

LAPORAN TUGAS
ALGORITMA DAN PEMROGRAMAN
“IMPLEMENTASI CLASS, OBJECT, DAN VALIDASI DATA PADA SISTEM
REGISTRASI AKUN”



disusun Oleh:
Annisa Layli Ramadhani
2511532024
Informatika Kelas B

Dosen Pengampu: Wahyudi. Dr.. S.T.M.T
Asisten Pratikum: Rahmad Dwirizki Olders

DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
TAHUN 2025

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan laporan praktikum pemrograman Java ini dengan baik. Laporan ini disusun sebagai salah satu bentuk pertanggungjawaban kegiatan praktikum sekaligus untuk memperdalam pemahaman mengenai konsep dasar dan penerapan bahasa pemrograman Java. Dalam penyusunan laporan ini, penulis mengucapkan terima kasih kepada dosen pengampu, asisten laboratorium, serta rekan-rekan yang telah memberikan bimbingan, dukungan, dan kerja sama selama proses praktikum berlangsung. Penulis menyadari laporan ini masih jauh dari sempurna, sehingga kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa mendatang. Semoga laporan ini dapat memberikan manfaat dan menambah wawasan bagi penulis maupun pembaca.

Padang, 15 November 2025

Penulis

DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
BAB I PENDAHULUAN.....	4
1.1 Latar Belakang	4
1.2 Tujuan	5
1.3 Manfaat	5
BAB II PEMBAHASAN	6
2.1 Soal.....	6
2.2 Pseudocode	6
2.3 Flowchart	7
2.4 Kode Program.....	8
2.5 Output Program dan Penjelasan Program dan Logika Validasi.....	11
BAB III PENUTUP	14
3.1 Kesimpulan	14
3.2 Saran.....	14
DAFTAR PUSTAKA	15

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam pengembangan perangkat lunak, proses pengelolaan data pengguna merupakan salah satu aspek fundamental yang banyak diterapkan pada berbagai sistem, seperti aplikasi web, layanan digital, maupun sistem informasi internal. Salah satu komponen dasarnya adalah fitur registrasi akun, yang memungkinkan pengguna untuk membuat identitas digital agar dapat mengakses layanan yang tersedia. Pada proses registrasi tersebut, diperlukan pengolahan data seperti username, password, email, dan PIN, disertai mekanisme validasi untuk memastikan data yang dimasukkan memenuhi standar keamanan dan kelayakan penggunaan.

Bahasa pemrograman java menyediakan dukungan yang kuat dalam membangun sistem seperti ini melalui konsep Object-Oriented Programming (OOP). Dengan memanfaatkan class dan object, data pengguna dapat disimpan secara terstruktur serta diproses melalui method yang memiliki tanggung jawab tertentu. Selain itu, penggunaan string manipulation pada java memungkinkan program melakukan analisis dan pengolahan data teks, seperti memeriksa panjang password, mencari karakter tertentu di dalam email, serta menampilkan format teks dalam bentuk uppercase atau lowercase.

Melalui praktikum ini, kita mahasiswa mempelajari penerapan konsep OOP dalam membangun program sederhana yang mensimulasikan proses registrasi akun. Program dirancang untuk menerima input pengguna, menyimpannya dalam sebuah objek, serta melakukan validasi terhadap password dan email melalui method khusus di dalam class data (Akun). Praktikum ini juga memberikan pemahaman mengenai pentingnya validasi data sebagai bagian dari prinsip keamanan dan integritas informasi. Dengan demikian, kegiatan ini tidak hanya melatih kemampuan teknis pemrograman java, tetapi juga

mengembangkan pemahaman mengenai proses yang umum ditemui pada aplikasi nyata.

1.2 Tujuan

Tujuan dari tugas praktikum antara lain sebagai berikut:

- 1.2.1 Menerapkan konsep dasar pemrograman berorientasi objek (OOP) seperti class, object, atribut, method, serta enkapsulasi dalam pembuatan program sederhana.
- 1.2.2 Mengembangkan program registrasi akun yang mampu menerima input dari pengguna dan mengolah data tersebut secara terstruktur menggunakan class terpisah.
- 1.2.3 Menerapkan validasi data pada email dan password menggunakan helper method untuk memastikan data pengguna memenuhi aturan yang ditetapkan.
- 1.2.4 Meningkatkan kemampuan dalam menggunakan class scanner untuk menerima input dari pengguna secara interaktif.

1.3 Manfaat

Manfaat dari tugas praktikum antara lain sebagai berikut:

- 1.3.1 Memahami cara kerja sistem registrasi akun, yang merupakan fitur umum dalam berbagai aplikasi modern.
- 1.3.2 Mampu membangun struktur program yang lebih terorganisir melalui pemisahan antara class data (Akun) dan class utama (Registrasi).
- 1.3.3 Mengembangkan pemahaman terhadap manipulasi string, sehingga dapat mengolah dan menampilkan data teks secara efektif.

BAB II PEMBAHASAN

2.1 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.

2.2 Pseudocode

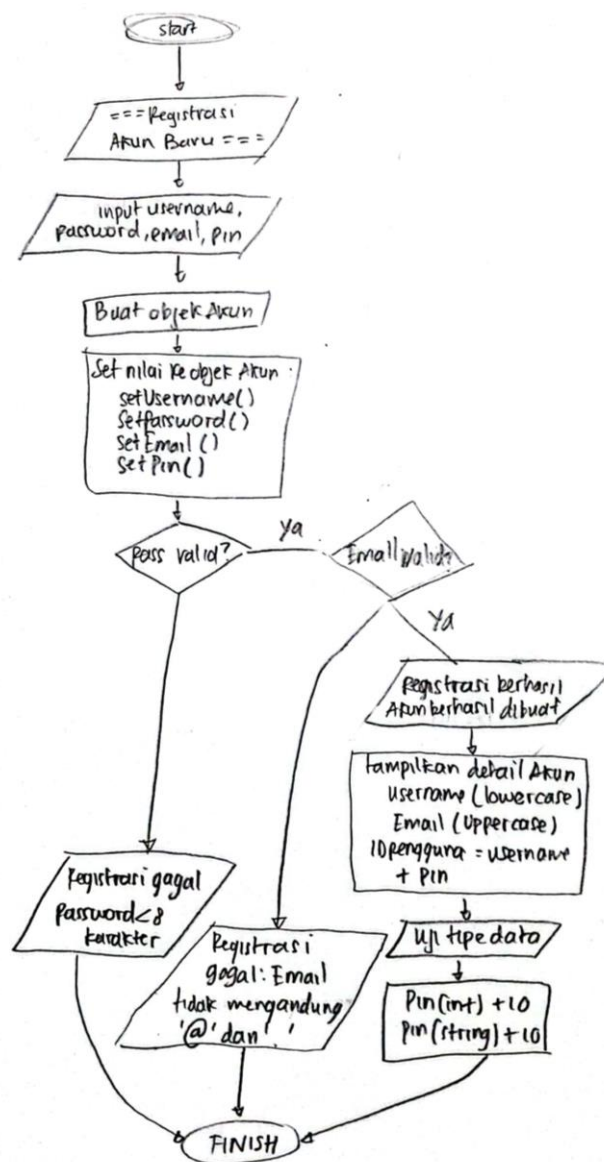
Pseudocode berikut disusun untuk menggambarkan alur logika program registrasi akun yang dibuat. Pseudocode ini menjelaskan langkah-langkah utama yang dilakukan program mulai dari penerimaan input pengguna, penyimpanan data ke dalam objek, proses validasi, hingga penampilan output akhir.

Judul Program Registrasi Akun Baru
Deklarasi Username, password, email : string Pin : integer Akun : object dari class akun validPass, validEmail : boolean
Algoritma <ol style="list-style-type: none">1. Mulai2. Tampilkan “===== REGISTRASI AKUN BARU =====”3. Input username, password, email, pin (bilangan 6 digit) dari pengguna4. Buat objek akun = new akun()5. Simpan data kedalam objek akun:<ul style="list-style-type: none">• A.setUsername(username)• A.setPassword(password)• A.setEmail(email)

- A.setpin(pin)
- 6. Lakukan validasi:
 - validPass = a.setPasswordValid()
 - validEmail = a.setEmailValid()
- 7. Jika validPass = false ATAU validEmail = false;
 - Tampilkan “--- REGISTRASI GAGAL ---“
 - Jika validPass = false;
Tampilkan pesan error password
 - Jika validEmail = false;
Tampilkan pesan error email
 - STOP program
- 8. Jika seluruh validasi berhasil;
 - Tampilkan “--- REGISTRASI BERHASIL ---“
 - Tampilkan “Akun untuk <username> telah berhasil dibuat”
- 9. Tampilkan detail akun:
 - Username lowercase
 - Email uppercase
 - Gabungan username(lowercase) + pin
- 10. Tampilkan pengujian tipe data:
 - PIN + 10 sebagai operasi integer
 - PIN + “10” sebagai operasi string
- 11. End

2.3 Flowchart

Flowchart berikut disusun untuk menggambarkan alur kerja program registrasi akun secara visual. Diagram ini menunjukkan urutan langkah mulai dari proses pengambilan input pengguna, pembentukan objek, validasi data, hingga penampilan output akhir. Dengan adanya flowchart, alur logika program menjadi lebih mudah dipahami dan dapat membantu dalam menganalisis proses yang terjadi di dalam



2.4 Kode Program

Akun.java

```

1. public class Akun {
2.
3.     private String username, password, email;
4.     private int pin;
5.
6.     // Setter
7.     public void setUsername (String username) {
8.         this.username = username;
9.     }
10.    public void setPassword (String password) {
11.        this.password = password;
12.    }
13.    public void setEmail (String email) {

```



```

14.         this.email = email;
15.     }
16.     public void setPin (int pin) {
17.         this.pin = pin;
18.     }
19.
20.     // Getter
21.     public String getUsername () {
22.         return username;
23.     }
24.     public String getPassword () {
25.         return password;
26.     }
27.     public String getEmail () {
28.         return email;
29.     }
30.     public int getPin () {
31.         return pin;
32.     }
33.
34.     // Validasi Password
35.     public boolean isPasswordValid () {
36.         return password != null && password.length()
>= 8;
37.     }
38.
39.     // Validasi Email
40.     public boolean isEmailValid () {
41.         if (email == null) return false;
42.         return email.contains("@") &&
email.contains(".");
43.     }
44.
45.
46. }

```

Registrasi.java

```

1. import java.util.Scanner;
2.
3. public class tugasAlproPekan7_2511532024 {
4.
5.     public static void main(String[] args) {
6.         Scanner input = new Scanner(System.in);
7.
8.         System.out.println("==== REGISTRASI AKUN BARU
====");
9.
10.        System.out.print("Masukkan Username: ");
11.        String username = input.nextLine();
12.        System.out.print("Masukkan Password: ");
13.        String password = input.nextLine();
14.        System.out.print("Masukkan Email: ");
15.        String email = input.nextLine();
16.        System.out.print("Masukkan PIN (6 digit): ");
17.        int pin = input.nextInt();

```

```

18.
19.         Akun a= new Akun();
20.         a.setUsername(username);
21.         a.setPassword(password);
22.         a.setEmail(email);
23.         a.setPin(pin);
24.
25.         boolean validPass = a.isPasswordValid();
26.         boolean validEmail = a.isEmailValid();
27.
28.         if (!validPass && !validEmail) {
29.             System.out.println("\n --- REGISTRASI
GAGAL ---");
30.
31.             if (!validPass) {
32.                 System.out.println("Password
Anda \" + a.getPassword() + "\" tidak valid (minimal 8
karakter)");
33.                 System.out.println("Silahkan
coba lagi");
34.             }
35.             if (!validEmail) {
36.                 System.out.println("Email Anda
\" + a.getEmail() + "\" tidak valid (harus mengandung '@' dan
'.')");
37.                 System.out.println("Silahkan
coba lagi");
38.             }
39.             return;
40.         }
41.
42.         System.out.println("\n --- REGISTRASI
BERHASIL ---");
43.         System.out.println("Akun untuk \" +
a.getUsername() + "\" telah berhasil dibuat.");
44.
45.         System.out.println("\n --- Detail Akun ---");
46.         System.out.println("Username (Lowercase): " +
a.getUsername().toLowerCase());
47.         System.out.println("Email (Uppercase): " +
a.getEmail().toUpperCase());
48.         System.out.println("ID Pengguna (Gabungan): "
+ a.getUsername().toLowerCase() + a.getPin());
49.
50.         System.out.println("\n --- Uji Tipe Data (PIN
Anda: " + a.getPin() + ") ---");
51.         System.out.println("PIN (int) + 10      = " +
(a.getPin() + 10));
52.         System.out.println("PIN (String) + \"10\" = "
+ (a.getPin() + "10"));
53.
54.     }
55.
56. }

```

2.5 Output Program dan Penjelasan Program dan Logika Validasi

```

===== 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@WAPRES.PAMAN
ID Pengguna (Gabungan): fufufafa67676767

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

```

```

===== REGISTRASI AKUN BARU =====
Masukkan Username: fufufafa67
Masukkan Password: 676767
Masukkan Email: fufufafa67waprespaman
Masukkan PIN (6 digit): 676767

--- REGISTRASI GAGAL ---
Password Anda "676767" tidak valid (minimal 8 karakter)
Silahkan coba lagi
Email Anda "fufufafa67waprespaman" tidak valid (harus mengandung '@' dan '.')
Silahkan coba lagi

```

Program yang dibuat pada tugas ini bertujuan untuk mensimulasikan proses registrasi akun baru dengan menggunakan bahasa pemrograman java. Program memanfaatkan konsep object-oriented programming (OOP), khususnya penggunaan class, object, enkapsulasi, method setter/getter, serta validasi data dengan helper method. Terdapat dua class utama yang digunakan, yaitu class Akun sebagai class data dan class registrasi sebagai class registrasi.

Pada class registrasi, program diawali dengan menampilkan judul dan meminta input dari pengguna, berupa username, password, email, dan pin. Data ini dibaca menggunakan class scanner, sehingga program dapat menerima masukan secara langsung dari keyboard. Setelah semua data berhasil diterima, program membuat sebuah objek baru dari kelas akun, lalu menyimpan semua nilai input tersebut ke dalam objek melalui

method setter. Dengan metode ini, seluruh data pengguna tersimpan secara terstruktur sesuai prinsip enkapsulasi dalam OOP.

Setelah proses penyimpanan data, program melakukan validasi menggunakan dua helper method yang didefinisikan di dalam class akun, yaitu `isPasswordValid()` dan `isEmailValid()`. Kedua method ini bertugas memastikan bahwa password dan email yang dimasukkan pengguna telah memenuhi ketentuan yang ditetapkan.

Validasi password dilakukan dengan memeriksa panjang karakter password. Aturannya adalah password harus memiliki minimal 8 karakter. Metode yang digunakan adalah sebagai berikut:

```
public boolean isPasswordValid(){
    Return password != null && password.length()>=8:
}
```

Jika password kurang dari 8 karakter, maka password dinyatakan tidak valid dan program menampilkan pesan kesalahan yang memberi tahu pengguna bahwa password harus terdiri dari minimal 8 karakter.

Validasi email dilakukan dengan memastikan bahwa email mengandung karakter '@' dan '.', yang merupakan syarat dasar format email. Metode validasi email yang digunakan adalah:

```
public boolean isEmailValid(){
    if (email == null) return false;
    return email.contains("@") && email.contains(".");
}
```

Jika email tidak mengandung salah satu dari dua karakter tersebut, maka email dinyatakan tidak valid dan program menampilkan pesan kesalahan yang menjelaskan bahwa email harus mengandung '@' dan '.'.

Apabila salah satu atau kedua validasi gagal, program akan menampilkan pesan REGISTRASI GAGAL beserta penjelasan detail kesalahan agar pengguna memahami bagian mana yang harus diperbaiki. Sebaliknya, jika seluruh validasi berhasil, program menampilkan pesan REGISTRASI BERHASIL dan melanjutkan dengan menampilkan detail akun, seperti username dalam format lowercase, email dengan format

uppercase, serta ID pengguna yang merupakan gabungan antara username (lowercase) dan pin.

Program juga menampilkan demonstrasi perbedaan operasi pada tipe data integer dan string melalui bagian Uji Tipe Data. Pada bagian ini, nilai PIN diuji dalam dua operasi berbeda: penjumlahan integer (PIN + 10) dan concatenation string (PIN + "10"). Dengan demikian, program ini tidak hanya menekankan pemahaman tentang input, validasi, dan manipulasi data, tetapi juga memperlihatkan perbedaan perlakuan antar tipe data dalam bahasa Java.

Melalui keseluruhan proses ini, program berhasil menggambarkan penerapan konsep OOP, validasi input, manipulasi string, dan alur logis dalam pembangunan aplikasi registrasi sederhana.

BAB III

PENUTUP

3.1 Kesimpulan

Dari praktikum ini dapat disimpulkan bahwa program registrasi akun berhasil dijalankan dengan menerapkan konsep OOP, seperti penggunaan class, object, method, dan validasi data. Program mampu menerima input, melakukan pengecekan password dan email, serta menampilkan hasil output sesuai ketentuan. Selain itu, praktikum ini membantu memahami cara kerja manipulasi string dan penggunaan percabangan dalam Java.

3.2 Saran

Dalam pengembangan selanjutnya, program ini dapat ditingkatkan dengan menambahkan fitur-fitur tambahan seperti verifikasi ulang password, format email yang lebih kompleks, pembatasan panjang PIN, serta penyimpanan data akun ke dalam file agar dapat digunakan kembali. Selain itu, antarmuka program dapat dibuat lebih interaktif untuk meningkatkan kenyamanan pengguna. Mahasiswa juga disarankan untuk terus memperdalam konsep OOP dan validasi input agar mampu mengembangkan aplikasi yang lebih kompleks dan aman di masa mendatang.

DAFTAR PUSTAKA

- [1] Oracle, “The Java™ Tutorials: Object-Oriented Programming Concepts,” Oracle, 2024. [Online]. Available: <https://docs.oracle.com/javase/tutorial/java/concepts/>
- [2] Oracle, “Classes and Objects (The Java™ Tutorials),” Oracle, 2024. [Online]. Available: <https://docs.oracle.com/javase/tutorial/java/javaOO/classes.html>
- [3] Oracle, “Strings (The Java™ Tutorials),” Oracle, 2024. [Online]. Available: <https://docs.oracle.com/javase/tutorial/java/data/strings.html>
- [4] Oracle, “Using the Scanner Class,” Oracle, 2024. [Online]. Available: <https://docs.oracle.com/javase/tutorial/essential/io/scanning.html>
- [5] Oracle, “Nested if Statements and Conditionals,” Oracle, 2024. [Online]. Available: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/ifs.html>
- [6] Oracle, “Java Platform, Standard Edition Documentation,” Oracle, 2024. [Online]. Available: <https://docs.oracle.com/javase/>