementation("androidx.test.ext:junit:1.1.5")
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")

    implementation ("com.squareup.retrofit2:retrofit:2.9.0")
    implementation ("com.squareup.retrofit2:converter-gson:2.9.0")

    implementation ("com.squareup.okhttp3:logging-interceptor:4.9.1")
    implementation ("com.squareup.retrofit2:converter-scalars:2.9.0")
}
```

### 2. Retrofit

```
package com.example.laporkelas.api;

import okhttp3.OkHttpClient;
import okhttp3.logging.HttpLoggingInterceptor;
import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;
import retrofit2.converter.scalars.ScalarsConverterFactory;

public class ApiClient {
    private static final String BASE_URL = "http://192.168.15.1:8089/";
    private static Retrofit retrofit = null;

    public static Retrofit getRetrofitInstance() {
        if (retrofit == null) {
            HttpLoggingInterceptor interceptor = new HttpLoggingInterceptor();
            interceptor.setLevel(HttpLoggingInterceptor.Level.BODY);
            OkHttpClient client = new OkHttpClient.Builder().addInterceptor(interceptor).build();

            retrofit = new Retrofit.Builder()
                .baseUrl(BASE_URL)
                .addConverterFactory(ScalarsConverterFactory.create())
                .addConverterFactory(GsonConverterFactory.create())
                .client(client)
                .build();
        }
        return retrofit;
    }

    public static Retrofit getRetrofitInstance(final String token) {
        HttpLoggingInterceptor interceptor = new HttpLoggingInterceptor();
        interceptor.setLevel(HttpLoggingInterceptor.Level.BODY);
        OkHttpClient.Builder httpClient = new OkHttpClient.Builder();
        httpClient.addInterceptor(interceptor);

        httpClient.addInterceptor(chain -> {
            okhttp3.Request original = chain.request();
            okhttp3.Request.Builder requestBuilder = original.newBuilder()
                .header("Authorization", "Bearer " + token)
                .method(original.method(), original.body());
            okhttp3.Request request = requestBuilder.build();
        });
    }
}
```

```

        return chain.proceed(request);
    });

OkHttpClient client = httpClient.build();
return new Retrofit.Builder()
    .baseUrl(BASE_URL)
    .addConverterFactory(ScalarsConverterFactory.create())
    .addConverterFactory(GsonConverterFactory.create())
    .client(client)
    .build();
}
}

```

Retrofit digunakan untuk mengambil token yang didapatkan dari login nanti untuk dikirim ke fitur-fitur lain karena semuanya memerlukan token untuk melakukan activity nya.

### 3. Register Activity

#### a. Database awal

<b>id</b>	<b>email</b>	<b>kelas</b>	<b>name</b>	<b>nim</b>	<b>password</b>
-----------	--------------	--------------	-------------	------------	-----------------

#### b. Kode buat register

- ◆ Api Service

```
@POST("/register")
Call<UserRegistrationModel> register(@Body UserRegistrationModel user);
```

Bsia dilihat disini request nya menggunakan Model UserRegisterModel dan responnya juga berupa UserRegisterModel jadi nanti hanya perlu membuat 1 model untuk request dan responnya sekaligus.

- ◆ UserRegisterModel

```

package com.example.laporkelas.model;

public class UserRegistrationModel {

    private String name;
    private String nim;
    private String kelas;
    private String email;
    private String password;

    public UserRegistrationModel(String name, String nim, String kelas, String email, String password) {
        this.name = name;
        this.nim = nim;
        this.kelas = kelas;
        this.email = email;
        this.password = password;
    }

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getNim() {
        return nim;
    }
    public void setNim(String nim) {
        this.nim = nim;
    }
    public String getKelas() {
        return kelas;
    }
    public void setKelas(String kelas) {
        this.kelas = kelas;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public String getPassword() {
        return password;
    }
}
```

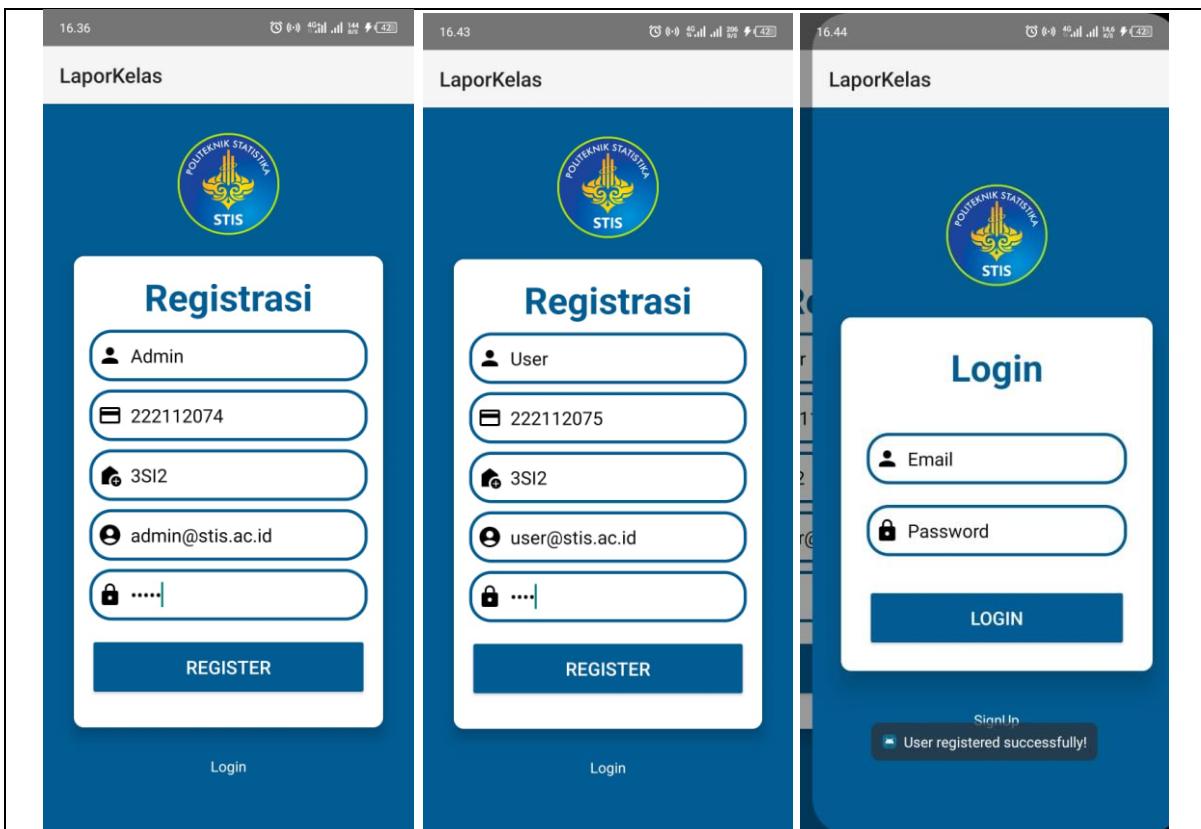
```

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

```

Digunakan untuk mengirim dan menerima request/respon dari web service nya nanti.

- ◆ Contoh kegiatan disini kita bikin 2 akun pertama [admin@stis.ac.id](mailto:admin@stis.ac.id) dan [user@stis.ac.id](mailto:user@stis.ac.id) untuk membedakan tampilan-tampilan selanjutnya



Kalau berhasil akan muncul toast registration berhasil dan akan berpindah ke halaman login untuk selanjutkan melakukan login

- ◆ Update database sekarang

	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	<b>id</b>	<b>email</b>	<b>kelas</b>	<b>name</b>	<b>nim</b>	<b>password</b>
<input type="checkbox"/>	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	1	admin@stis.ac.id	3SI2	Admin	222112074	\$2a\$10\$XGVZ0WHK3bLKc2bG6BJWLl77xs0tne1QuVFFeGMGEz...
<input type="checkbox"/>	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	2	user@stis.ac.id	3SI2	User	222112075	\$2a\$10\$bdJIMbQYoyAkfLQBzBg/ZerWoQeuQy1z4668s4EwnGL...

Dari sini dapat dilihat kalau proses register sudah dapat dilakukan dengan baik dan benar.

#### 4. Login Acitivity

##### ◆ ApiService

```
@POST("/login")
Call<AuthResponse> login(@Body AuthRequest authRequest);
```

Terlihat untuk model yang diperlukan ada AuthRequest untuk melakukan request dan AuthRespon digunakan untuk menerima respon. Request respon punya bentu berbeda sehingga harus membuat kelas keduanya.

##### ◆ Model

Request	Response
<pre>public class AuthRequest {     private String email;     private String password;      public AuthRequest(String email, String password) {         this.email = email;         this.password = password;     }      // Tambahkan getter dan setter     public String getEmail() {         return email;     }     public void setEmail(String email) {         this.email = email;     }     public String getPassword() {         return password;     }     public void setPassword(String password) {         this.password = password;     } }</pre>	<pre>package com.example.laporkelas.model;  import java.util.List;  public class AuthResponse {     private String email;     private String accessToken;     private List&lt;String&gt; roles;      public AuthResponse(String email, String accessToken, List&lt;String&gt; roles) {         this.email = email;         this.accessToken = accessToken;         this.roles = roles;     }      // Add getters and setters for the fields      public String getEmail() {         return email;     }     public void setEmail(String email) {         this.email = email;     }     public String getAccessToken() {         return accessToken;     }      public void setAccessToken(String accessToken) {         this.accessToken = accessToken;     }     public List&lt;String&gt; getRoles() {         return roles;     }     public void setRoles(List&lt;String&gt; roles) {         this.roles = roles;     } }</pre>

Disini bisa dilihat untuk request dia mengirim email dan password sedangkan responnya berupa email, accessToken dan juga List dari Role yang ada.

##### ◆ LoginActivity

```
package com.example.laporkelas.activity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.example.laporkelas.R;
import com.example.laporkelas.api.Api;
import com.example.laporkelas.api.ApiClient;
import com.example.laporkelas.model.AuthRequest;
import com.example.laporkelas.model.AuthResponse;

import java.util.ArrayList;
import retrofit2.Call;
```

```

import retrofit2.Callback;
import retrofit2.Response;
import retrofit2.Retrofit;

public class LoginActivity extends AppCompatActivity {

    private EditText emailEditText;
    private EditText passwordEditText;
    private Button loginButton;
    private TextView tv;

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

        emailEditText = findViewById(R.id.emailEditText);
        passwordEditText = findViewById(R.id.passwordEditText);
        loginButton = findViewById(R.id.loginButton);
        tv = findViewById(R.id.signupText);

        loginButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                performLogin();
            }
        });

        tv.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                // Handle Profile Icon click
                register();
            }
        });
    }

    private void performLogin() {
        String email = emailEditText.getText().toString();
        String password = passwordEditText.getText().toString();

        Retrofit retrofit = ApiClient.getRetrofitInstance();
        Api api = retrofit.create(Api.class);
        Call<AuthResponse> call = api.login(new AuthRequest(email, password));

        call.enqueue(new Callback<AuthResponse>() {
            @Override
            public void onResponse(Call<AuthResponse> call, Response<AuthResponse> response) {
                if (response.isSuccessful() && response.body() != null) {
                    Toast.makeText(LoginActivity.this, "Login Success!", Toast.LENGTH_SHORT).show();
                    AuthResponse authResponse = response.body();
                    String token = authResponse.getAccessToken();

                    ArrayList<String> roles = new ArrayList<>(authResponse.getRoles());

                    Intent intent = new Intent(LoginActivity.this, DashboardActivity.class);
                    intent.putExtra("TOKEN", token);
                    intent.putStringArrayListExtra("ROLES", roles);
                    startActivity(intent);
                } else {
                    Toast.makeText(LoginActivity.this, "Email atau Password Salah",
                    Toast.LENGTH_SHORT).show();
                }
            }

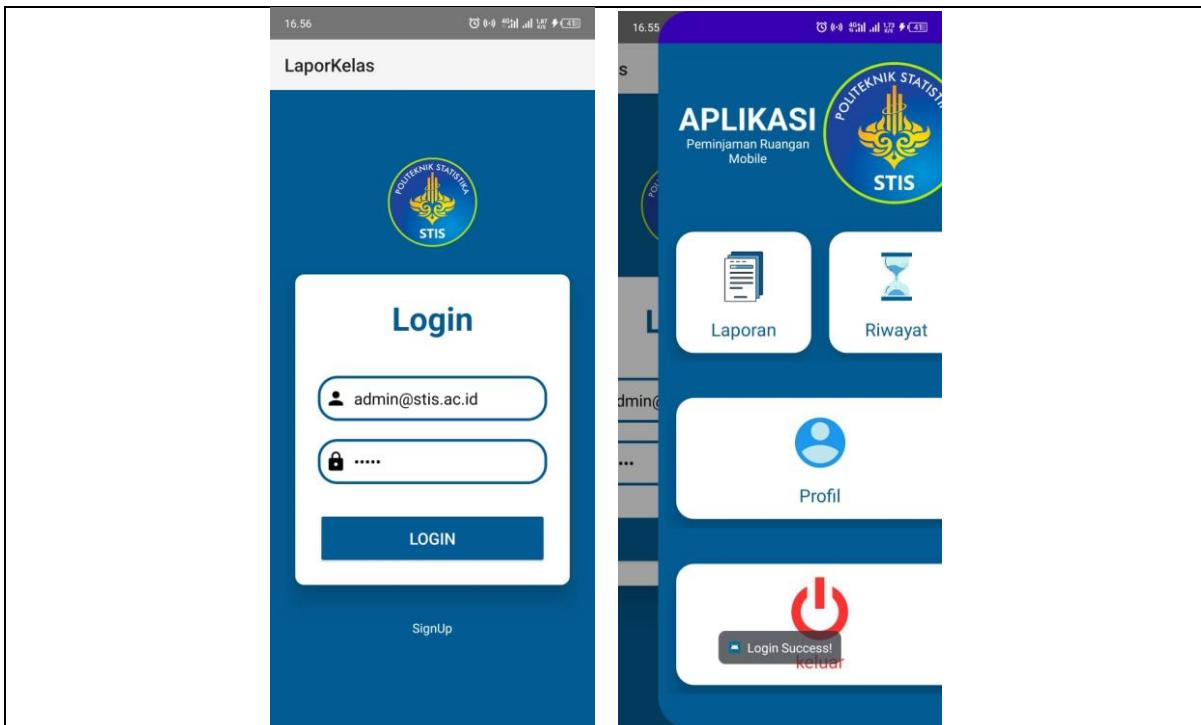
            @Override
            public void onFailure(Call<AuthResponse> call, Throwable t) {
                Toast.makeText(LoginActivity.this, "Unexpected Error " + t.getMessage(),
                Toast.LENGTH_SHORT).show();
            }
        });
    }

    private void register(){
        Intent intent = new Intent(LoginActivity.this, RegistrationActivity.class);
        startActivity(intent);
        Toast.makeText(LoginActivity.this, "Register", Toast.LENGTH_SHORT).show();
    }
}

```

Disini bisa dilihat kalau berhasil dia akan mengintent ke dashboard dengan mengirim token yang didapat dan juga roles dari pengguna yang login.

◆ Contoh Aksi



Jika login berhasil maka akan masuk ke laman dashboard dan muncul Toast Login Success

## 5. DashboardActivity

◆ Kode

```
package com.example.laporkelas.activity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

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

import com.example.laporkelas.R;
import com.example.laporkelas.model.*;

import java.util.ArrayList;

public class DashboardActivity extends AppCompatActivity {

    private TextView profileText, riwayatText, laporanText, usersText, konfirmasiText, exitText;
    private ImageView profileImage, riwayatImage, laporanImage, usersImage, konfirmasiImage, exitImage;

    private CardView penggunaCard, konfirmasiCard;
    private String token;

    ArrayList<String> roles;

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

        profileText = findViewById(R.id.profileText);
        profileImage = findViewById(R.id.profileImage);

        laporanText = findViewById(R.id.laporanText);
        laporanImage = findViewById(R.id.laporanImage);
        riwayatText = findViewById(R.id.riwayatText);
        riwayatImage = findViewById(R.id.riwayatImage);
        penggunaCard = findViewById(R.id.penggunaCard);
        konfirmasiCard = findViewById(R.id.konfirmasiCard);
```

```
exitText = findViewById(R.id.exitText);
exitImage = findViewById(R.id.exitImage);
token = getIntent().getStringExtra("TOKEN");
roles = getIntent().getStringArrayListExtra("ROLES");

profileText.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // Handle Profile Icon click
        openProfile();
    }
});
profileImage.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // Handle Profile Icon click
        openProfile();
    }
});

// Setup click listener for Create Report Button
laporanText.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        createReport();
    }
});
laporanImage.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        createReport();
    }
});

exitText.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        exit();
    }
});
exitImage.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        exit();
    }
});

// Setup click listener for View History Button
riwayatText.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        viewHistory();
    }
});
riwayatImage.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        viewHistory();
    }
});

usersText = findViewById(R.id.usersText);
usersImage = findViewById(R.id.usersImage);
if (userHasAdminRole()) {
    usersText.setVisibility(View.VISIBLE);
    penggunaCard.setVisibility(View.VISIBLE);
    konfirmasiCard.setVisibility(View.VISIBLE);
    usersImage.setVisibility(View.VISIBLE);
} else {
    usersText.setVisibility(View.GONE);
    penggunaCard.setVisibility(View.GONE);
    konfirmasiCard.setVisibility(View.GONE);
    usersImage.setVisibility(View.GONE);
}
usersText.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        viewAllUsers();
    }
});
usersImage.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        viewAllUsers();
    }
});
```

```

konfirmasiText = findViewById(R.id.konfirmasiText);
konfirmasiImage = findViewById(R.id.konfirmasiImage);

        konfirmasiText.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                konfirmasi();
            }
        });
        konfirmasiImage.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                konfirmasi();
            }
        });
    }

private boolean userHasAdminRole() {
    for (String role : roles) {
        if ("ROLE_ADMIN".equals(role)) {
            return true;
        }
    }
    return false;
}

private void viewAllUsers() {
    // Intent to open an activity that shows all users
    Intent intent = new Intent(DashboardActivity.this, UsersActivity.class);
    intent.putExtra("TOKEN", token);
    startActivity(intent);
    Toast.makeText(DashboardActivity.this, "View All Users", Toast.LENGTH_SHORT).show();
}

private void openProfile() {
    // Intent to open profile activity (example)
    Intent intent = new Intent(DashboardActivity.this, ProfileActivity.class);
    intent.putExtra("TOKEN", token);
    startActivity(intent);
    Toast.makeText(DashboardActivity.this, "Profile Clicked", Toast.LENGTH_SHORT).show();
}

private void createReport() {
    Intent intent = new Intent(DashboardActivity.this, LaporanActivity.class);
    intent.putExtra("TOKEN", token);
    startActivity(intent);
    Toast.makeText(DashboardActivity.this, "Create Report", Toast.LENGTH_SHORT).show();
}

private void viewHistory() {
    Intent intent = new Intent(DashboardActivity.this, RiwayatActivity.class);
    intent.putExtra("TOKEN", token);
    intent.putStringArrayListExtra("ROLES", roles);
    startActivity(intent);
    Toast.makeText(DashboardActivity.this, "View History", Toast.LENGTH_SHORT).show();
}

private void konfirmasi() {
    Intent intent = new Intent(DashboardActivity.this, KonfirmasiActivity.class);
    intent.putExtra("TOKEN", token);
    startActivity(intent);
    Toast.makeText(DashboardActivity.this, "Konfirmasi", Toast.LENGTH_SHORT).show();
}

private void exit() {
    Intent intent = new Intent(DashboardActivity.this, LoginActivity.class);
    startActivity(intent);
    Toast.makeText(DashboardActivity.this, "Login", Toast.LENGTH_SHORT).show();
}
}

```

Namanya Dashboard dia hanya menjadi jembatan penghubung fitur-fitur selanjutnya. Disini perlu dilihat ketika ada yang di klik dia pasti mengirimkan token dan beberapa ada yang role untuk melihat role yang masuk. Disana juga membedakan tampilan admin user dengan membatasinya dengan role.

◆ Contoh tampilan

User	Admin
	

Bisa dilihat di bagian user tidak muncul Pengguna dan Konfirmasi karena itu hanya bisa dilihat oleh Admin saja.

## 6. Fitur Laporan

◆ ApiService

```
@PATCH("/laporan/lapor/{kelas}")
Call<String> updateLaporan(
    @Header("Authorization") String token,
    @Path("kelas") Long kelas,
    @Body LaporanDto laporanDto
);

@GET("/laporan/all")
Call<List<LaporanDto>> getLaporanAll(@Header("Authorization") String token);

@GET("/laporan/kelas/{kelas}")
Call<List<LaporanDto>> getLaporanByKelas(
    @Header("Authorization") String token,
    @Path("kelas") Long kelas);

```

Disini dalam 1 fitur laporan ada beberapa endpoint yang langsung dijalankan. Requestnya memerlukan LaporanDto jadi membuat Model LaporanDto

◆ Model

```
package com.example.laporkelas.model;

public class LaporanDto {
    private Long kelas;
    private String proyektor;
    private String kursi;
    private String papan;
    private String spidol;
    private String penghapus;
    private String ac;
    private String jamDinding;
    private String lainnya;

    public LaporanDto() {
    }

    public Long getKelas() {
    }
}
```

```

        return kelas;
    }

    public void setKelas(Long kelas) {
        this.kelas = kelas;
    }

    public String getProyektor() {
        return proyektor;
    }

    public void setProyektor(String proyektor) {
        this.proyektor = proyektor;
    }

    public String getKursi() {
        return kursi;
    }

    public void setKursi(String kursi) {
        this.kursi = kursi;
    }

    public String getPapan() {
        return papan;
    }

    public void setPapan(String papan) {
        this.papan = papan;
    }

    public String getSpidol() {
        return spidol;
    }

    public void setSpidol(String spidol) {
        this.spidol = spidol;
    }

    public String getPenghapus() {
        return penghapus;
    }

    public void setPenghapus(String penghapus) {
        this.penghapus = penghapus;
    }

    public String getAc() {
        return ac;
    }

    public void setAc(String ac) {
        this.ac = ac;
    }

    public String getJamDinding() {
        return jamDinding;
    }

    public void setJamDinding(String jamDinding) {
        this.jamDinding = jamDinding;
    }

    public String getLainnya() {
        return lainnya;
    }

    public void setLainnya(String lainnya) {
        this.lainnya = lainnya;
    }
}

```

## ◆ Kode LaporanActivity

```

package com.example.laporkelas.activity;

import android.content.Intent;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

```

```
import com.example.laporkelas.R;
import com.example.laporkelas.adapter.LaporanAdapter;
import com.example.laporkelas.api.Api;
import com.example.laporkelas.api.ApiClient;
import com.example.laporkelas.model.LaporanDto;

import java.util.ArrayList;
import java.util.List;

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

public class LaporanActivity extends AppCompatActivity {

    private RecyclerView recyclerView;
    private LaporanAdapter adapter;
    private String token;

    private EditText kelasEditText;
    private Button searchButton, buatLaporan;
    Long kelas;

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

        token = getIntent().getStringExtra("TOKEN");

        recyclerView = findViewById(R.id.recyclerViewLaporan);
        recyclerView.setLayoutManager(new LinearLayoutManager(this));

        adapter = new LaporanAdapter(new ArrayList<>());
        recyclerView.setAdapter(adapter);

        kelasEditText = findViewById(R.id.kelasEditText);
        searchButton = findViewById(R.id.searchButton);
        buatLaporan = findViewById(R.id.buatLaporanButton);

        buatLaporan.setOnClickListener(v -> buatLaporan());
        searchButton.setOnClickListener(v -> performSearch());
        loadLaporan();
    }

    private void buatLaporan() {
        Intent intent = new Intent(LaporanActivity.this, BuatLaporanActivity.class);
        intent.putExtra("TOKEN", token);
        startActivity(intent);
    }

    private void performSearch() {
        String kelasString = kelasEditText.getText().toString();
        if (!kelasString.isEmpty()) {
            try {
                kelas = Long.parseLong(kelasString);
                searchLaporanByKelas(kelas);
            } catch (NumberFormatException e) {
                Toast.makeText(this, "Invalid class number", Toast.LENGTH_SHORT).show();
            }
        } else {
            Toast.makeText(this, "Please enter a class number", Toast.LENGTH_SHORT).show();
        }
    }

    private void searchLaporanByKelas(long kelas) {
        Retrofit retrofit = ApiClient.getRetrofitInstance(token);
        Api api = retrofit.create(Api.class);
        Call<List<LaporanDto>> call = api.getLaporanByKelas(token, kelas);

        call.enqueue(new Callback<List<LaporanDto>>() {
            @Override
            public void onResponse(Call<List<LaporanDto>> call, Response<List<LaporanDto>> response) {
                if (response.isSuccessful() && response.body() != null) {
                    adapter.updateData(response.body());
                } else {
                    loadLaporan();
                }
            }

            @Override
            public void onFailure(Call<List<LaporanDto>> call, Throwable t) {
                Toast.makeText(LaporanActivity.this, "Network error: " + t.getMessage(),
                Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

```

    }

    private void loadLaporan() {
        Retrofit retrofit = ApiClient.getRetrofitInstance(token);
        Api api = retrofit.create(Api.class);
        Call<List<LaporanDto>> call = api.getLaporanAll(token);

        call.enqueue(new Callback<List<LaporanDto>>() {
            @Override
            public void onResponse(Call<List<LaporanDto>> call, Response<List<LaporanDto>> response) {
                if (response.isSuccessful() && response.body() != null) {
                    adapter = new LaporanAdapter(response.body());
                    recyclerView.setAdapter(adapter);
                } else {
                    Toast.makeText(LaporanActivity.this, "Error fetching data",
                    Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onFailure(Call<List<LaporanDto>> call, Throwable t) {
                Toast.makeText(LaporanActivity.this, "Network error: " + t.getMessage(),
                Toast.LENGTH_SHORT).show();
            }
        });
    }
}

```

Di sini sudah menerapkan adapter jadi harus membuat kelas adapter untuk memberikan tampilannya.

#### ◆ LaporanAdapter

```

package com.example.laporkelas.adapter;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.example.laporkelas.R;
import com.example.laporkelas.model.LaporanDto;

import java.util.List;

public class LaporanAdapter extends RecyclerView.Adapter<LaporanAdapter.ViewHolder> {

    private final List<LaporanDto> laporanList;

    public LaporanAdapter(List<LaporanDto> laporanList) {
        this.laporanList = laporanList;
    }

    public void updateData(List<LaporanDto> newLaporanList) {
        laporanList.clear();
        laporanList.addAll(newLaporanList);
        notifyDataSetChanged();
    }

    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.item_laporan, parent,
        false);
        return new ViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
        LaporanDto laporan = laporanList.get(position);
        holder.kelasTextView.setText(""+laporan.getKelas());
        holder.proyektorTextView.setText(laporan.getProyektor());
        holder.kursiTextView.setText(laporan.getKursi());
        holder.papanTextView.setText(laporan.getPapan());
        holder.spidolTextView.setText(laporan.getSpidol());
        holder.penghapusTextView.setText(laporan.getPenghapus());
        holder.acTextView.setText(laporan.getAc());
        holder.jamDindingTextView.setText(laporan.getJamDinding());
        holder.lainnyaTextView.setText(laporan.getLainnya());
    }
}

```

```

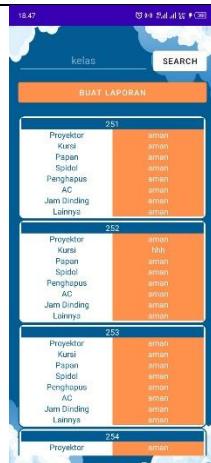
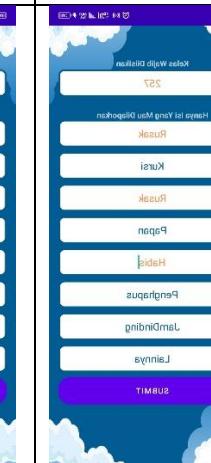
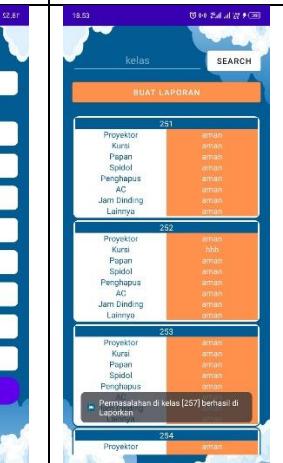
@Override
public int getItemCount() {
    return laporanList.size();
}

static class ViewHolder extends RecyclerView.ViewHolder {
    TextView kelasTextView, proyektorTextView, kursiTextView, papanTextView,
        spidolTextView, penghapusTextView, acTextView, jamDindingTextView, lainnyaTextView;

    ViewHolder(View itemView) {
        super(itemView);
        kelasTextView = itemView.findViewById(R.id.kelasTextView);
        proyektorTextView = itemView.findViewById(R.id.proyektorTextView);
        kursiTextView = itemView.findViewById(R.id.kursiTextView);
        papanTextView = itemView.findViewById(R.id.papanTextView);
        spidolTextView = itemView.findViewById(R.id.spidolTextView);
        penghapusTextView = itemView.findViewById(R.id.penghapusTextView);
        acTextView = itemView.findViewById(R.id.acTextView);
        jamDindingTextView = itemView.findViewById(R.id.jamDindingTextView);
        lainnyaTextView = itemView.findViewById(R.id.lainnyaTextView);
    }
}

```

◆ Contoh kegiatan

Contoh tampilan saat masuk ke Laporan	Contoh search laporan dengan memasukkan kelas nya	Contoh tampilan form mengajukan laporan	Contoh kalau input kelas dan laporan.	Contoh kalau laporan berhasil dikirimkan
				

## 7. Fitur Profile

### ◆ Api

```
@GET("/users/profile")
Call<UserDto> myProfile(@Header("Authorization")String token);

@PATCH("/users/update/{userId}")
Call<String> updateUser(
    @Path("userId") Long userId,
    @Body UserDto userDto,
    @Query("emailValidation") String emailValidation,
    @Query("passwordValidation") String passwordValidation
),
```

Karena disini hanya perlu membuat object UserDto jadi hanya perlu membuat UserDto aja, String sudah otomatis terbaca sebagai object.

### ◆ Model

```
package com.example.laporkelas.model;

public class UserDto {
    private Long id;
    private String name;
    private String nim;
    private String kelas;
    private String email;
    private String password;

    public UserDto(String name, String email, String nim, String kelas, String password) {
        this.name = name;
        this.email = email;
        this.nim = nim;
        this.kelas = kelas;
        this.password = password;
    }

    // Getters and Setters
    public Long getId() {
        return id;
    }
    public void setId(Long id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getNim() {
        return nim;
    }
    public void setNim(String nim) {
        this.nim = nim;
    }
    public String getKelas() {
        return kelas;
    }
    public void setKelas(String kelas) {
        this.kelas = kelas;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
}
```

## ◆ ProfileActivity

```
package com.example.laporkelas.activity;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.example.laporkelas.R;
import com.example.laporkelas.api.Api;
import com.example.laporkelas.api.ApiClient;
import com.example.laporkelas.model.UserDto;

import java.util.ArrayList;

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

public class ProfileActivity extends AppCompatActivity {
    private TextView tvEmail, tvName, tvNim, tvKelas;
    private ImageView imgBack;
    private Api api;
    private String token;
    ArrayList<String> roles;

    Long id;
    Button editProfile;

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

        token = getIntent().getStringExtra("TOKEN");
        roles = getIntent().getStringArrayListExtra("ROLES");
        tvEmail = findViewById(R.id.tvEmail);
        tvName = findViewById(R.id.tvName);
        tvNim = findViewById(R.id.tvNim);
        tvKelas = findViewById(R.id.tvKelas);
        editProfile = findViewById(R.id.editProfile);
        imgBack = findViewById(R.id.imageViewProfile);
        loadMyProfile();

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

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

    private void loadMyProfile() {
        Retrofit retrofit = ApiClient.getRetrofitInstance(token);
        Api api = retrofit.create(Api.class);
        Call<UserDto> call = api.myProfile(token); // Assuming myProfile endpoint exists

        call.enqueue(new Callback<UserDto>() {
            @Override
            public void onResponse(Call<UserDto> call, Response<UserDto> response) {
                if (response.isSuccessful() && response.body() != null) {
                    UserDto userDto = response.body();
                    tvEmail.setText(userDto.getEmail());
                    tvName.setText(userDto.getName());
                    tvNim.setText(userDto.getNim());
                    tvKelas.setText(userDto.getKelas());
                    id = userDto.getId().longValue();
                } else {
                    Toast.makeText(ProfileActivity.this, "Failed to load profile: " + response.code(),
                    Toast.LENGTH_SHORT).show();
                }
            }
        });

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

```

        Toast.makeText(ProfileActivity.this, "Error: " + t.getMessage(),
        Toast.LENGTH_SHORT).show();
    }
}

private void editProfile() {
    Intent intent = new Intent(ProfileActivity.this, EditProfileActivity.class);
    intent.putExtra("TOKEN", token);
    intent.putExtra("ID", id);
    intent.putStringArrayListExtra("ROLES", roles);
    startActivity(intent);
}

private void menu(){
    Intent intent = new Intent(ProfileActivity.this, DashboardActivity.class);
    intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP | Intent.FLAG_ACTIVITY_SINGLE_TOP);
    intent.putExtra("TOKEN", token);
    intent.putStringArrayListExtra("ROLES", roles);
    startActivity(intent);
}

}

```

Disini mengambil data dari user yang login menggunakan authentication di web service juga. Kemudian ditampilkan di layoutnya. Disini juga ada tombol edit untuk mengedit profil sehingga di intent ke EditProfileActivity

#### ◆ EditProfileActivity

```

package com.example.laporkelas.activity;

import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import com.example.laporkelas.R;
import com.example.laporkelas.api.Api;
import com.example.laporkelas.api.ApiClient;
import com.example.laporkelas.model.UserDto;

import java.util.ArrayList;

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

public class EditProfileActivity extends AppCompatActivity {
    private EditText etName, etEmail, etNim, etKelas, etPassword, etEmailValidation,
    etPasswordValidation;
    private Button btnUpdateProfile;
    private Api api;
    private Long userId;
    private String token;
    ArrayList<String> roles;

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

        // Initialize the EditTexts and Button
        etName = findViewById(R.id.editTextName);
        etEmail = findViewById(R.id.editTextEmail);
        etNim = findViewById(R.id.editTextNim);
        etKelas = findViewById(R.id.editTextKelas);
        etPassword = findViewById(R.id.editTextPassword);
        etEmailValidation = findViewById(R.id.editTextEmailValidation);
        etPasswordValidation = findViewById(R.id.editTextPasswordValidation);
        btnUpdateProfile = findViewById(R.id.buttonUpdateProfile);

        userId = getIntent().getLongExtra("ID", 0);
        token = getIntent().getStringExtra("TOKEN");
        roles = getIntent().getStringArrayListExtra("ROLES");

        // Initialize Retrofit and the API interface
        Retrofit retrofit = ApiClient.getRetrofitInstance(token);
        api = retrofit.create(Api.class);
    }
}

```

```

        btnUpdateProfile.setOnClickListener(v -> updateUserProfile());
    }

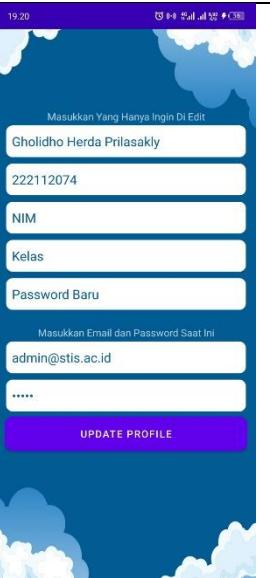
    private void updateUserProfile() {
        // Gather user input
        String name = etName.getText().toString();
        String email = etEmail.getText().toString();
        String nim = etNim.getText().toString();
        String kelas = etKelas.getText().toString();
        String password = etPassword.getText().toString();
        String emailValidation = etEmailValidation.getText().toString();
        String passwordValidation = etPasswordValidation.getText().toString();

        // Create a UserDto object with the user input
        UserDto userDto = new UserDto(name, email, nim, kelas, password);
        if (TextUtils.isEmpty(name)) {
            userDto.setName(null);
        }
        if (TextUtils.isEmpty(email)) {
            userDto.setEmail(null);
        }
        if (TextUtils.isEmpty(nim)) {
            userDto.setNim(null);
        }
        if (TextUtils.isEmpty(kelas)) {
            userDto.setKelas(null);
        }
        if (TextUtils.isEmpty(password)) {
            userDto.setPassword(null);
        }
        // Make the API call to update the user profile
        api.updateUser(userId, userDto, emailValidation, passwordValidation).enqueue(new
Callback<String>() {
    @Override
    public void onResponse(Call<String> call, Response<String> response) {
        if (response.isSuccessful()) {
            Toast.makeText(EditProfileActivity.this, "Profile updated successfully!",
Toast.LENGTH_SHORT).show();
            Intent intent = new Intent(EditProfileActivity.this, ProfileActivity.class);
            intent.putExtra("TOKEN", token);
            intent.putStringArrayListExtra("ROLES", roles);
            startActivity(intent);
        } else {
            Toast.makeText(EditProfileActivity.this, "Failed to update profile: " +
response.code(), Toast.LENGTH_SHORT).show();
        }
    }
    @Override
    public void onFailure(Call<String> call, Throwable t) {
        Toast.makeText(EditProfileActivity.this, "Update error: " + t.getMessage(),
Toast.LENGTH_SHORT).show();
    }
});
}
}

```

Disini memasukkan dengan methode patch hanya element tertentu saja yang mau diubah tidak perlu semuanya. Dan perlu memasukkan email serta current password sebagai validasi edit profile nya. Jika bisa dan berhasil akan kembali ke halaman profile.

◆ Contoh kegiatan

Tampilan saat mengklik profile akan menunjukkan data diri si login seperti di bawah ini.	Jika di klik edit, dia akan menampilkan kolom form untuk mengisi part yang akan di update.	Contoh inputan mengganti nama dan email *sudah direvisi <a href="mailto:222112074@stis.ac.id">222112074@stis.ac.id</a> tapi salah ambil gambar pak 😊	Contoh hasil update yang semula nama admin menjadi nama Gholidho Herda dan email nya juga berubah
			

Di dalam database nya pun juga berubah sehingga bisa dikatakan proses profile dan update profile sudah berhasil.

	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	<b>id</b>	<b>email</b>	<b>kelas</b>	<b>name</b>	<b>nim</b>	<b>password</b>
<input type="checkbox"/>	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	1	222112074@stis.ac.id	3SI2	Gholidho Herda Prilasakly	222112074	\$2a\$10\$XGVZ0WHk3bLkc2bG6BJWLl77xs0tne1QuVFFeGMGEz...
<input type="checkbox"/>	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	2	user@stis.ac.id	3SI2	User	222112075	\$2a\$10\$bdJIMbQYoyAkfLQBzBg/ZerWoQeuQy1z4668s4EwnGL...

## 8. Fitur Riwayat

### ◆ Api

```
@GET("/riwayat/all")
Call<List<RiwayatDto>> getAllRiwayats(@Header("Authorization") String authToken);

@GET("/riwayat/search")
Call<List<RiwayatDto>> findRiwayatByDateOrClass(
    @Header("Authorization") String authToken,
    @Query("date") String date,
    @Query("class") Long id
);

@DELETE("/riwayat/delete/{id}")
Call<String> deleteRiwayatById(@Path("id") Long id);
```

Disini dalam 1 rangkaian proses riwayat, ada menampilkan semua riwayat, search riwayat by date atau kelasnya, terus menghapus riwayat (khusus admin nanti). Karena dia butuh RiwayatDto maka harus membuat model RiwayatDto

### ◆ RiwayatDto

```
package com.example.laporkelas.model;

public class UserDto {
    private Long id;
    private String name;
    private String nim;
    private String kelas;
    private String email;
    private String password;

    public UserDto(String name, String email, String nim, String kelas, String password) {
        this.name = name;
        this.email = email;
        this.nim = nim;
        this.kelas = kelas;
        this.password = password;
    }

    // Getters and Setters
    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getNim() {
        return nim;
    }

    public void setNim(String nim) {
        this.nim = nim;
    }

    public String getKelas() {
        return kelas;
    }

    public void setKelas(String kelas) {
        this.kelas = kelas;
    }

    public String getEmail() {
        return email;
    }

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

    public String getPassword() {
```

```

        return password;
    }

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

```

## ◆ RiwayatActivity

```

package com.example.laporkelas.activity;

import android.content.Intent;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import com.example.laporkelas.R;
import com.example.laporkelas.adapter.RiwayatAdapter;
import com.example.laporkelas.api.Api;
import com.example.laporkelas.api.ApiClient;
import com.example.laporkelas.model.RiwayatDto;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.List;
import java.util.Locale;

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

public class RiwayatActivity extends AppCompatActivity implements RiwayatAdapter.OnRiwayatListener {

    private RecyclerView recyclerView;
    private RiwayatAdapter adapter;
    private String token;
    private EditText searchEditText;
    private Button searchButton;
    ArrayList<String> roles;

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

        recyclerView = findViewById(R.id.recyclerView);
        recyclerView.setLayoutManager(new LinearLayoutManager(this));

        searchEditText = findViewById(R.id.searchEditText);
        searchButton = findViewById(R.id.searchButton);

        token = getIntent().getStringExtra("TOKEN");
        roles = getIntent().getStringArrayListExtra("ROLES");
        searchButton.setOnClickListener(v -> performSearch());
        loadRiwayats();
    }

    private boolean isDate(String text) {
        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd", Locale.getDefault());
        try {
            dateFormat.parse(text);
            return true; // Parsing succeeded, the text is a valid date
        } catch (ParseException e) {
            return false; // Parsing failed, the text is not a valid date
        }
    }

    private void performSearch() {
        String searchText = searchEditText.getText().toString();
        if (isDate(searchText)) {
            searchRiwayats(searchText, null);
        } else {
            try {
                long classId = Long.parseLong(searchText);
                searchRiwayats(null, classId);
            } catch (NumberFormatException e) {
                loadRiwayats();
            }
        }
    }

    private void searchRiwayats(String date, Long classId) {
        ApiClient.getClient().getRiwayatService().searchRiwayat(date, classId).enqueue(new Callback<List<RiwayatDto>>() {
            @Override
            public void onResponse(Call<List<RiwayatDto>> call, Response<List<RiwayatDto>> response) {
                if (response.isSuccessful()) {
                    List<RiwayatDto> riwayatDtos = response.body();
                    if (riwayatDtos != null) {
                        adapter.setRiwayatDtos(riwayatDtos);
                    }
                }
            }

            @Override
            public void onFailure(Call<List<RiwayatDto>> call, Throwable t) {
                Toast.makeText(getApplicationContext(), "Error: " + t.getMessage(), Toast.LENGTH_SHORT).show();
            }
        });
    }

    private void loadRiwayats() {
        ApiClient.getClient().getRiwayatService().loadRiwayats().enqueue(new Callback<List<RiwayatDto>>() {
            @Override
            public void onResponse(Call<List<RiwayatDto>> call, Response<List<RiwayatDto>> response) {
                if (response.isSuccessful()) {
                    List<RiwayatDto> riwayatDtos = response.body();
                    if (riwayatDtos != null) {
                        adapter.setRiwayatDtos(riwayatDtos);
                    }
                }
            }

            @Override
            public void onFailure(Call<List<RiwayatDto>> call, Throwable t) {
                Toast.makeText(getApplicationContext(), "Error: " + t.getMessage(), Toast.LENGTH_SHORT).show();
            }
        });
    }
}

```

```

        }
    }

    private void loadRiwayats() {
        Retrofit retrofit = ApiClient.getRetrofitInstance(token);
        Api api = retrofit.create(Api.class);
        Call<List<RiwayatDto>> call = api.getAllRiwayats(token);

        call.enqueue(new Callback<List<RiwayatDto>>() {
            @Override
            public void onResponse(Call<List<RiwayatDto>> call, Response<List<RiwayatDto>> response) {
                if (response.isSuccessful() && response.body() != null) {
                    adapter = new RiwayatAdapter(response.body(), RiwayatActivity.this);
                    recyclerView.setAdapter(adapter);
                } else {
                    Toast.makeText(RiwayatActivity.this, "Error fetching data",
                    Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onFailure(Call<List<RiwayatDto>> call, Throwable t) {
                Toast.makeText(RiwayatActivity.this, "Network error: " + t.getMessage(),
                Toast.LENGTH_SHORT).show();
            }
        });
    }

    private void searchRiwayats(String date, Long classId) {
        Retrofit retrofit = ApiClient.getRetrofitInstance(token);
        Api api = retrofit.create(Api.class);

        Call<List<RiwayatDto>> call;

        if (date != null) {
            call = api.findRiwayatByDateOrClass(token, date, null);
        } else {
            call = api.findRiwayatByDateOrClass(token, null, classId);
        }

        call.enqueue(new Callback<List<RiwayatDto>>() {
            @Override
            public void onResponse(Call<List<RiwayatDto>> call, Response<List<RiwayatDto>> response) {
                if (response.isSuccessful() && response.body() != null) {
                    adapter = new RiwayatAdapter(response.body(), RiwayatActivity.this);
                    recyclerView.setAdapter(adapter);
                } else {
                    Toast.makeText(RiwayatActivity.this, "Error fetching data",
                    Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onFailure(Call<List<RiwayatDto>> call, Throwable t) {
                Toast.makeText(RiwayatActivity.this, "Network error: " + t.getMessage(),
                Toast.LENGTH_SHORT).show();
            }
        });
    }

    @Override
    public void onRiwayatClick(RiwayatDto riwayat) {
        if (userHasAdminRole()) {
            Intent intent = new Intent(this, RiwayatDetailsActivity.class);
            intent.putExtra("RiwayatDto", riwayat);
            intent.putExtra("TOKEN", token);
            startActivity(intent);
        } else {
            Toast.makeText(this, "Access Denied: You must be an admin to view details.",
            Toast.LENGTH_LONG).show();
        }
    }

    private boolean userHasAdminRole() {
        return roles != null && roles.contains("ROLE_ADMIN");
    }
}

```

Disini untuk menampilkan data membuat recyclerView sehingga perlu membuat kelas adapter. Nanti elemen adapter bisa di klik oleh **role\_admin** sehingga masuk ke halaman RiwayatDetailsActivity.

## ◆ RiwayatAdapter

```
package com.example.laporkelas.adapter;

import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.example.laporkelas.R;
import com.example.laporkelas.activity.RiwayatDetailsActivity;
import com.example.laporkelas.model.RiwayatDto;

import java.text.SimpleDateFormat;
import java.util.List;
import java.util.Locale;

public class RiwayatAdapter extends RecyclerView.Adapter<RiwayatAdapter.ViewHolder> {

    private final List<RiwayatDto> riwayatList;
    private final OnRiwayatListener onRiwayatListener;

    public interface OnRiwayatListener {
        void onRiwayatClick(RiwayatDto riwayat);
    }

    public RiwayatAdapter(List<RiwayatDto> riwayatList, OnRiwayatListener onRiwayatListener) {
        this.riwayatList = riwayatList;
        this.onRiwayatListener = onRiwayatListener;
    }

    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.item_riwayat, parent, false);
        return new ViewHolder(view, onRiwayatListener);
    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
        RiwayatDto riwayat = riwayatList.get(position);
        holder.bind(riwayat);
    }

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

    public class ViewHolder extends RecyclerView.ViewHolder {
        TextView reporterName, nim, reportDate, reportedClass, status;

        public ViewHolder(View itemView, OnRiwayatListener onRiwayatListener) {
            super(itemView);
            reporterName = itemView.findViewById(R.id.reporterNameTextView);
            nim = itemView.findViewById(R.id.nimTextView);
            reportDate = itemView.findViewById(R.id.reportDateTextView);
            reportedClass = itemView.findViewById(R.id.reportedClassTextView);
            status = itemView.findViewById(R.id.statusTextView);

            itemView.setOnClickListener(view -> {
                int position = getAdapterPosition();
                if (position != RecyclerView.NO_POSITION) {
                    onRiwayatListener.onRiwayatClick(riwayatList.get(position));
                }
            });
        }

        void bind(RiwayatDto riwayat) {
            reporterName.setText(riwayat.getReporterName());
            nim.setText(riwayat.getNim());

            SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd", Locale.getDefault());
            String formattedDate = dateFormat.format(riwayat.getReportDate());
            reportDate.setText(formattedDate);

            reportedClass.setText(String.valueOf(riwayat.getReportedClass()));
            status.setText(riwayat.getStatus());
        }
    }
}
```

## ◆ RiwayatDetailsActivity

```
package com.example.laporkelas.activity;

import android.content.Intent;
import android.os.Bundle;
import android.widget.TextView;
import android.widget.Button;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import com.example.laporkelas.R;
import com.example.laporkelas.model.RiwayatDto;
import com.example.laporkelas.api.Api;
import com.example.laporkelas.api.ApiClient;

import java.text.SimpleDateFormat;
import java.util.Locale;

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

public class RiwayatDetailsActivity extends AppCompatActivity {

    private TextView tvReporterName, tvNim, tvReportDate, tvReportedClass, tvStatus;
    private Button btnDelete;
    private RiwayatDto riwayat;

    String token;

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

        tvReporterName = findViewById(R.id.tvReporterName);
        tvNim = findViewById(R.id.tvNim);
        tvReportDate = findViewById(R.id.tvReportDate);
        tvReportedClass = findViewById(R.id.tvReportedClass);
        tvStatus = findViewById(R.id.tvStatus);

        riwayat = (RiwayatDto) getIntent().getSerializableExtra("RiwayatDto");
        token = getIntent().getStringExtra("TOKEN");

        if (riwayat != null) {
            displayRiwayatDetails(riwayat);
        }

        btnDelete = findViewById(R.id.btnDelete);
        btnDelete.setOnClickListener(v -> deleteRiwayat(riwayat.getId()));
    }

    private void displayRiwayatDetails(RiwayatDto riwayat) {
        tvReporterName.setText(riwayat.getReporterName());
        tvNim.setText(riwayat.getNim());

        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd", Locale.getDefault());
        String formattedDate = dateFormat.format(riwayat.getReportDate());
        tvReportDate.setText(formattedDate);

        tvReportedClass.setText(String.valueOf(riwayat.getReportedClass()));
        tvStatus.setText(riwayat.getStatus());
    }

    private void deleteRiwayat(Long id) {
        Api api = ApiClient.getRetrofitInstance(token).create(Api.class);
        Call<String> call = api.deleteRiwayatById(id);

        call.enqueue(new Callback<String>() {
            @Override
            public void onResponse(Call<String> call, Response<String> response) {
                if (response.isSuccessful()) {
                    Toast.makeText(RiwayatDetailsActivity.this, "Riwayat deleted successfully",
                    Toast.LENGTH_SHORT).show();
                    Intent intent = new Intent(RiwayatDetailsActivity.this, DashboardActivity.class);
                    intent.putExtra("TOKEN", token);
                    finish();
                } else {
                    Toast.makeText(RiwayatDetailsActivity.this, "Gagal Hapus Riwayat",
                    Toast.LENGTH_SHORT).show();
                    finish();
                }
            }
        });
    }
}
```

```

        @Override
        public void onFailure(Call<String> call, Throwable t) {
            Toast.makeText(RiwayatDetailsActivity.this, "Gagal Hapus Riwayat",
Toast.LENGTH_SHORT).show();
            finish();
        }
    });
}
}

```

◆ Contoh Riwayat

Tampilan saat masuk ke fitur Riwayat baik user maupun admin disini.	Saat melakukan fitur search dengan kelas 257 sebagai keyword	Saat klik element riwayat tapi sebagai user tidak bisa karena dibatasi admin	Kalau admin klik bakal muncul detail ini dan ada tombol delete	Kalau di klik delete langsung kehapus dan ini tampilan yang ini.
				

## 9. Fitur Konfirmasi (Khusus Admin)

### ◆ Api

```
@POST("/laporan/konfirmasi/{kelas}")
Call<String> konfirmasiPerubahan(
    @Path("kelas") Long kelas,
    @Body List<String> attributes
);
```

### ◆ KonfirmasiActivity

```
package com.example.laporkelas.activity;

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

import com.example.laporkelas.R;
import com.example.laporkelas.api.*;
import androidx.appcompat.app.AppCompatActivity;

import java.util.ArrayList;
import java.util.List;

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

public class KonfirmasiActivity extends AppCompatActivity {

    private EditText kelasEditText;
    private Button konfirmasiButton;
    private String token;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_konfirmasi); // You'll need to create this layout

        token = getIntent().getStringExtra("TOKEN");

        kelasEditText = findViewById(R.id.kelasEditText);
        konfirmasiButton = findViewById(R.id.konfirmasiButton);

        konfirmasiButton.setOnClickListener(v -> performKonfirmasi());
    }

    private void performKonfirmasi() {
        String kelasStr = kelasEditText.getText().toString();
        if (!kelasStr.isEmpty()) {
            try {
                long kelas = Long.parseLong(kelasStr);
                List<String> attributes = getAttributesToKonfirmasi(); // Implement this method based on
your UI
                sendKonfirmasiRequest(kelas, attributes);
            } catch (NumberFormatException e) {
                Toast.makeText(this, "Invalid class number", Toast.LENGTH_SHORT).show();
            }
        } else {
            Toast.makeText(this, "Please enter a class number", Toast.LENGTH_SHORT).show();
        }
    }

    private void sendKonfirmasiRequest(long kelas, List<String> attributes) {
        Retrofit retrofit = ApiClient.getRetrofitInstance(token);
        Api api = retrofit.create(Api.class);
        Call<String> call = api.konfirmasiPerubahan(kelas, attributes);

        call.enqueue(new Callback<String>() {
            @Override
            public void onResponse(Call<String> call, Response<String> response) {
                if (response.isSuccessful()) {
                    Toast.makeText(KonfirmasiActivity.this, "Success: " + response.body(),
                    Toast.LENGTH_SHORT).show();
                    Intent intent = new Intent(KonfirmasiActivity.this, LaporanActivity.class);
                    intent.putExtra("TOKEN", token);
                    startActivity(intent);
                } else {
                    Toast.makeText(KonfirmasiActivity.this, "Error: " + response.message(),
                    Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
}
```

```

        Toast.LENGTH_SHORT).show();
    }
}

@Override
public void onFailure(Call<String> call, Throwable t) {
    Toast.makeText(KonfirmasiActivity.this, "Network error: " + t.getMessage(), Toast.LENGTH_SHORT).show();
}
});

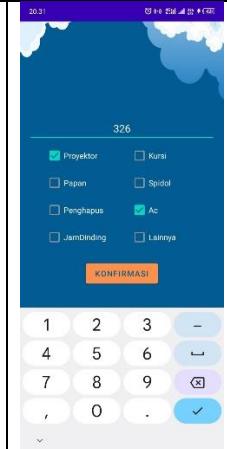
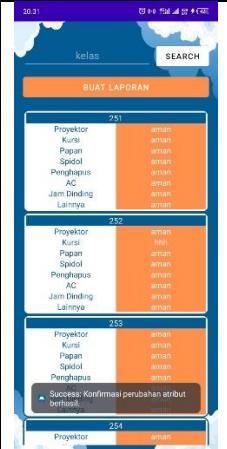
private List<String> getAttributesToKonfirmasi() {
    List<String> attributes = new ArrayList<>();
    if (((CheckBox) findViewById(R.id.checkBoxJamDinding)).isChecked()) {
        attributes.add("jamDinding");
    }
    if (((CheckBox) findViewById(R.id.checkBoxAc)).isChecked()) {
        attributes.add("ac");
    }
    if (((CheckBox) findViewById(R.id.checkBoxKursi)).isChecked()) {
        attributes.add("kursi");
    }
    if (((CheckBox) findViewById(R.id.checkBoxProyektor)).isChecked()) {
        attributes.add("proyektor");
    }
    if (((CheckBox) findViewById(R.id.checkBoxPapan)).isChecked()) {
        attributes.add("papan");
    }
    if (((CheckBox) findViewById(R.id.checkBoxSpidol)).isChecked()) {
        attributes.add("spidol");
    }
    if (((CheckBox) findViewById(R.id.checkBoxPenghapus)).isChecked()) {
        attributes.add("penghapus");
    }
    if (((CheckBox) findViewById(R.id.checkBoxLainnya)).isChecked()) {
        attributes.add("lainnya");
    }

    return attributes;
}
}

```

Disini memilih checkbox, dan jika di cek nanti elemen yang di cek akan berubah menjadi aman seperti sedia kala.

#### ◆ Contoh kegiatan

Lihat dulu misal kelas 326 ada Proyektor sama Ac yang bermasalah	Masuk konfirmasi dan cari kelas 326 terus cek proyektor dan ac nya	Kalau berhasil masuk halaman laporan dan muncul toast	Ketika di search semua sudah menjadi aman semuanya
			

## 10. Pengguna

### ◆ Api

```
@GET("/users/all")
Call<List<User>> getAllUsers(@Header("Authorization") String token);

@GET("/users/cari")
Call<List<User>> searchUsers(@Query("keyword") String keyword);
@DELETE("/users/delete/{userId}")
Call<Void> deleteUser(@Path("userId") Long userId);
```

Terlihat perlu menggunakan User model sehingga perlu membuat User Model

### ◆ User Model dan semua kelas terkait

```
package com.example.laporkelas.model;

import java.io.Serializable;
import java.util.HashSet;
import java.util.Set;

public class User implements Serializable {
    private Long id;
    private String name;
    private String nim;
    private String kelas;
    private String email;
    private String password;
    private Set<Role> roles = new HashSet<>();

    // Tambahkan konstruktor, getter, dan setter sesuai kebutuhan

    public User() {
        // Constructor kosong diperlukan untuk penggunaan ORM (misalnya, Room)
    }

    public User(Long id, String name, String nim, String kelas, String email, String password) {
        this.id = id;
        this.name = name;
        this.nim = nim;
        this.kelas = kelas;
        this.email = email;
        this.password = password;
    }

    // Getter dan Setter untuk properti lainnya

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getNim() {
        return nim;
    }

    public void setNim(String nim) {
        this.nim = nim;
    }

    public String getKelas() {
        return kelas;
    }

    public void setKelas(String kelas) {
        this.kelas = kelas;
    }

    public String getEmail() {
        return email;
    }

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

```

    }

    public String getPassword() {
        return password;
    }

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

    public Set<Role> getRoles() {
        return roles;
    }

    public void setRoles(Set<Role> roles) {
        this.roles = roles;
    }
}

```

```

package com.example.laporkelas.model;

public class Role {
    private Integer id;
    private ERole name;

    public Role() {
        // Constructor kosong diperlukan untuk penggunaan ORM (misalnya, Hibernate)
    }

    public Role(Integer id, ERole name) {
        this.id = id;
        this.name = name;
    }

    // Getter dan Setter

    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
    }

    public ERole getName() {
        return name;
    }

    public void setName(ERole name) {
        this.name = name;
    }
}

package com.example.laporkelas.model;

public enum ERole {
    ROLE_USER,
    ROLE_ADMIN
}

```

## ◆ UserActivity

```

package com.example.laporkelas.activity;

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

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

import com.example.laporkelas.R;
import com.example.laporkelas.adapter.UserAdapter;
import com.example.laporkelas.api.Api;
import com.example.laporkelas.api.ApiClient;
import com.example.laporkelas.model.User;

import java.util.ArrayList;
import java.util.List;

```

```

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

public class UsersActivity extends AppCompatActivity implements UserAdapter.OnUserListener {

    private RecyclerView recyclerView;
    private UserAdapter adapter;
    private String token;

    private EditText searchUserEditText;
    private Button searchUserButton;

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

        token = getIntent().getStringExtra("TOKEN");
        recyclerView = findViewById(R.id.recyclerViewUsers);
        recyclerView.setLayoutManager(new LinearLayoutManager(this));

        searchUserButton = findViewById(R.id.searchUserButton);
        searchUserEditText = findViewById(R.id.searchUserEditText);

        searchUserButton.setOnClickListener(v -> performSearch());
        loadUsers();
    }

    private void loadUsers() {
        Retrofit retrofit = ApiClient.getRetrofitInstance(token);
        Api api = retrofit.create(Api.class);
        Call<List<User>> call = api.getAllUsers(token);

        call.enqueue(new Callback<List<User>>() {
            @Override
            public void onResponse(Call<List<User>> call, Response<List<User>> response) {
                if (response.isSuccessful() && response.body() != null) {
                    adapter = new UserAdapter(response.body(), UsersActivity.this);
                    recyclerView.setAdapter(adapter);
                } else {
                    Toast.makeText(UsersActivity.this, "Error!!", Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onFailure(Call<List<User>> call, Throwable t) {
                Toast.makeText(UsersActivity.this, "Fatal Error", Toast.LENGTH_SHORT).show();
            }
        });
    }

    private void performSearch() {
        String searchText = searchUserEditText.getText().toString();
        if (!searchText.isEmpty()) {
            Retrofit retrofit = ApiClient.getRetrofitInstance(token);
            Api api = retrofit.create(Api.class);
            Call<List<User>> call = api.searchUsers(searchText);

            call.enqueue(new Callback<List<User>>() {
                @Override
                public void onResponse(Call<List<User>> call, Response<List<User>> response) {
                    if (response.isSuccessful() && response.body() != null) {
                        adapter.updateData(response.body());
                    } else {
                        Toast.makeText(UsersActivity.this, "No users found or error",
                        Toast.LENGTH_SHORT).show();
                    }
                }

                @Override
                public void onFailure(Call<List<User>> call, Throwable t) {
                    Toast.makeText(UsersActivity.this, "Network error: " + t.getMessage(),
                    Toast.LENGTH_SHORT).show();
                }
            });
        } else {
            loadUsers();
        }
    }

    @Override
    public void onUserClick(User user) {
        // Handle the user click event
        Intent intent = new Intent(this, UserDetailsActivity.class);
    }
}

```

```

        intent.putExtra("USER", user); // User should implement Serializable or Parcelable
        intent.putExtra("TOKEN", token);
        startActivity(intent);
    }
}

```

## ◆ UserAdapter

```

package com.example.laporkelas.adapter;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.example.laporkelas.R;
import com.example.laporkelas.model.User;

import java.util.List;

public class UserAdapter extends RecyclerView.Adapter<UserAdapter.ViewHolder> {

    private final List<User> userList;
    private OnUserListener onUserListener;

    public interface OnUserListener {
        void onUserClick(User user);
    }

    public UserAdapter(List<User> userList, OnUserListener onUserListener) {
        this.userList = userList;
        this.onUserListener = onUserListener;
    }

    public void updateData(List<User> newUserList) {
        userList.clear();
        userList.addAll(newUserList);
        notifyDataSetChanged();
    }

    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.item_user, parent, false);
        return new ViewHolder(view, onUserListener);
    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
        User user = userList.get(position);
        holder.itemView.setTag(user); // Set the tag for itemView with the user object
        holder.userNameTextView.setText(user.getName());
        holder.userEmailTextView.setText(user.getEmail());
        holder.userNimTextView.setText(user.getNim() + " -");
        holder.userKelasTextView.setText("- " + user.getKelas());
    }

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

    static class ViewHolder extends RecyclerView.ViewHolder {
        TextView userNameTextView, userEmailTextView, userNimTextView, userKelasTextView;

        ViewHolder(View itemView, OnUserListener onUserListener) {
            super(itemView);
            userNameTextView = itemView.findViewById(R.id.userNameTextView);
            userEmailTextView = itemView.findViewById(R.id.userEmailTextView);
            userNimTextView = itemView.findViewById(R.id.userNimTextView);
            userKelasTextView = itemView.findViewById(R.id.userKelasTextView);

            itemView.setOnClickListener(view -> {
                if (onUserListener != null && getAdapterPosition() != RecyclerView.NO_POSITION) {
                    onUserListener.onUserClick((User) itemView.getTag());
                }
            });
        }
    }
}

```

## ◆ UserDetailsActivity

```
package com.example.laporkelas.activity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.example.laporkelas.R;
import com.example.laporkelas.api.Api;
import com.example.laporkelas.api.ApiClient;
import com.example.laporkelas.model.User;
import com.example.laporkelas.adapter.*;

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

public class UserDetailsActivity extends AppCompatActivity {

    private TextView tvUserName, tvUserEmail, tvUserNim, tvUserKelas;
    private Button btnDeleteUser;
    private User user;
    private String token;

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

        // Initialize views
        tvUserName = findViewById(R.id.tvUserName);
        tvUserEmail = findViewById(R.id.tvUserEmail);
        tvUserNim = findViewById(R.id.tvUserNim);
        tvUserKelas = findViewById(R.id.tvUserKelas);
        btnDeleteUser = findViewById(R.id.btnDeleteUser);

        // Retrieve user from intent
        user = (User) getIntent().getSerializableExtra("USER");
        token = getIntent().getStringExtra("TOKEN");
        if (user != null) {
            displayUserDetails(user);
        }
    }

    btnDeleteUser.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (user != null) {
                deleteUser(user.getId());
            }
        }
    });
}

private void displayUserDetails(User user) {
    tvUserName.setText(user.getName());
    tvUserEmail.setText(user.getEmail());
    tvUserNim.setText(user.getNim());
    tvUserKelas.setText(user.getKelas());
}

private void deleteUser(Long userId) {
    // Retrofit instance
    Api api = ApiClient.getRetrofitInstance(token).create(Api.class);
    Call<Void> call = api.deleteUser(userId);

    call.enqueue(new Callback<Void>() {
        @Override
        public void onResponse(Call<Void> call, Response<Void> response) {
            if (response.isSuccessful()) {
                Toast.makeText(UserDetailsActivity.this, "User "+user.getName()+" Berhasil Dihapus",
                Toast.LENGTH_SHORT).show();
                Intent intent = new Intent(UserDetailsActivity.this, UserAdapter.class);
                finish(); // Close this activity
            } else {
                Toast.makeText(UserDetailsActivity.this, "Failed to delete user",
                Toast.LENGTH_SHORT).show();
            }
        }
    });
}
```

```

        @Override
        public void onFailure(Call<Void> call, Throwable t) {
            Toast.makeText(UserDetailsActivity.this, "Error: " + t.getMessage(),
Toast.LENGTH_SHORT).show();
        }
    });
}

```

◆ Contoh kegiatan

Tampilan ketika masuk menu pengguna.	Ketika melakukans search dengan berdasarkan nama	Ketika melakukans search dengan berdasarkan email
Ketika click elemen user muncul ini dan tombol delete.	Ketika berhasil di hapus lihat view profile lagi dia user sudah hilang dan tinggal admin.	Bonus ketika tombol log out di tekan.

## **LAMPIRAN**

Kurang lebih itu tadi fitur yang ada di android kali ini.

Untuk demo video dapat dilihat pada link berikut :

<https://drive.google.com/drive/folders/1HsMT4u7ByqUJvpenxsVp05AcRfRArfMf>

Untuk link full code dapat diakses pada :

[https://gitlab.com/pbo8573208/ppk3si2\\_222112074\\_gholidho-herda-prilasakly/-/tree/main/UAS-PPK?ref\\_type=heads](https://gitlab.com/pbo8573208/ppk3si2_222112074_gholidho-herda-prilasakly/-/tree/main/UAS-PPK?ref_type=heads)