

LAPORAN TUGAS
PEMROGRAMAN ALGORITMA PEMROGRAMAN
SISTEM REGISTRASI KREDENSIAL PENGGUNA



Disusun Oleh:

Gina Ramadhani

Nim: 2511533014

Dosen Pengampu: DR. Wahyudi, S.T, M.T

Asisten Pratikum: Rahmad Dwi Rizki Olders

DAPERTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
TAHUN 2025

Soal:

Sistem Registrasi Kredensial Pengguna

Buatlah program untuk mensimulasikan proses registrasi akun baru. Program harus dapat mengambil input pengguna, menyimpannya dalam sebuah *object*, dan melakukan validasi data berdasarkan aturan yang ditentukan sebelum menampilkan hasilnya.

Struktur Program:

1. Class Data: Akun.java

- Class ini bertugas sebagai *blueprint* untuk menyimpan data akun.

- Atribut yang harus ada: username, password, email (semua String), dan pinAngka (int).
- Gunakan setter dan getter jika dibutuhkan untuk mengelola data privat tersebut.
- Wajib membuat *helper method* validasi berikut di dalam *class* ini:
 - a. `public boolean isPasswordValid()`: Mengembalikan true jika panjang password 8 karakter atau lebih.
 - b. `public boolean isEmailValid()`: Mengembalikan true jika email mengandung karakter @ dan “.”

2. Class Utama: Registrasi.java

- Class ini memiliki main method.
- Bertugas mengambil input, membuat *object* Akun, dan memanggil *setter*.
- Meminta input berikut dari pengguna:
 - a. Username (String)
 - b. Password (String)
 - c. Email (String)
 - d. PIN Angka (int, misal 6 digit)
- Melakukan pengecekan if-else dengan memanggil method `isPasswordValid()` dan `isEmailValid()` dari *object* Akun.

1. Pseudocode Program

Judul
Sistem Registrasi Kredensial Pengguna
Deklarasi
DECLARE username AS STRING DECLARE password AS STRING DECLARE email AS STRING DECLARE pin AS INTEGER DECLARE akun AS Akun
Pseudocode
// Mulai Program DISPLAY "=== SISTEM REGISTRASI ===" // Input Data INPUT username INPUT password INPUT email INPUT pin // Simpan Data ke Objek Akun CREATE akun AS Akun SET akun.username = username SET akun.password = password SET akun.email = email SET akun.pinAngka = pin // Validasi Password IF akun.isPasswordValid() = FALSE THEN DISPLAY "Error: Password minimal 8 karakter!" STOP PROGRAM END IF

```
// Validasi Email
```

```
IF akun.isEmailValid() = FALSE THEN
```

```
    DISPLAY "Error: Email harus mengandung '@' dan '!'"
```

```
    STOP PROGRAM
```

```
END IF
```

```
// Jika Validasi Lulus
```

```
DISPLAY "Registrasi Berhasil!"
```

```
DISPLAY "Username (lowercase): " + akun.username.toLowerCase()
```

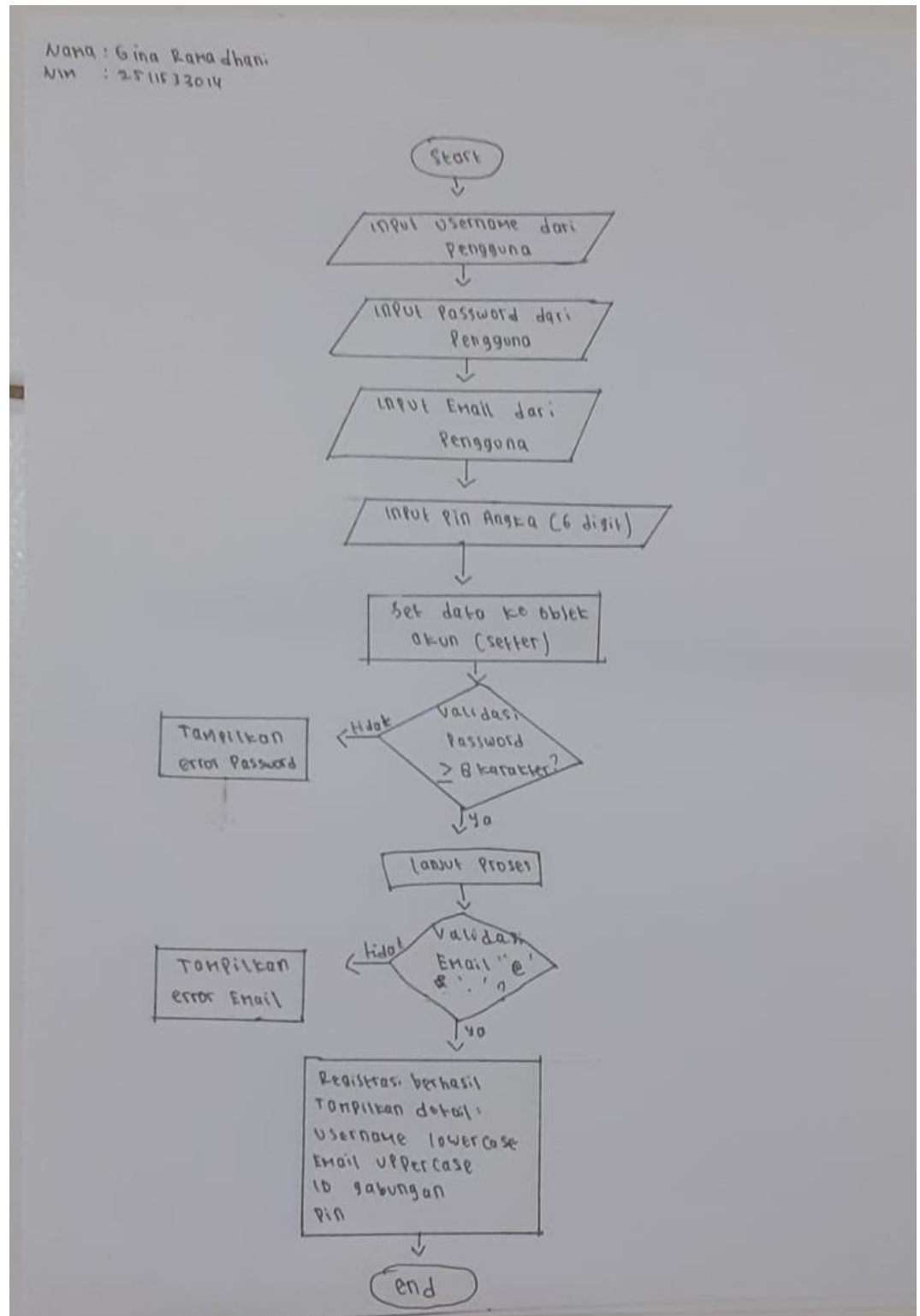
```
DISPLAY "Email (UPPERCASE): " + akun.email.toUpperCase()
```

```
DISPLAY "ID Gabungan (Username + PIN): " + akun.username +  
akun.pinAngka
```

```
DISPLAY "PIN: " + akun.pinAngka
```

```
END PROGRAM
```

2. Flowchart Program



3. Kode Program

a. Class akun

```
1 package pekan7_2511533014;
2
3 public class Akun {
4     private String username;
5     private String password;
6     private String email;
7     private int pinAngka;
8
9     // Setter
10    public void setUsername(String username) {
11        this.username = username;
12    }
13
14    public void setPassword(String password) {
15        this.password = password;
16    }
17
18    public void setEmail(String email) {
19        this.email = email;
20    }
21
22    public void setPinAngka(int pinAngka) {
23        this.pinAngka = pinAngka;
24    }
25
26    // Getter
27    public String getUsername() {
28        return username;
29    }
30
31    public String getPassword() {
32        return password;
33    }
34
35    public String getEmail() {
36        return email;
37    }
38
39    public int getPinAngka() {
40        return pinAngka;
41    }
42
43    // Validasi minimal
44    public boolean isPasswordValid() {
45        return password.length() >= 8;
46    }
47
48    public boolean isEmailValid() {
49        return email.contains("@") && email.contains(".");
50    }
51 }
52
53
54
```

b. Class tugasAlproPekan7_2511533014

```
1 package pekan7_2511533014;
2
3 import java.util.Scanner;
4 public class tugasAlproPekan7_2511533014 {
5     public static void main(String[] args) {
6
7         Scanner input = new Scanner(System.in);
8         Akun akun = new Akun();
9
10        System.out.println("==== REGISTRASI AKUN BARU =====");
11
12        System.out.print("Masukkan Username: ");
13        String username = input.nextLine();
14
15        System.out.print("Masukkan Password: ");
16        String password = input.nextLine();
17
18        System.out.print("Masukkan Email: ");
19        String email = input.nextLine();
20
21        System.out.print("Masukkan PIN (6 digit): ");
22        int pin = input.nextInt();
23
24        // Set data
25        akun.setUsername(username);
26        akun.setPassword(password);
27        akun.setEmail(email);
28        akun.setPinAngka(pin);
29
30        // Validasi
31        if (!akun.isPasswordValid()) {
32            System.out.println("\n--- REGISTRASI GAGAL ---");
33            System.out.println("Password Anda \"" + akun.getPassword() + "\" tidak valid (minimal 8 karakter).");
34            System.out.println("Silakan coba lagi.");
35            return;
36        }
37
38        if (!akun.isEmailValid()) {
39            System.out.println("\n--- REGISTRASI GAGAL ---");
40            System.out.println("Email Anda \"" + akun.getEmail() + "\" tidak valid (harus mengandung '@' dan '.').");
41            System.out.println("Silakan coba lagi.");
42            return;
43        }
44
45        // Jika lolos validasi:
46        System.out.println("\n--- REGISTRASI BERHASIL ---");
47        System.out.println("Akun untuk \"" + akun.getUsername() + "\" telah berhasil dibuat.");
48
49        // Detail Akun
50        System.out.println("\n--- Detail Akun ---");
51        System.out.println("Username (Lowercase): " + akun.getUsername().toLowerCase());
52        System.out.println("Email (Uppercase) : " + akun.getEmail().toUpperCase());
53        System.out.println("ID Pengguna (Gabungan): " + akun.getUsername() + akun.getPinAngka());
54
55        // Uji Tipe Data
56        System.out.println("\n--- Uji Tipe Data (PIN Anda: " + akun.getPinAngka() + ") ---");
57        System.out.println("PIN (int) + 10 = " + (akun.getPinAngka() + 10));
58        System.out.println("PIN (String) + \"10\" = " + (akun.getPinAngka() + "10"));
59
60        input.close();
61    }
62 }
63
```

4. Output Program

a. Output gagal validasi

```
<terminated> tugasAlproPekan7_2511533014 [Java Application] C:\Users\rara0\p2\pool\plugi
==== REGISTRASI AKUN BARU =====
Masukkan Username: gina
Masukkan Password: 12121212
Masukkan Email: nanananana
Masukkan PIN (6 digit): 456789
|
--- REGISTRASI GAGAL ---
Email Anda "nanananana" tidak valid (harus mengandung '@' dan '.').
Silakan coba lagi.
```

b. Output berhasil validasi

```
<terminated> tugasAlproPekan7_2511533014 [Java Application] C:\U
===== REGISTRASI AKUN BARU =====
Masukkan Username: gina
Masukkan Password: 12121212
Masukkan Email: nanananana200@gmail.com
Masukkan PIN (6 digit): 918273
|
--- REGISTRASI BERHASIL ---
Akun untuk "gina" telah berhasil dibuat.

--- Detail Akun ---
Username (Lowercase): gina
Email (Uppercase)      : NANANANANA200@GMAIL.COM
ID Pengguna (Gabungan): gina918273

--- Uji Tipe Data (PIN Anda: 918273) ---
PIN (int) + 10          = 918283
PIN (String) + "10"    = 91827310
```

5. Penjelasan Singkat Program

Program ini merupakan Sistem Registrasi Kredensial Pengguna yang berfungsi untuk menyimpan data akun baru, melakukan validasi, dan menampilkan hasil registrasi secara rapi. Program menggunakan dua class:

1. Class Akun (Class Data)

- Menyimpan atribut akun: username, password, email, dan pinAngka.
- Memiliki *setter* dan *getter* untuk mengakses data privat.
- Memiliki helper method validasi:
 - `isPasswordValid()` → Mengembalikan true jika password panjangnya ≥ 8 karakter.
 - `isEmailValid()` → Mengembalikan true jika email mengandung karakter @ dan ..

2. Class Registrasi (Class Utama/Driver)

- Mengambil input dari pengguna menggunakan *Scanner*.
- Membuat objek Akun dan menyimpan data input ke objek tersebut.

- Memanggil helper *method* untuk melakukan validasi password dan email.
- Menampilkan output sesuai hasil validasi:
 - Gagal validasi → Menampilkan pesan error yang jelas.
 - Berhasil validasi → Menampilkan detail akun dengan beberapa manipulasi string:
 - Username → lowercase
 - Email → UPPERCASE
 - ID gabungan → username + PIN

Logika Validasi

1. Password

- Program memanggil method `akun.isPasswordValid()`.
- Jika password kurang dari 8 karakter, program menampilkan pesan error:
"Error: Password minimal 8 karakter!"
- Jika password valid, program melanjutkan ke validasi email.

2. Email

- Program memanggil method `akun.isEmailValid()`.
- Jika email tidak mengandung @ atau ., program menampilkan pesan error:
"Error: Email harus mengandung '@' dan '!'"
- Jika email valid, program menampilkan semua data akun dengan manipulasi string sesuai format.

3. Output Berhasil

- Username → diubah ke lowercase menggunakan `toLowerCase()`
- Email → diubah ke UPPERCASE menggunakan `toUpperCase()`
- ID gabungan → digabungkan antara username + PIN
- PIN → ditampilkan apa adanya