

UTS PEMROGRAMAN MOBILE 1

Ditujukan untuk Memenuhi Salah Satu Tugas UTS Pemrograman Mobile 1

Dosen Pengampu : Nova Agustina, M.kom..



Disusun oleh :

Muhammad Nizham Hibatullah

(23552011241)

TIF RP 23 CID B

TEKNIK INFORMATIKA

UNIVERSITAS TEKNOLOGI BANDUNG

2025

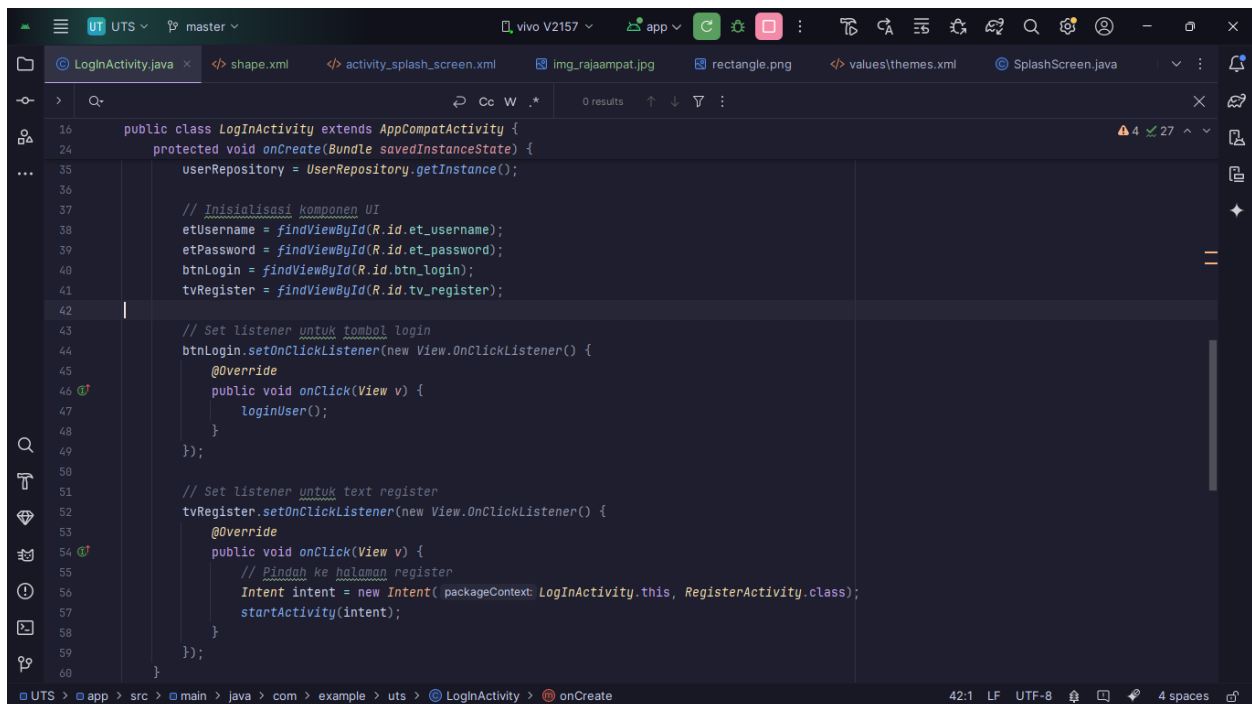
Essay

1. Apa fungsi setOnClickListener?

setOnClickListener adalah method yang digunakan untuk menangani klik pada sebuah komponen UI, seperti Button, ImageView, atau TextView. Saat pengguna mengklik elemen tersebut, kode di dalam listener akan dijalankan.

2. Apa syarat pemanggilan method setOnClickListener? Buat contohnya dan screenshot source code nya!

- Komponen seperti button harus sudah diinisiasi dengan benar menggunakan findViewById
- View tidak boleh null, Pastikan view tersebut memang ada dalam layout XML dan ID nya benar.
- Pemanggilan setOnClickListener dilakukan setelah setContentView()

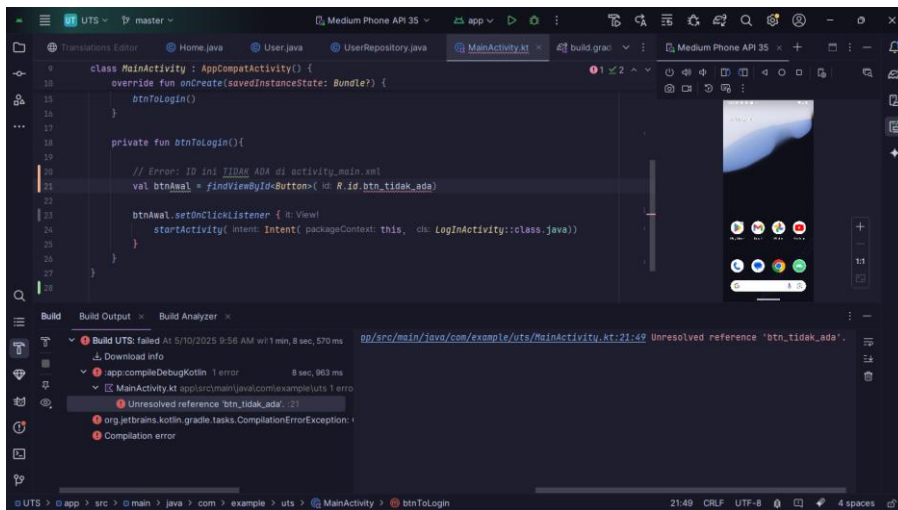


```
16 public class LoginActivity extends AppCompatActivity {
24     protected void onCreate(Bundle savedInstanceState) {
35         userRepository = UserRepository.getInstance();
36
37         // Inisialisasi komponen UI
38         etUsername = findViewById(R.id.et_username);
39         etPassword = findViewById(R.id.et_password);
40         btnLogin = findViewById(R.id.btn_login);
41         tvRegister = findViewById(R.id.tv_register);
42
43         // Set listener untuk tombol login
44         btnLogin.setOnClickListener(new View.OnClickListener() {
45             @Override
46             public void onClick(View v) {
47                 loginUser();
48             }
49         });
50
51         // Set listener untuk text register
52         tvRegister.setOnClickListener(new View.OnClickListener() {
53             @Override
54             public void onClick(View v) {
55                 // Pindah ke halaman register
56                 Intent intent = new Intent(packageContext, RegisterActivity.class);
57                 startActivity(intent);
58             }
59         });
60     }
}
```

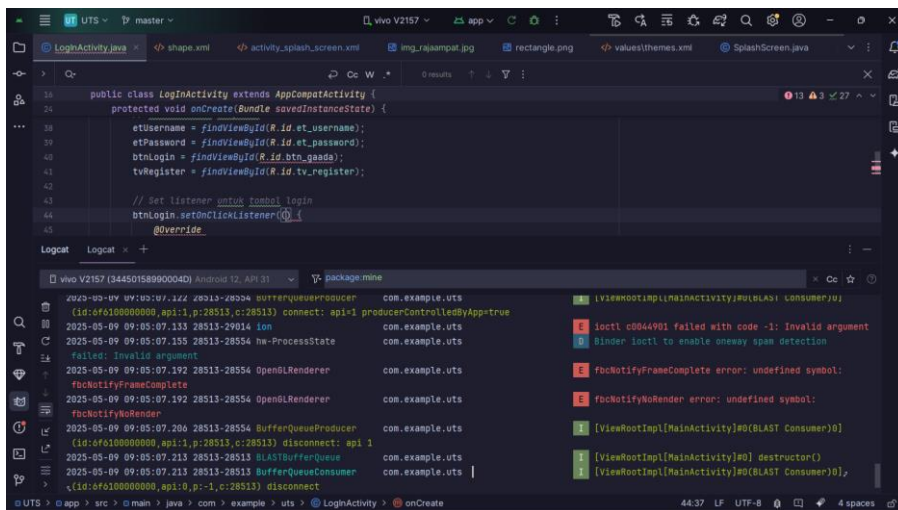
3. Error apa yang terjadi jika file kotlin salah menginisialisasi findViewById atau objek pada xml belum diinisialisasi? Screenshot logcat-nya!

Jika dalam file Kotlin terjadi kesalahan saat menginisialisasi `findViewById` atau objek pada XML belum diinisialisasi (misalnya, ID tidak ditemukan atau `findViewById` dipanggil sebelum `setContentView`), maka error yang biasanya terjadi adalah **NullPointerException** atau **Unresolved reference**

Ini Errornya



Ini logcatnya



4. Buat sebuah contoh program untuk menampilkan pesan error **NullPointerException**!

`import android.os.Bundle`

```
import android.widget.TextView
```

```
import androidx.appcompat.app.AppCompatActivity
```

```
class MainActivity : AppCompatActivity() {
```

```
    private var textView: TextView? = null
```

```
    override fun onCreate(savedInstanceState: Bundle?) {
```

```
        super.onCreate(savedInstanceState)
```

```
        // Kesalahan: Tidak memanggil setContentView terlebih dahulu
```

```
        // setContentView(R.layout.activity_main) // INI TIDAK ADA!
```

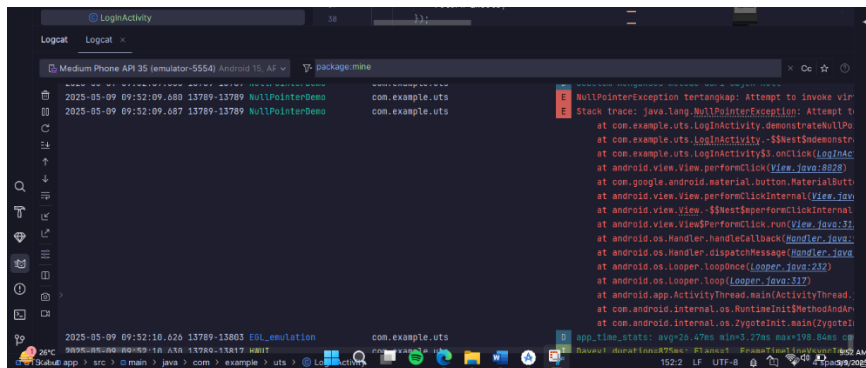
```
        textView = findViewById(R.id.textView) // Akan menghasilkan null
```

```
        textView!!.text = "Halo Dunia" // NullPointerException terjadi di sini
```

```
    }
```

```
}
```

Contoh error jika di logcat



Studi Kasus

a. Pengertian Aplikasi

Aplikasi ini merupakan aplikasi yang bernama Skald, diambil dari bahasa bangsa Viking. Aplikasi ini adalah aplikasi yang digunakan untuk memesan tiket pariwisata. Aplikasi ini dirancang di Android Studio dan dirancancang dengan bahasa Kotlin+Java

b. Link Github dan Video

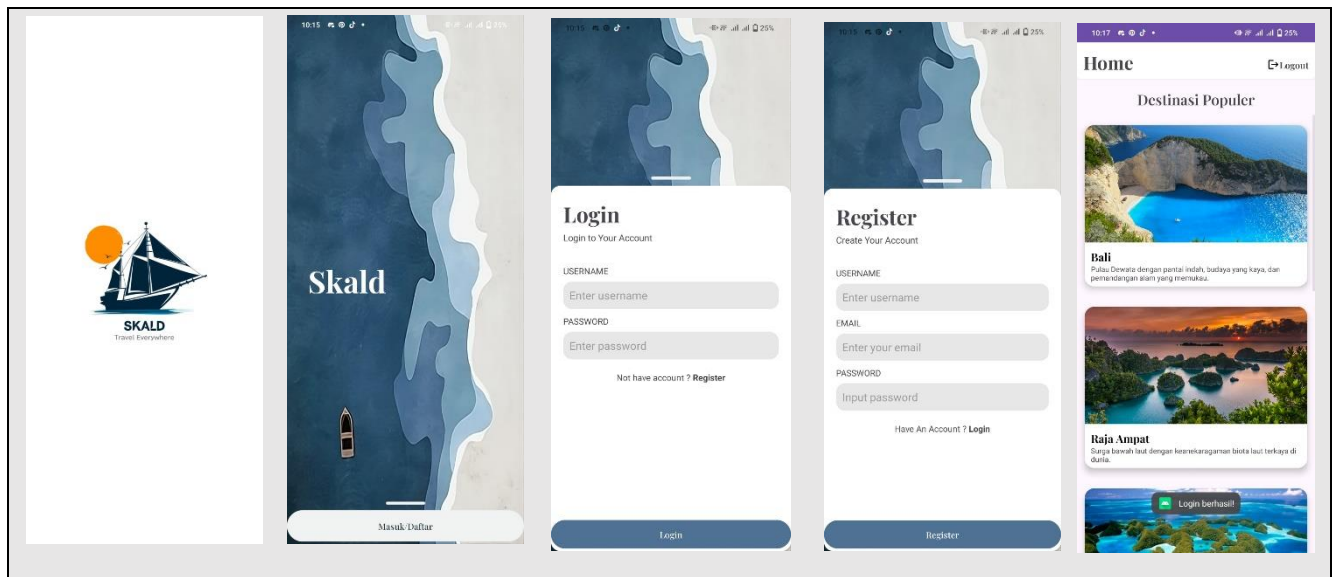
- Github

https://github.com/Nzzhm/UTS_AndroidStudio

- Drive

<https://drive.google.com/drive/folders/10tfWaDzrAWy49sR2Lp09S6FVVYLRHclI>

c. Gambar



d. Penjelasan Kode

1) SplashScreen.java

```
package com.example.uts
```

(Package tempat file ini berada)

```
import android.content.Intent
```

(Digunakan untuk berpindah antar activity)

```
import android.os.Bundle
```

(Digunakan untuk menyimpan data ketika activity dibuat)

```
import android.widget.Button
```

(Untuk menggunakan widget tombol dari layout)

```
import androidx.activity.enableEdgeToEdge
```

(Mengaktifkan tampilan layout full screen / edge-to-edge)

```
import androidx.appcompat.app.AppCompatActivity
```

(Activity dasar yang mendukung fitur Android modern)

```
class MainActivity : AppCompatActivity()
```

(Mendeklarasikan MainActivity sebagai turunan dari AppCompatActivity)

```
override fun onCreate(savedInstanceState: Bundle?)
```

(Fungsi utama yang pertama kali dijalankan saat activity dimulai)

```
super.onCreate(savedInstanceState)
```

(Memanggil fungsi onCreate dari class induk)

```
enableEdgeToEdge()
```

(Mengaktifkan tampilan layar penuh sampai ke tepi layar)

```
setContentView(R.layout.activity_main)
```

(Menetapkan tampilan/layout XML yang digunakan pada activity ini)

```
btnToLogin()
```

(Memanggil fungsi btnToLogin untuk menangani tombol)

```
private fun btnToLogin()
```

(Fungsi untuk mengatur aksi ketika tombol diklik)

```
val btnAwal = findViewById<Button>(R.id.btn_awal)
```

(Menghubungkan variabel dengan tombol dari layout yang memiliki ID btn_awal)

```
btnAwal.setOnClickListener{
```

(Menambahkan aksi ketika tombol ditekan)

```
startActivity(Intent(this, LogInActivity::class.java))
```

(Memulai activity baru yaitu LogInActivity)

2) **LoginActivity.java**

```
package com.example.uts;
```

```
import android.content.Intent;
```

```
import android.os.Bundle;
```

```
import android.widget.TextView;
```

```
import android.view.View;
```

```
import androidx.activity.EdgeToEdge;
```

```
import androidx.appcompat.app.AppCompatActivity;
```

```
import androidx.core.graphics.Insets;
```

```
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
```

```
public class LoginActivity extends AppCompatActivity
(Mendefinisikan class LoginActivity yang mewarisi AppCompatActivity)
```

```
private EditText etUsername, etPassword;
(Mendeklarasikan dua input field untuk username dan password)
```

```
private Button btnLogin;
(Mendeklarasikan tombol login)
```

```
private TextView tvRegister;
(Mendeklarasikan teks untuk navigasi ke halaman register)
```

```
private UserRepository userRepository;
(Mendeklarasikan objek penyimpanan dan pengecekan data user)
```

```
@Override
protected void onCreate(Bundle savedInstanceState)
(Fungsi yang dijalankan pertama kali saat activity dibuat)
```

```
super.onCreate(savedInstanceState);
(Memanggil onCreate dari superclass)
```

```
EdgeToEdge.enable(this);
(Mengaktifkan edge-to-edge layout untuk seluruh layar)
```



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

(Menetapkan layout XML untuk LoginActivity)

```
ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v,  
insets) -> {
```

(Mengatur agar layout tidak tertutup status bar atau navigation bar)

```
Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
```

(Mengambil ukuran dari status bar dan nav bar)

```
v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
```

(Memberikan padding agar konten tidak tertutup)

```
return insets;
```

(Mengembalikan nilai insets)

```
userRepository = UserRepository.getInstance();
```

(Mendapatkan instance dari class UserRepository)

```
etUsername = findViewById(R.id.et_username);
```

(Menghubungkan EditText untuk username dari layout)

```
etPassword = findViewById(R.id.et_password);
```

(Menghubungkan EditText untuk password dari layout)

```
btnLogin = findViewById(R.id.btn_login);
```

(Menghubungkan tombol login dari layout)

```
tvRegister = findViewById(R.id.tv_register);
```

(Menghubungkan teks "Register" dari layout)

```
btnLogin.setOnClickListener(new View.OnClickListener() {  
(Menambahkan event ketika tombol login diklik)
```

```
@Override  
public void onClick(View v) {  
loginUser();  
(Memanggil fungsi loginUser ketika tombol diklik)
```

```
tvRegister.setOnClickListener(new View.OnClickListener() {  
(Menambahkan event ketika teks register diklik)
```

```
@Override  
public void onClick(View v) {
```

```
Intent intent = new Intent(LoginActivity.this, RegisterActivity.class);  
(Membuat intent untuk berpindah ke RegisterActivity)
```

```
startActivity(intent);  
(Memulai activity RegisterActivity)
```

```
private void loginUser()  
(Fungsi untuk menangani proses login)
```

```
String username = etUsername.getText().toString().trim();  
(Mengambil teks username dari input dan menghapus spasi di awal/akhir)
```

```
String password = etPassword.getText().toString().trim();  
(Mengambil teks password dari input dan menghapus spasi di awal/akhir)
```

```
if (username.isEmpty() || password.isEmpty()) {
```

(Mengecek apakah username atau password kosong)

```
Toast.makeText(this, "Username dan password harus diisi",  
Toast.LENGTH_SHORT).show();
```

(Menampilkan pesan kesalahan)

```
return;
```

(Menghentikan proses login jika ada input kosong)

```
boolean loginSuccess = userRepository.loginUser(username, password);
```

(Mengecek apakah login berhasil melalui UserRepository)

```
if (loginSuccess) {
```

```
    Toast.makeText(this, "Login berhasil!", Toast.LENGTH_SHORT).show();
```

(Menampilkan pesan bahwa login berhasil)

```
    Intent intent = new Intent(LoginActivity.this, Home.class);
```

(Membuat intent untuk berpindah ke halaman Home)

```
    startActivity(intent);
```

(Memulai activity Home)

```
    finish();
```

(Menutup halaman login agar tidak bisa kembali ke sini)

```
} else {
```

```
    Toast.makeText(this, "Username atau password salah",
```

```
    Toast.LENGTH_SHORT).show();
```

(Menampilkan pesan kesalahan jika login gagal)

3) **RegisterActivity.java**

```
package com.example.uts;
```

(Package tempat class RegisterActivity berada)

```
import android.content.Intent;
```

(Digunakan untuk berpindah antar activity)

```
import android.os.Bundle;
```

(Menyimpan data saat activity dibuat)

```
import android.view.View;
```

(Digunakan untuk menangani interaksi tampilan)

```
import android.widget.TextView;
```

(Digunakan untuk menampilkan teks)

```
import androidx.activity.EdgeToEdge;
```

(Mengaktifkan tampilan edge-to-edge modern)

```
import androidx.appcompat.app.AppCompatActivity;
```

(Activity dasar dengan dukungan fitur modern Android)

```
import androidx.core.graphics.Insets;
```

(Digunakan untuk mengatur jarak dari sisi layar)

```
import androidx.core.view.ViewCompat;
```

(Dukungan kompatibilitas untuk tampilan)

```
import androidx.core.view.WindowInsetsCompat;
```

(Digunakan untuk mengatur padding agar tidak tertutup status/navigation bar)

```
import android.widget.Button;
```

(Widget tombol)

```
import android.widget.EditText;
```

(Widget input teks)

```
import android.widget.Toast;
```

(Untuk menampilkan pesan pop-up singkat)

```
public class RegisterActivity extends AppCompatActivity
```

(Menentukan class RegisterActivity sebagai turunan dari AppCompatActivity)

```
private EditText etUsername, etEmail, etPassword;
```

(Menentukan input username, email, dan password)

```
private Button btnRegister;
```

(Menentukan tombol register)

```
private TextView tvLogin;
```

(Menentukan teks untuk kembali ke halaman login)

```
private UserRepository userRepository;
```

(Menentukan class repository untuk menyimpan data user)

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState)
```

(Fungsi yang dipanggil pertama kali saat activity dijalankan)

```
super.onCreate(savedInstanceState);
```

(Memanggil method onCreate dari superclass)

```
EdgeToEdge.enable(this);
```

(Mengaktifkan edge-to-edge layout agar layout memenuhi seluruh layar)

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

(Menetapkan file layout XML untuk activity register)

```
ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v,  
insets) -> {
```

(Mengatur padding otomatis supaya tidak tertutup status bar atau navigation bar)

```
Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
```

(Mendapatkan ukuran status bar dan navigation bar)

```
v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
```

(Memberikan padding sesuai ukuran sistem bar)

```
return insets;
```

(Mengembalikan nilai insets)

```
userRepository = UserRepository.getInstance();
```

(Mengambil instance UserRepository untuk menyimpan data user)

```
etUsername = findViewById(R.id.et_username);
```

(Menghubungkan input username dari layout ke variabel)

```
etEmail = findViewById(R.id.et_email);
```

(Menghubungkan input email dari layout ke variabel)

```
etPassword = findViewById(R.id.et_password);
```

(Menghubungkan input password dari layout ke variabel)

```
btnRegister = findViewById(R.id.btn_register);
```

(Menghubungkan tombol register dari layout ke variabel)

```
tvLogin = findViewById(R.id.tv_login);
```

(Menghubungkan teks login dari layout ke variabel)

```
btnRegister.setOnClickListener(new View.OnClickListener() {
```

(Menambahkan aksi ketika tombol register diklik)

```
@Override
```

```
public void onClick(View v) {
```

```
registerUser();
```

(Memanggil fungsi registerUser ketika tombol diklik)

```
tvLogin.setOnClickListener(new View.OnClickListener() {
```

(Menambahkan aksi ketika teks "Login" diklik)

```
@Override
```

```
public void onClick(View v) {
```

```
finish();
```

(Menutup halaman Register dan kembali ke halaman Login)

```
private void registerUser()
```

(Fungsi untuk melakukan proses pendaftaran akun)

```
String username = etUsername.getText().toString().trim();
```

(Mengambil input username dan menghapus spasi di awal/akhir)

```
String email = etEmail.getText().toString().trim();
```

(Mengambil input email dan menghapus spasi di awal/akhir)

```
String password = etPassword.getText().toString().trim();
```

(Mengambil input password dan menghapus spasi di awal/akhir)

```
if (username.isEmpty() || email.isEmpty() || password.isEmpty()) {
```

(Mengecek jika ada input yang kosong)

```
Toast.makeText(this, "Semua field harus diisi", Toast.LENGTH_SHORT).show();
```

(Menampilkan pesan kesalahan)

```
return;
```

(Menghentikan proses registrasi)

```
if (!email.contains("@")) {
```

(Mengecek apakah format email mengandung '@')

```
Toast.makeText(this, "Format email tidak valid", Toast.LENGTH_SHORT).show();
```

(Menampilkan pesan kesalahan)

```
return;
```

(Menghentikan proses registrasi)

```
if (password.length() < 6) {
```

(Mengecek panjang password minimal 6 karakter)

```
Toast.makeText(this, "Password minimal 6 karakter",
```

```
Toast.LENGTH_SHORT).show();
```

(Menampilkan pesan kesalahan)

```
return;
```

(Menghentikan proses registrasi)


```
boolean registerSuccess = userRepository.registerUser(username, email, password);
```

(Mencoba menyimpan data user, dan menyimpan hasilnya ke variabel)

```
if (registerSuccess) {
```

```
    Toast.makeText(this, "Registrasi berhasil! Silakan login",
```

```
    Toast.LENGTH_SHORT).show();
```

(Menampilkan pesan bahwa registrasi berhasil)

```
    finish();
```

(Menutup halaman register untuk kembali ke login)

```
} else {
```

```
    Toast.makeText(this, "Username atau email sudah digunakan",
```

```
    Toast.LENGTH_SHORT).show();
```

(Menampilkan pesan kesalahan jika username/email sudah terdaftar)

4) **RegisterActivity.java** (untuk Array list)

```
package com.example.uts;
```

(Mendeklarasikan package tempat class berada)

```
import java.util.ArrayList;
```

(Mengimpor kelas ArrayList dari Java untuk menyimpan daftar user)

```
public class UserRepository {
```

(Mendeklarasikan class UserRepository)

```
    private static UserRepository instance;
```

(Menampung satu-satunya instance dari UserRepository menggunakan pola Singleton)

```
private ArrayList<User> userList;
```

(Menampung daftar user yang telah terdaftar)

```
private UserRepository() {
```

(Constructor dibuat private agar hanya bisa diakses dari dalam class ini)

```
userList = new ArrayList<>();
```

(Membuat objek ArrayList kosong untuk menyimpan user)

```
userList.add(new User("admin", "admin@example.com", "admin123"));
```

(Menambahkan user default/admin untuk testing awal)

```
}
```

```
public static UserRepository getInstance() {
```

(Metode untuk mengakses instance UserRepository)

```
if (instance == null) {
```

(Jika instance belum dibuat)

```
instance = new UserRepository();
```

(Buat instance baru dari UserRepository)

```
}
```

```
return instance;
```

(Mengembalikan instance yang sudah ada atau baru dibuat)

```
public boolean registerUser(String username, String email, String password) {
```

(Metode untuk mendaftarkan user baru)

```
for (User user : userList) {
```

(Melakukan perulangan untuk mengecek user yang sudah ada)

```
if (user.getUsername().equals(username) || user.getEmail().equals(email)) {
```

(Jika username atau email sudah terdaftar)

```
return false;
```

(Maka pendaftaran gagal, kembalikan false)

```
userList.add(new User(username, email, password));
```

(Jika lolos semua pengecekan, tambahkan user baru ke list)

```
return true;
```

(Kembalikan true sebagai tanda pendaftaran berhasil)

```
public boolean loginUser(String username, String password) {
```

(Metode untuk login user)

```
for (User user : userList) {
```

(Melakukan perulangan terhadap semua user)

```
if (user.getUsername().equals(username) && user.getPassword().equals(password)) {
```

(Jika username dan password cocok)

```
return true;
```

(Login berhasil, kembalikan true)

```
return false;
```

(Jika tidak ditemukan user yang cocok, login gagal)

```
public ArrayList<User> getAllUsers() {
```

(Metode untuk mendapatkan semua user — bisa untuk debugging)

```
return userList;
```

(Mengembalikan daftar user)

5) **RegisterActivity.java**

```
package com.example.uts;
```

(Menentukan bahwa file ini berada di package com.example.uts)

```
import android.os.Bundle;
```

(Mengimpor class Bundle untuk menyimpan state)

```
import androidx.appcompat.app.AppCompatActivity;
```

(Mengimpor AppCompatActivity sebagai induk dari activity ini)

```
public class Home extends AppCompatActivity {
```

(Mendeklarasikan class Home yang merupakan activity utama setelah login)

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
```

(Method yang dipanggil pertama kali saat activity dibuat)

```
super.onCreate(savedInstanceState);
```

(Memanggil implementasi superclass untuk membuat activity)

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

(Menghubungkan activity ini dengan layout XML bernama activity_home)