

LAPORAN ALGORITMA DAN PEMROGRAMAN
TUGAS PEKAN 7 STRING PADA JAVA



Oleh :

DIKA GIWANDA

NIM 2511533025

Kelas :

A

Dosen Pengampu :

DR. WAHYUDI, S.T, M.T

Asisten Praktikum :

JOVANTRI IMMANUEL GULO

**FAKULTAS TEKNOLOGI INFORMASI
DEPARTEMEN INFORMATIKA
UNIVERSITAS ANDALAS
PADANG, NOVEMBER 2025**

Instruksi:

1. Buatlah program Java untuk studi kasus tunggal yang dijelaskan di bawah ini.
2. Program wajib dibuat menggunakan Scanner untuk input data dari pengguna.
3. Program wajib mengimplementasikan konsep:
 - Class & Object (minimal 2 class: 1 class data, 1 class utama/driver).
 - Method (termasuk setter/getter jika dibutuhkan, dan helper method untuk validasi).
 - Logika Percabangan (if-else atau switch-case).
 - Operasi dan manipulasi String (misalnya: .length(), .toUpperCase(), .toLowerCase(), .indexOf(), .concat(), + untuk string, dan escape character \").
4. Wajib menuliskan:
 - Pseudocode program
 - Flowchart program (ditulis tangan, difoto/scan, masukkan ke laporan)
 - Kode program Java (file .java)
 - Screenshot output program (1 kasus gagal validasi dan 1 kasus berhasil).
5. Simpan file Java dengan nama: tugasAlproPekan7_NIM.java (untuk class utama) dan Akun.java (untuk class data).
6. Buat laporan lengkap dalam format Word (.docx), kemudian konversi ke PDF dan unggah ke repository GitHub.
7. Logika validasi (aturan password/email) harus dijelaskan di laporan (ditulis di bagian pseudocode atau penjelasan singkat program).

Dan untuk soalnya gambar di bawah ini:

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.

Output:

1. Jika Gagal Validasi: Program harus menampilkan pesan error yang jelas.

Contoh:

```
--- REGISTRASI GAGAL ---
Email Anda "fufufafa67" tidak valid (harus mengandung '@' dan '.').
Silakan coba lagi.
```

2. Jika Berhasil Validasi: Program harus menampilkan semua detail akun dan hasil manipulasi String sesuai format di bawah ini.

```
===== REGISTRASI AKUN BARU =====
Masukkan Username: fufufafa67
Masukkan Password: 6767676767
Masukkan Email: fufufafa67@wapres.paman
Masukkan PIN (6 digit): 676767

--- REGISTRASI BERHASIL ---
Akun untuk "fufufafa67" telah berhasil dibuat.

--- Detail Akun ---
Username (Lowercase): fufufafa67
Email (Uppercase) : FUFUFAFA67@NAPRES.PAMAN
ID Pengguna (Gabungan): fufufafa67676767

--- Uji Tipe Data (PIN Anda: 676767) ---
PIN (int) + 10      = 676777
PIN (String) + "10" = 67676710
```

KODE PROGRAM

1. Class Data: Akun.java

```
package TugasPekan7_2511533025;

public class Akun_2511533025 {
    // Atribut privat
    private String username;
    private String password;
    private String email;
    private int pinAngka;

    // Konstruktor default
    public Akun_2511533025() {
        this.username = "";
        this.password = "";
        this.email = "";
        this.pinAngka = 0;
    }
    // Setter untuk username
    public void setUsername(String username) {
        this.username = username;
    }
    // Getter untuk username
    public String getUsername() {
        return username;
    }
    // Setter untuk password
    public void setPassword(String password) {
        this.password = password;
    }
    // Getter untuk password
    public String getPassword() {
        return password;
    }
    // Setter untuk email
    public void setEmail(String email) {
        this.email = email;
    }
    // Getter untuk email
    public String getEmail() {
        return email;
    }
    // Setter untuk pinAngka
    public void setPinAngka(int pinAngka) {
        this.pinAngka = pinAngka;
    }
    // Getter pin angka
    public int getPinAngka() {
        return pinAngka;
    }
    // Helper method: Validasi password (minimal 8 karakter)
    public boolean isPasswordValid() {
        return password.length() >= 8;
    }
    // Helper method: Validasi email (harus mengandung '@' dan '.')
    public boolean isEmailValid() {
```

```

        return email.contains("@") && email.contains(".");
    }

}

```

2. Class Utama: Registrasi

```

package TugasPekan7_2511533025;
import java.util.Scanner;

import UjiCobaTugasPekan7.Akun;
public class Registrasi_2511533025 {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("===== REGISTRASI AKUN BARU =====");
        // Input data dari pengguna
        System.out.print("Masukan Username : ");
        String inputUsername = scanner.nextLine();
        System.out.print("Masukan Password : ");
        String inputPassword = scanner.nextLine();
        System.out.print("Masukan Email : ");
        String inputEmail = scanner.nextLine();
        System.out.print("Masukan PIN (6 digit): ");
        int inputPin = scanner.nextInt();

        // Membuat objek Akun
        Akun akunBaru = new Akun();

        // Menggunakan setter untuk menyimpan data ke dalam objek
        akunBaru.setUsername(inputUsername);
        akunBaru.setPassword(inputPassword);
        akunBaru.setEmail(inputEmail);
        akunBaru.setPinAngka(inputPin);

        // Validasi menggunakan method dari objek Akun
        if (!akunBaru.isPasswordValid()) {
            System.out.println("\n--- REGISTRASI GAGAL ---");
            System.out.println("Password Anda \" " +
akunBaru.getPassword() + "\" tidak valid (harus minimal 8 karakter).");
            System.out.println("Silahkan coba lagi.");
            scanner.close();
            return; // Keluar jika password tidak valid
        }
        if (!akunBaru.isEmailValid()) {
            System.out.println("\n--- REGISTRASI GAGAL ---");
            System.out.println("Email Anda \" " + akunBaru.getEmail() +
"\\" tidak valid (harus mengandung '@' dan '.').");
            System.out.println("Silahkan coba lagi.");
            scanner.close();
            return; // Keluar jika email tidak valid
        }
        // Jika validasi berhasil
        System.out.println("\n--- REGISTRASI BERHASIL ---");
        System.out.println("Akun untuk \" " + akunBaru.getUsername() +
"\\" telah berhasil dibuat.\n");
    }
}

```

```
// Menampilkan detail akun sesuai format
System.out.println("--- Detail Akun ---");
System.out.println("Username (Lowercase) : " +
akunBaru.getUsername().toLowerCase());
System.out.println("Email (Uppercase) : " +
akunBaru.getEmail().toUpperCase());
System.out.println("ID Pengguna (Gabungan): " +
akunBaru.getUsername() + akunBaru.getPinAngka());
System.out.println("\n--- Uji Tipe Data (PIN Anda: " +
akunBaru.getPinAngka() + ") ---");
int pinInt = akunBaru.getPinAngka();
String pinString = String.valueOf(pinInt); // Konversi int ke
String
System.out.println("PIN (int) + 10      = " + (pinInt + 10));
System.out.println("PIN (String) + \"10\" = " + pinString +
"10");
scanner.close();

}

}
```

HASIL KODE PROGRAM

1. Jika Gagal Validasi

a. Jika Password Salah

```
===== REGISTRASI AKUN BARU =====
Masukan Username : Dika Giowanda
Masukan Password : 013579
Masukan Email    : dikagiowanda33@gmail.com
Masukan PIN (6 digit): 131517

--- REGISTRASI GAGAL ---
Password Anda "013579" tidak valid (harus minimal 8 karakter).
Silahkan coba lagi.
```

b. Jika Email Salah

```
===== REGISTRASI AKUN BARU =====
Masukan Username : Dika Giowanda
Masukan Password : 01357917
Masukan Email    : dikagiowanda33gmail.com
Masukan PIN (6 digit): 131517

--- REGISTRASI GAGAL ---
Email Anda "dikagiowanda33gmail.com" tidak valid (harus
mengandung '@' dan '.').
Silahkan coba lagi.
```

2. Jika Berhasil Validasi

```
===== REGISTRASI AKUN BARU =====
Masukan Username : Dika Giowanda
Masukan Password : 01357917
Masukan Email    : dikagiowanda33@gmail.com
Masukan PIN (6 digit): 131517

--- REGISTRASI BERHASIL ---
Akun untuk "Dika Giowanda" telah berhasil dibuat.
```

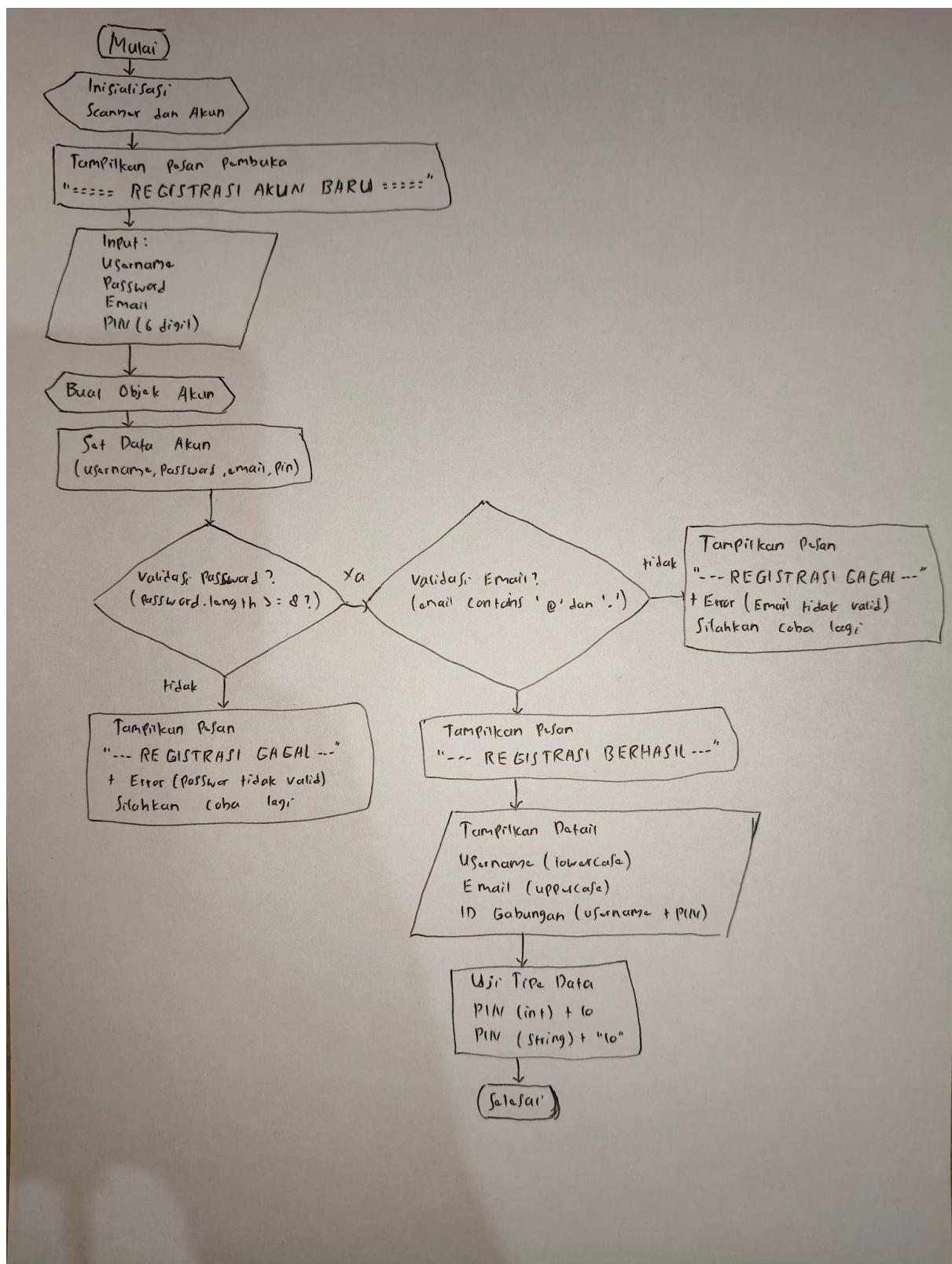
```
--- Detail Akun ---
Username (Lowercase) : dika giowanda
Email (Uppercase)    : DIKAGIOWANDA33@GMAIL.COM
ID Pengguna (Gabungan): Dika Giowanda131517
```

```
--- Uji Tipe Data (PIN Anda: 131517) ---
PIN (int) + 10      = 131527
PIN (String) + "10" = 13151710
```

BAHASA NATURAL

1. Mulai.
2. Inisialisai: siapkan objek akun untuk menyimpan data pengguna dan objek scanner untuk membaca input dari pengguna.
3. Tampilkan Header: Tampilkan pesan pembuka “===== REGISTRASI AKUN BARU =====” di layer.
4. Input Data Pengguna: Minta pengguna memasukkan empat informasi:
 - Username (sebagai teks)
 - Password (sebagai teks)
 - Email (sebagai teks)
 - PIN Angka (sebagai angka, misalnya 6 digit)
5. Buat Objek Akun: Buat sebuah objek baru dari kelas Akun.
6. Simpan Data ke Objek: Simpan semua data yang diinput oleh pengguna ke dalam objek Akun menggunakan metode setter (misalnya, setUsername, setPassword, setEmail, setPinAngka).
7. Validasi Password: Periksa apakah password yang dimasukan valid.
 - Jika password valid (panjangnya 8 karakter atau lebih), lanjut ke langkah 8.
 - Jika password tidak valid, tampilkan pesan error “--- REGISTRASI GAGAL ---” dan jelaskan bahwa password harus minimal 8 karakter, lalu hentikan program.
8. Validasi Email: Periksa apakah email yang dimasukan valid.
 - Jika email valid (mengandung karakter ‘@’ dan ‘.’), lanjut ke langkah 9.
 - Jika email tidak valid, tampilkan pesan error “--- REGISTRASI GAGAL ---” dan jelaskan bahwa email harus mengandung ‘@’ dan ‘.’, lalu hentikan program.
9. Tampilkan Pesan Sukses: Jika kedua validasi berhasil, tampilkan pesan “--- REGISTRASI BERHASIL ---” dan konfirmasi bahwa akun utnuk username tersebut telah berhasil dibuat.
10. Tampilkan Detail Akun: Tampilkan detail akun yang telah terdaftar dengan format tertentu:
 - Username dalam huruf kecil (lowercase)
 - Email dalam huruf besar (uppercase)
 - ID Pengguna yang merupakan gabungan dari username dan PIN.
11. Uji Tipe Data PIN: Lakukan dua operasi pada PIN untuk menunjukkan perbedaan tipe data:
 - Tambahkan nilai 10 ke PIN sebagai bilangan bulat (int) dan tampilkan hasilnya.
 - Gabungkan string “10” ke PIN sebagai teks (string) dan tampilkan hasilnya.
12. Selesai: Program selesai dijalankan.

FLOWCHART



BAHASA PSEUDOCODE

Judul:

Program Registrasi Akun Pengguna

{program untuk mensimulasikan proses pendaftaran akun baru dengan validasi data}

Deklarasi:

```
Var username : String  
Var password : String  
Var email : String  
Var pinAngka : Integer  
// objek  
Var akunBaru : Akun // Objek dari kelas Akun  
Var scanner : Scanner // Objek untuk input
```

Pseudocode:

1. mulai
2. Inisialisasi scanner // Siapkan scanner untuk input
3. Inisialisasi akunBaru. // Buat objek baru dari kelas Akun
4. Print “===== REGISTRASI AKUN BARU =====”
5. Print “Masukan Username:”
6. Input username
7. Print “Masukan Password:”
8. Input password
9. Print “Masukan Email”
10. Input email
11. Print “Masukan PIN (6 digit):”
12. Input pinAngka
13. akunBaru.setUsername(username)
14. akunBaru.setPassword(password)
15. akunBaru.setEmail(email)
16. akunBaru.setPinAngka(pinAngka)

17. if NOT akunBaru.isPasswordValid() Then
18. ‘Print “--- REGISTRASI GAGAL ---”
19. ‘Print “Password Anda \”” + akunBaru.getPassword() + “\” tidak valid (harus minimal 8 karakter).”
20. ‘Print “Silahkan coba lagi.”
21. ‘End’
22. End if
23. if NOT akunBaru.isEmailValid() Then
24. ‘Print “--- REGISTRASI GAGAL ---”
25. ‘Print “Email Anda \”” + akunBaru.getEmail() + “\” tidak valid (harus mengandung ‘@’ dan ‘.’).”
26. ‘Print “Silahkan coba lagi.”
27. ‘End’
28. End if
29. Print “--- REGISTRASI BERHASIL ---”
30. Print “Akun untuk \”” + akunBaru.getUsername() + “\” telah berhasil dibuat.
31. Print “--- Detail Akun ---”
32. Print “Username(Lowercase): “ + akunBaru.getUsername().toLowerCase()
33. Print “Email (Uppercase):”+ akunBaru.getEmail().toUpperCase()
34. Print “ID Pengguna (Gabungan): “ + akunBaru.getUsername() + akunBaru.getPinAngka()
35. Print “--- Uji Tipe Data (PIN Anda: “ + akunBaru.getPinAngka() + “)
36. pinInt ← akunBaru.getPinAngka()
37. pinString ← ConvertToString(pinInt)
38. Print “PIN (int) + 10 = “ + (pinInt + 10)
39. Print “PIN (String) + \”10\” = “ + pinString + “10”
40. Selesai

GAMBAR HASIL NGODING

1. Class Data: Akun

The screenshot shows the Eclipse IDE interface. The code editor displays a Java class named 'Akun_2511533025' with private attributes for username, password, email, and pinAngka, and corresponding getter and setter methods. It also includes helper methods for password and email validation. The terminal below shows the execution of the application, starting with a registration prompt and ending with a success message and a 'Detail Akun' command.

```
1 package TugasPekan7_2511533025;
2
3 public class Akun_2511533025 {
4     // Atribut privat
5     private String username;
6     private String password;
7     private String email;
8     private int pinAngka;
9
10    // Konstruktor default
11    public Akun_2511533025() {
12        this.username = "";
13        this.password = "";
14        this.email = "";
15        this.pinAngka = 0;
16    }
17    // Setter untuk username
18    public void setUsername(String username) {
19        this.username = username;
20    }
21    // Getter untuk username
22    public String getUsername() {
23        return username;
24    }
25    // Setter untuk password
26    public void setPassword(String password) {
27        this.password = password;
28    }
29    // Getter untuk password
30    public String getPassword() {
31        return password;
32    }
33    // Setter untuk email
34    public void setEmail(String email) {
35        this.email = email;
36    }
37    // Getter untuk email
38    public String getEmail() {
39        return email;
40    }
41    // Setter untuk pinAngka
42    public void setPinAngka(int pinAngka) {
43        this.pinAngka = pinAngka;
44    }
45    // Getter pin angka
46    public int getPinAngka() {
47        return pinAngka;
48    }
49    // Helper method: Validasi password (minimal 8 karakter)
50    public boolean isPasswordValid() {
51        return password.length() >= 8;
52    }
53    // Helper method: Validasi email (harus mengandung '@' dan '.')
54    public boolean isEmailValid() {
55        return email.contains "@" && email.contains ".";
56    }
57
58 }

```

Problems @ Javadoc Declaration Console X Progress Git S
<terminated> Registrasi_2511533025 [Java Application] /Applications/Eclipse.app/Cor
===== REGISTRASI AKUN BARU =====
Masukan Username : Dika Giowanda
Masukan Password : 01357917
Masukan Email : dikagiowanda33@gmail.com
Masukan PIN (6 digit): 131517

--- REGISTRASI BERHASIL ---
Akun untuk "Dika Giowanda" telah berhasil dibuat.

--- Detail Akun ---

2. Class Utama: Registrasi

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows files like DeklarasiVar..., HitungPertal..., OperatorAri..., Registrasi.java, Akun_2511533..., and Registrasi_2... (with a status of 39).
- Code Editor:** Displays the Java code for the `Registrasi` class. The code handles user input for username, password, email, and PIN, creates a new account object, and prints validation messages. It also includes validation logic for password length, email format, and PIN length.
- Output Console:** Shows the execution results:
 - User input: Masukan Username : Dika Giowanda, Masukan Password : 01357917, Masukan Email : dikagiowanda33@gmail.com, Masukan PIN (6 digit): 131517
 - Validation messages: "REGISTRASI GAGAL" for invalid password length and "REGISTRASI BERHASIL" for valid input.
 - Success message: "Akun untuk \"Dika Giowanda\" telah berhasil dibuat.\n"
 - Detailed account information printed at the end.

Outputnya

1. Jika Validasi Gagal

```
Problems @ Javadoc Declaration Console X Progress Git S
<terminated> Registrasi_2511533025 [Java Application] /Applications/Eclipse.app/Contents/Eclipse/plugins/org.eclipse.core.runtime/.cp...
===== REGISTRASI AKUN BARU =====
Masukan Username : Dika Giowanda
Masukan Password : 013579
Masukan Email : dikagiowanda33@gmail.com
Masukan PIN (6 digit): 131517

--- REGISTRASI GAGAL ---
Password Anda "013579" tidak valid (harus minimal 8 karakter).
Silahkan coba lagi.|
```

```
Problems @ Javadoc Declaration Console X Progress Git Staging i Install Java 2
<terminated> Registrasi_2511533025 [Java Application] /Applications/Eclipse.app/Contents/Eclipse/plugins/org.eclipse.core.runtime/.cp...
===== REGISTRASI AKUN BARU =====
Masukan Username : Dika Giowanda
Masukan Password : 01357917
Masukan Email : dikagiowanda33@gmail.com
Masukan PIN (6 digit): 131517

--- REGISTRASI GAGAL ---
Email Anda "dikagiowanda33@gmail.com" tidak valid (harus mengandung '@' dan '.').
Silahkan coba lagi.
```

2. Jika Validasi Berhasil

```
Problems @ Javadoc Declaration Console X Progress
<terminated> Registrasi_2511533025 [Java Application] /Applications/Eclipse.app/Contents/Eclipse/plugins/org.eclipse.core.runtime/.cp...
===== REGISTRASI AKUN BARU =====
Masukan Username : Dika Giowanda
Masukan Password : 01357917
Masukan Email : dikagiowanda33@gmail.com
Masukan PIN (6 digit): 131517
|
--- REGISTRASI BERHASIL ---
Akun untuk "Dika Giowanda" telah berhasil dibuat.

--- Detail Akun ---
Username (Lowercase) : dika giowanda
Email (Uppercase) : DIKAGIOWANDA33@GMAIL.COM
ID Pengguna (Gabungan): Dika Giowanda131517

--- Uji Tipe Data (PIN Anda: 131517) ---
PIN (int) + 10 = 131527
PIN (String) + "10" = 13151710
```

Penjelasan Singkat Kode Program

Kode program ini dirancang untuk mengumpulkan data registrasi (username, password, email, PIN) dari pengguna dan memvalidasi data tersebut sebelum mencetak konfirmasi registrasi yang berhasil. Program menggunakan paradigma pemrograman berorientasi objek (OOP) dengan mendefinisikan dua kelas:

1. Akun: Kelas ini menyimpan data pengguna (atribut privat) seperti username, password, email, dan PIN. Ia menyediakan metode setter dan getter untuk mengakses dan memanipulasi data ini.
2. Registrasi: Kelas utama yang berisi metode main. Di sinilah logika utama program berjalan: yaitu menerima input pengguna, membuat objek akun, menyimpan data ke objek tersebut, memanggil metode validasi, dan menampilkan pesan keberhasilan atau kegagalan serta detail akun.

Alur utamanya adalah:

1. Minta input dari pengguna
2. Simpan input ke dalam objek Akun.
3. Validasi:
 - Periksa apakah password memiliki panjang minimal 8 karakter.
 - Periksa apakah email mengandung karakter @ dan (.).
 - Jika salah satu validasi gagal, tampilkan pesan kesalahan dan hentikan program.
4. Jika semua validasi berhasil, tampilkan pesan bahwa registrasi berhasil.
5. Cetak detail akun dengan format tertentu (misalnya, username huruf kecil, email huruf besar).
6. Lakukan demonstrasi sederhana perbedaan antara tipe data int dan string pada PIN (misalnya, 123456 + 10 vs “123456” + “10”).

Secara keseluruhan, program ini menggabungkan konsep dasar Java seperti input/output, kelas dan objek, setter/getter, validasi logika, dan manipulasi string.