

LAPORAN PROYEK UAS

“ Aplikasi Mobile Laboratorium Komputer Politeknik Statistika STIS”



Dosen Pengampu:

Ibnu Santoso, S.S.T., M.T

Disusun oleh:

Blessy Munthia Purba (222212537) (3SI2)

Kamis, 31 Januari 2025

**PROGRAM STUDI D-IV KOMPUTASI STATISTIK
POLITEKNIK STATISTIKA STIS
2025**

Soal:

Buatlah aplikasi mobile dengan spesifikasi sebagai berikut.

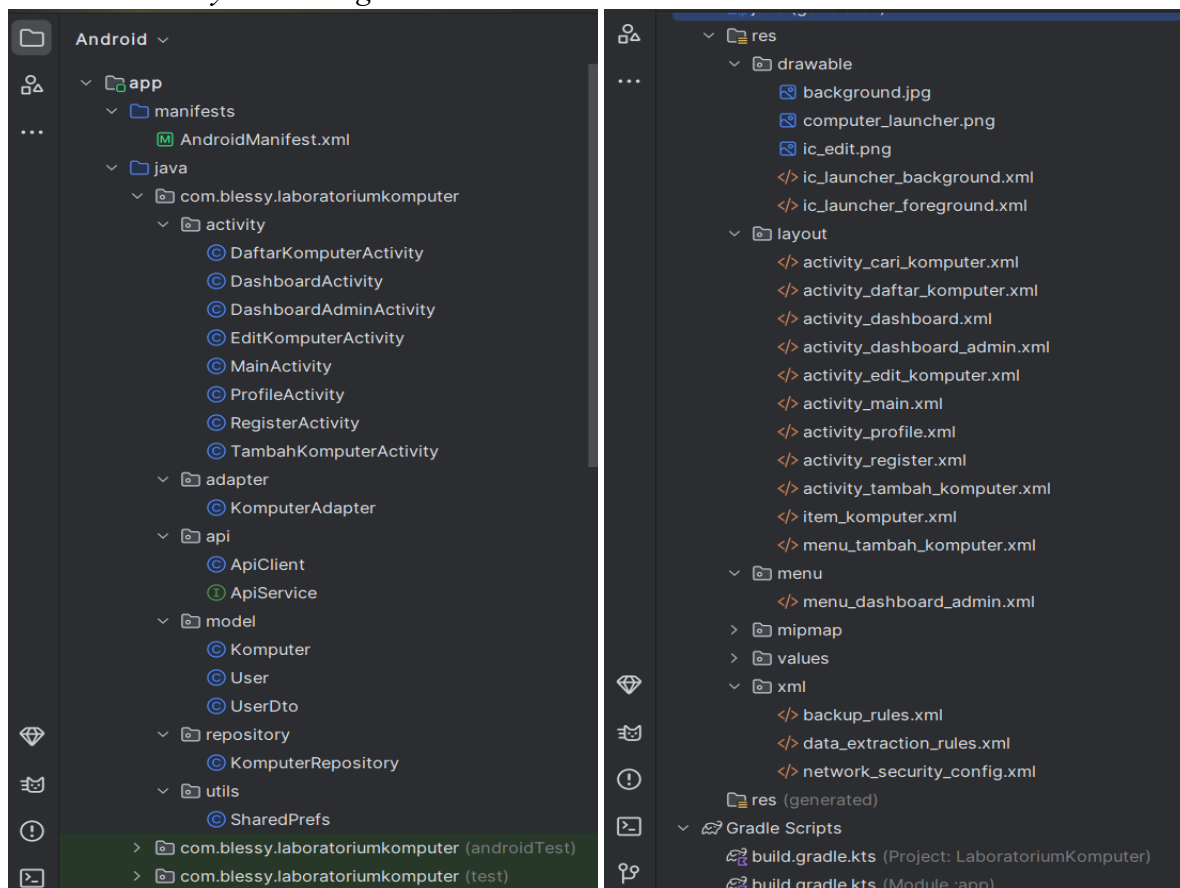
1. Pengguna aplikasi dapat melakukan register dan login serta dapat mengubah profil pengguna.
2. Tambahkan fitur lainnya sesuai dengan topik Anda minimal 3 activity/tampilan layar.

Ketentuan:

1. Aplikasi mobile yang dibuat harus menggunakan/mengonsumsi API dari webservice. Anda dapat menggunakan webservice yang telah Anda kerjakan pada Ujian Tengah Semester (UTS) atau dengan penyempurnaan.
2. Aplikasi dapat dibangun menggunakan native Android atau hybrid/cross-platform native.
3. Kumpulkan hasil pekerjaan Anda dalam laporan berformat PDF yang memuat informasi sebagai berikut:
 - a) Screenshot setiap tampilan aplikasi yang disertai dengan penjelasan penggunaannya dan penjelasan kode program yang terkait.
 - b) Tautan video capture aplikasi Anda yang menunjukkan semua fitur berjalan sesuai ketentuan.
 - c) link git

Tampilan Aplikasi dengan Penjelasan Kode Program:

1. Directory Kode Program:



2. *Penjelasan Singkat Keperluan Layanan Aplikasi*

Web Service dan Aplikasi Mobile Lab-Komputer Polstat STIS ini dibuat untuk keperluan monitoring perangkat elektronik (komputer) di Polstat STIS, yang akan menyediakan informasi dasar seperti IP Address, kapan dibelinya perangkat, masih bergaransi apa tidak, kapan terakhir pemeliharaan perangkat, dan kondisinya masih baik/rusak/perlu pemeliharaan. Di Aplikasi yang saya buat ini dapat melakukan register terlebih dahulu jika belum memiliki akun, yang digunakan oleh 3 role yaitu:

- 1). Admin: Pegawai/Staf IT di Polstat STIS
- 2). Mahasiswa : Angkatan 63-66 Polstat STIS
- 3). Pengamat: Pejabat/Petinggi di Polstat STIS

3. *Source Code ApiClient dan ApiService di package .api*

a. *ApiClient*

```
package com.blessy.laboratoriumkomputer.api;

import com.google.gson.Gson;
import com.google.gson.GsonBuilder;

import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

public class ApiClient {
    private static final String BASE_URL =
"http://192.168.0.114:8088/";
    private static Retrofit retrofit;

    // Menyediakan instance Retrofit
    public static Retrofit getClient() {
        if (retrofit == null) {
            Gson gson = new GsonBuilder()
                .setLenient() // Mengizinkan parsing JSON
                .create();

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

    // Menyediakan ApiService untuk digunakan di bagian lain
    aplikasi
    public static ApiService getApiService() {
        return getClient().create(ApiService.class);
    }
}
```

Pada kode di atas, Retrofit digunakan untuk melakukan permintaan HTTP dan menangani JSON. BASE_URL adalah URL dasar tempat semua API dimulai.

Karena masih memakai ipAddress (localhost laptop), maka API Web Service harus dijalankan dahulu untuk menjalankan Aplikasi Mobile. Gson digunakan untuk mengonversi data antara objek Java dan JSON. Sedangkan GsonConverterFactory memberi Retrofit kemampuan untuk mengonversi JSON menggunakan Gson. Sementara ApiClient adalah kelas untuk mengelola instance Retrofit dan menyediakan layanan API yang dapat digunakan di tempat lain dalam aplikasi.

b. *ApiService*

```
package com.blessy.laboratoriumkomputer.api;

import com.blessy.laboratoriumkomputer.model.Komputer;
import com.blessy.laboratoriumkomputer.model.User;
import java.util.List;
import okhttp3.ResponseBody;
import retrofit2.Call;
import retrofit2.http.Body;
import retrofit2.http.DELETE;
import retrofit2.http.GET;
import retrofit2.http.Header;
import retrofit2.http.POST;
import retrofit2.http.PUT;
import retrofit2.http.Path;
import retrofit2.http.Query;

public interface ApiService {

    @POST("auth/register")
    Call<ResponseBody> register(@Body User user);

    @POST("auth/login")
    Call<ResponseBody> login(@Body User user);

    @GET("komputer")
    Call<List<Komputer>> getKomputer(@Header("Authorization")
String token);

    @POST("komputer") // Sesuaikan dengan endpoint yang sesuai di
server Anda
    Call<ResponseBody> tambahKomputer(@Header("Authorization")
String token, @Body Komputer komputer);

    @GET("komputer/{id}")
    Call<Komputer> getKomputerById(
        @Header("Authorization") String token, // Token untuk
otentikasi
        @Path("id") Long id // ID alumni yang akan diambil
    );

    @PUT("komputer/{id}")
    Call<String> editKomputer(
        @Header("Authorization") String token,
        @Path("id") Long id,
        @Body Komputer komputer
    );

    @GET("komputer/cari")
    Call<List<Komputer>> cariKomputer(@Header("Authorization")
String authorization, @Query("keyword") String keyword);

    @GET("auth/profile")
```

```

        Call<User> getProfile(@Header("Authorization") String token);
// Menambahkan header untuk token

        @PUT("auth/profile/edit")
        Call<User> editProfile(@Header("Authorization") String token,
@Body User user);

        @DELETE("auth/profile/delete")
        Call<Void> deleteAccount(@Header("Authorization") String
token, @Body User user);
    }

```

4. Source Code SharedPreferences di package .utils

```

package com.blessy.laboratoriumkomputer.utils;

import android.content.Context;
import android.content.SharedPreferences;

public class SharedPreferences {
    private final SharedPreferences prefs;

    // Konstruktor yang menerima context untuk akses
    SharedPreferences
    public SharedPreferences(Context context) {
        // Menyimpan SharedPreferences dengan nama 'user_prefs' dan
        mode PRIVATE
        this.prefs = context.getSharedPreferences("user_prefs",
Context.MODE_PRIVATE);
    }

    // Menyimpan role pengguna (misalnya admin, user, dll)
    public void saveUserRole(String role) {
        SharedPreferences.Editor editor = prefs.edit();
        editor.putString("role", role);
        editor.apply(); // Gunakan apply() agar tidak terjadi
        blocking operasi
    }

    // Mendapatkan role pengguna, default null jika tidak ditemukan
    public String getUserRole() {
        return prefs.getString("role", null); // null sebagai default
    }

    // Menyimpan token autentikasi (misalnya token JWT)
    public void saveToken(String token) {
        SharedPreferences.Editor editor = prefs.edit();
        editor.putString("auth_token", token); // Menyimpan token
        dengan key 'auth_token'
        editor.apply(); // Gunakan apply() agar data tersimpan tanpa
        blocking UI thread
    }

    public void saveLoginStatus(boolean status) {
        SharedPreferences.Editor editor = prefs.edit();
        editor.putBoolean("is_logged_in", status);
        editor.apply();
    }
}

```

```

        // Mendapatkan token autentikasi, jika tidak ada mengembalikan
        null
        public String getToken() {
            return prefs.getString("auth_token", null); // Mengambil
            token, null jika tidak ditemukan
        }

        // Menyediakan method untuk memeriksa apakah token ada atau tidak
        public boolean isTokenExists() {
            return getToken() != null;
        }
    }
}

```

5. Source Code AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <!-- Permission for accessing external storage -->
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission
        android:name="android.permission.ACCESS_NETWORK_STATE" />

    <application
        android:networkSecurityConfig="@xml/network_security_config"
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.LaboratoriumKomputer"
        tools:targetApi="31">
        <activity
            android:name=".activity.MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER"
            />
            </intent-filter>
        </activity>
        <activity
            android:name=".activity.DashboardAdminActivity"
            android:label="Admin Dashboard"
            android:exported="false" />
        <activity
            android:name=".activity.RegisterActivity"
            android:label="Register"
            android:exported="false" />

        <activity
            android:name=".activity.TambahKomputerActivity"
            android:label="Tambah Data Komputer"

```

```

        android:exported="false" />

        <activity
            android:name=".activity.EditKomputerActivity"
            android:label="Edit Data Komputer"
            android:exported="false" />

    </application>

</manifest>

```

6. Source Code KomputerAdapter

```

package com.blessy.laboratoriumkomputer.adapter;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
import com.blessy.laboratoriumkomputer.R;
import com.blessy.laboratoriumkomputer.model.Komputer;
import com.blessy.laboratoriumkomputer.activity.EditKomputerActivity;

import java.util.List;

public class KomputerAdapter extends
RecyclerView.Adapter<KomputerAdapter.ViewHolder> {
    private List<Komputer> komputerList;
    private Context context;

    // Constructor untuk menerima context
    public KomputerAdapter(Context context) {
        this.context = context;
    }

    // Setter untuk list data
    public void setKomputerList(List<Komputer> komputerList) {
        this.komputerList = komputerList;
        notifyDataSetChanged(); // Notifikasi perubahan data
    }

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

    @Override
    public void onBindViewHolder(@NonNull ViewHolder holder, int
position) {

```

```

        Komputer komputer = komputerList.get(position);
        holder.ruangKomputer.setText(komputer.getRuangKomputer());
        holder.ipAddress.setText(komputer.getIpAddress());

        // Listener untuk tombol edit
        holder.editKomputerButton.setOnClickListener(v -> {
            Intent intent = new Intent(context,
EditKomputerActivity.class);
            intent.putExtra("id", komputer.getId()); // Kirim ID
komputer
            context.startActivity(intent);
        });
    }

    @Override
    public int getItemCount() {
        return (komputerList != null) ? komputerList.size() : 0;
    }

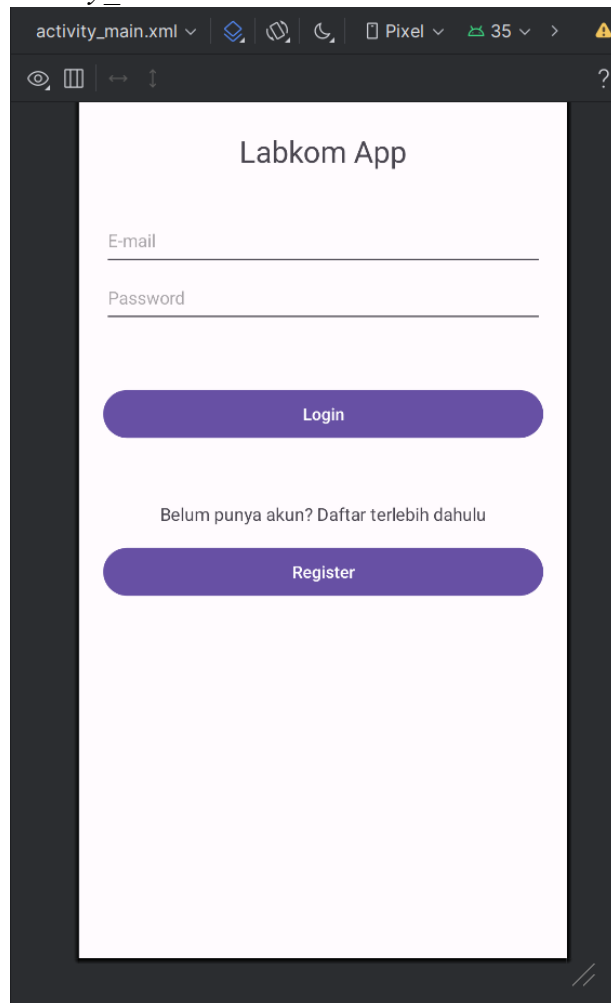
    // ViewHolder untuk menghubungkan layout item dengan adapter
    public static class ViewHolder extends RecyclerView.ViewHolder {
        TextView ruangKomputer, ipAddress;
        ImageView editKomputerButton;

        public ViewHolder(@NonNull View itemView) {
            super(itemView);
            ruangKomputer =
itemView.findViewById(R.id.ruangKomputer);
            ipAddress = itemView.findViewById(R.id.ipAddress);
            editKomputerButton =
itemView.findViewById(R.id.edit_komputer_button); // Pastikan ID
sesuai XML
        }
    }
}

```

Secara keseluruhan, KomputerAdapter berfungsi untuk mengelola data dan menampilkan informasi komputer di RecyclerView, serta memungkinkan interaksi seperti membuka halaman untuk mengedit data komputer, menyambungkan data dengan tampilan layout, dan mengubah tampilan item.

7. *Layout Tiap Activity*
a. *activity_main.xml*



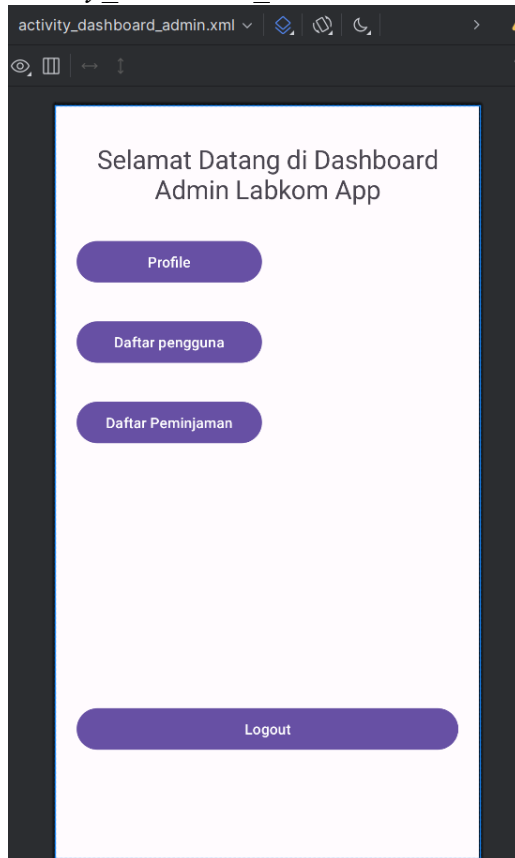
b. *activity_register.xml*

The screenshot displays a mobile application interface for account registration. The title "Registrasi Akun" is centered at the top. Below the title, there are six input fields stacked vertically, each with a label above it: "Nama", "Username", "E-mail", "Password", "NIM", and "Kelas". At the bottom of the form, there are two purple buttons: "Daftar" and "Kembali". The interface is shown within a dark-themed window with a toolbar at the top and a status bar at the bottom.

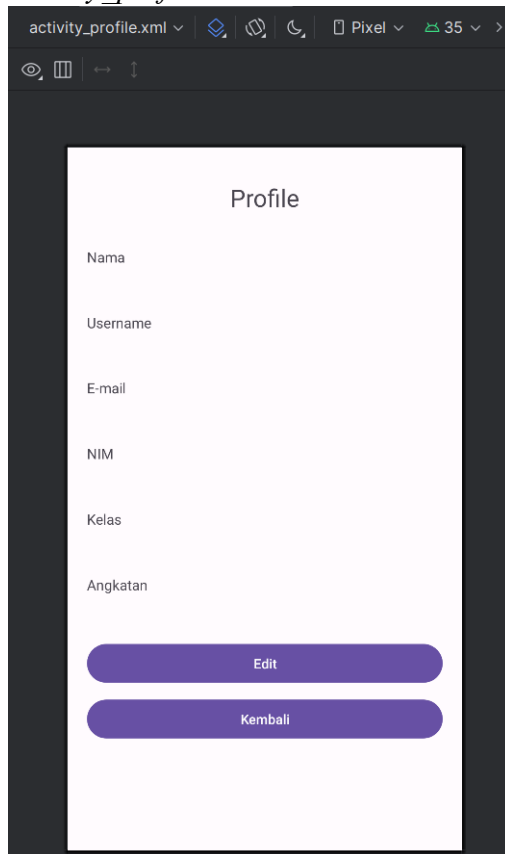
c. *activity_dashboard.xml*

The screenshot displays a mobile application interface for a dashboard. The title "Selamat Datang di Dashboard Labkom App" is centered at the top. Below the title, there is a purple button labeled "Profile". Underneath the button, the text "Daftar Komputer" is displayed. Below this text, there is a list of items labeled "Item 0" through "Item 9". At the bottom of the dashboard, there is a purple button labeled "Logout". The interface is shown within a dark-themed window with a toolbar at the top and a status bar at the bottom.

d. *activity_dashboard_admin.xml*



e. *activity_profile.xml*



f. *activity_edit_profile.xml*

Edit Profile

Nama

Username

E-mail

NIM

Kelas

Angkatan

Simpan Perubahan

Kembali

g. *activity_edit_komputer.xml*

Edit Komputer

IP Address

Ruang Komputer

Tanggal Beli

Garansi

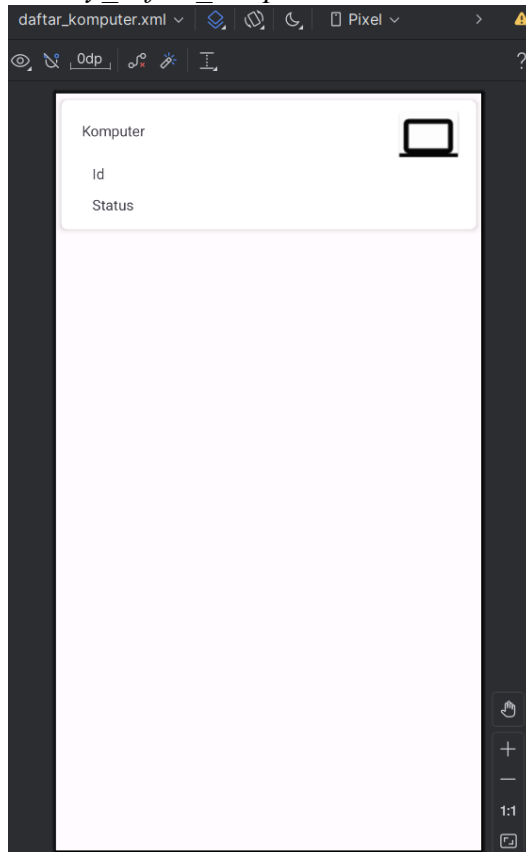
Pemeliharaan Terakhir

Kondisi

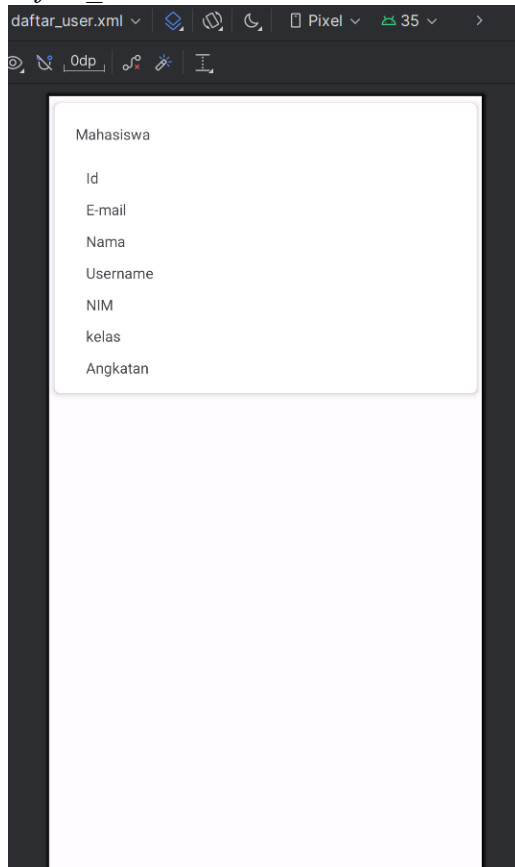
Simpan Perubahan

Kembali

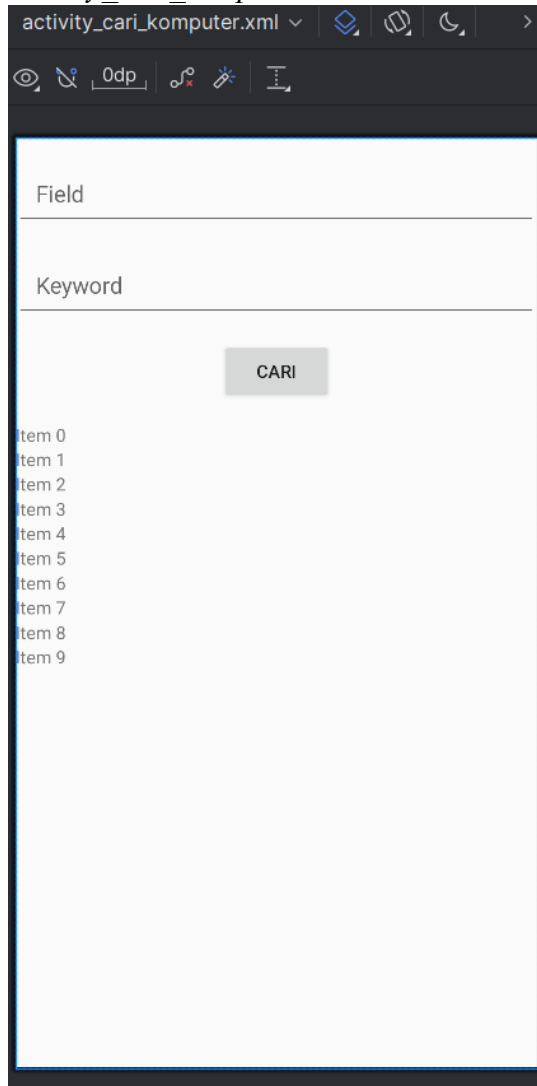
h. activity_daftar_komputer.xml



i. daftar_user.xml



j. *activity_cari_komputer.xml*



Alur Penjelasan Video Capture Aplikasi:

1. Menjelaskan entitas/model yang ada di aplikasi mobile lab-komputer
2. Menampilkan runing API Web Service di Postman sesuai method-method kegunaan Aplikasi Web Service
3. Menjelaskan sedikit penggunaan dan penjelasan kode program di Aplikasi Mobile
4. Menjalankan Aplikasi Mobile di Aplikasi Studio dan memperlihatkan runing yang masih eror.
5. Link tautan video: https://s.stis.ac.id/VideoCaptureLabkom_blessy

Link Git:

https://s.stis.ac.id/git_uas-ppk_blessy-munthia-purba

Lampiran: Hasil Running Web Service API Postman dan Database

1. Untuk Entitas/Model User

The image displays three screenshots of the Postman API client, each showing a successful POST request and its corresponding JSON response.

First Screenshot: Register

Request: POST `http://127.0.0.1:8088/register`

Body (raw):

```
1 "nomorInduk": "222212537",
2 "role": "USER",
3 "nama": "Bleci MP",
4 "email": "blecimp@gmail.com",
5 "password": "@31Juli2003",
6 "kelas": "3SI2",
7 "angkatan": 64,
8 "nomorWa": "082277490922"
```

Response (pretty):

```
1 "nomorInduk": "222212537",
2 "role": "USER",
3 "nama": "Bleci MP",
4 "email": "blecimp@gmail.com",
5 "password": "$2a$10$cXuidj5Dr6b4w/o9K4I",
6 "kelas": "3SI2",
7 "angkatan": 64,
8 "nomorWa": "082277490922",
9 "enabled": true,
10 "username": "222212537",
11 "accountNonLocked": true,
12 "authorities": null,
13 "accountNonExpired": true,
14 "credentialsNonExpired": true
```

Second Screenshot: Change Password

Request: POST `http://127.0.0.1:8088/changePassword`

Body (raw):

```
1 "nomorInduk": "222212537",
2 "password": "31Juli2003"
```

Response (pretty):

```
1 "nomorInduk": "222212537",
2 "role": "USER",
3 "nama": "Bleci MP",
4 "email": "blecimp@gmail.com",
5 "password": "$2a$10$xF.OZNmnbn5mPm4bz.",
6 "kelas": "3SI2",
7 "angkatan": 64,
8 "nomorWa": "082277490922",
9 "enabled": true,
10 "username": "222212537",
11 "accountNonLocked": true,
12 "authorities": null,
13 "accountNonExpired": true,
14 "credentialsNonExpired": true
```

Third Screenshot: Login

Request: POST `http://127.0.0.1:8088/login`

Body (raw):

```
1 "nomorInduk": "222212537",
2 "password": "31Juli2003"
```

Response (pretty):

```
1 "nomorInduk": "222212537",
2 "accessToken": "eyJhbGciOiJIUzUxMiJ9.eyJ1c2Vybm91dCI6ImBleCimpZm9udC5jb20iLCJ0eXBlIjoiVXNlciIsImVhbnQIOnR5bGU6ImF1dG8iLCJ1b3R0b28iOiJ222212537IiwiaWF0IjoxNjU0MjU0MDAwfQ."
3
```

Server: localhost:3306 » Database: lab_komputer » Table: user

Showing rows 0 - 1 (2 total, Query took 0.0077 seconds.)

CT * FROM `user`

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Options

	nomor_induk	angkatan	email	kelas	nama	nomor_wa	password
Edit Copy Delete	222112333	63	rani@gmail.com	4SI1	Rani Dingrat	082212331233	\$2a\$10\$SqetwEFs
Edit Copy Delete	222212537	64	blecimp@gmail.com	3SI2	Bleci MP	082277490922	\$2a\$10\$rFOZNmn

☐ Check all With selected: Edit Copy Delete Export

2. Untuk Entitas/Model Komputer

POST http://127.0.0.1:8088/komputer Send

Body raw JSON Beautify

```

1 {
2   "id": "1",
3   "ruangId": "243",
4   "namaKomputer": "Komputer
5     1",
6   "status": "TERSEDIA"

```

Body Pretty JSON

```

1 {
2   "ruangId": "243",
3   "namaKomputer": "Komputer 1",
4   "status": "TERSEDIA",
5   "_links": {
6     "self": {
7       "href": "http://127.0.0.1:8088/
8     },
9     "komputer": {
10      "href": "http://127.0.0.1:8088/
11    }
12  }
13 }

```

SELECT * FROM `komputer`

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	id	garansi	ip_address	kondisi	pemeliharaan_terakhir	ruang_komputer	tanggal_beli
<input type="checkbox"/> Edit Copy Delete	1	1	192.168.0.101	Baik	2024-01-15	Lab 1	2023-05-01
<input type="checkbox"/> Edit Copy Delete	2	0	192.168.0.102	Kurang Baik	2023-11-20	Lab 2	2023-03-22
<input type="checkbox"/> Edit Copy Delete	3	1	192.168.0.103	Baik	2024-01-10	Lab 3	2022-09-14
<input type="checkbox"/> Edit Copy Delete	4	1	192.168.0.104	Baik	2024-02-01	Lab 1	2023-06-01
<input type="checkbox"/> Edit Copy Delete	5	0	192.168.0.105	PERLU PEMELIHARAAN	2024-01-12	Lab 2	2022-12-01
<input type="checkbox"/> Edit Copy Delete	6	1	192.168.0.106	Baik	2024-01-25	Lab 3	2023-07-01
<input type="checkbox"/> Edit Copy Delete	7	0	192.168.0.107	RUSAK	2023-10-10	Lab 4	2021-10-01
<input type="checkbox"/> Edit Copy Delete	8	1	192.168.0.108	PERLU PEMELIHARAAN	2023-12-15	Lab 5	2022-04-01
<input type="checkbox"/> Edit Copy Delete	9	1	192.168.0.109	Baik	2024-02-01	Lab 1	2023-09-01
<input type="checkbox"/> Edit Copy Delete	10	0	192.168.0.110	PERLU PEMELIHARAAN	2023-11-20	Lab 2	2023-02-01
<input type="checkbox"/> Edit Copy Delete	11	1	192.168.0.111	Baik	2024-01-18	Lab 3	2023-01-01
<input type="checkbox"/> Edit Copy Delete	12	0	192.168.0.112	RUSAK	2023-09-10	Lab 4	2020-05-01
<input type="checkbox"/> Edit Copy Delete	13	1	192.168.0.113	Baik	2024-01-10	Lab 5	2021-12-01
<input type="checkbox"/> Edit Copy Delete	14	0	192.168.0.114	PERLU PEMELIHARAAN	2023-10-25	Lab 1	2021-03-01
<input type="checkbox"/> Edit Copy Delete	15	1	192.168.0.115	Baik	2024-01-30	Lab 2	2022-01-01

☐ Check all With selected: Edit Copy Delete Export