

LAPORAN TUGAS PRAKTIKUM
PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN
“STRING, CLASS & OBJECT PADA JAVA”



disusun Oleh:
FILZI JELILA INDA ROBBANI
2511533019
Dosen Pengampu:
Dr. WAHYUDI, S.T, M.T.
Asisten Praktikum:
JOVANTRI IMMANUEL GULO

DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
2025

KATA PENGANTAR

Dengan memanjatkan puji syukur kehadirat Allah SWT, atas limpahan rahmat dan karunia-Nya sehingga praktikum Algoritma dan Pemrograman kali ini dapat saya selesaikan dengan baik dan lancar. Shalawat serta salam semoga senantiasa tercurahkan kepada junjungan kita Nabi Muhammad SAW.

Laporan ini saya susun untuk memenuhi salah satu tugas pada mata kuliah Praktikum Algoritma dan Pemrograman di Universitas Andalas. Laporan ini diharapkan dapat menambah pemahaman terhadap konsep dasar pemrograman berorientasi objek serta penerapannya dalam studi kasus sederhana, yaitu pembuatan sistem registrasi akun dengan mengelola data menggunakan objek, memvalidasi input menggunakan *method*, dan memanfaatkan berbagai fungsi *string* untuk menghasilkan program yang lebih dinamis.

Saya menyampaikan terima kasih kepada dosen pengampu dan asisten praktikum yang telah membimbing serta memberikan arahan selama proses pembelajaran di kelas maupun di laboratorium komputer. Selain itu, saya juga berterima kasih kepada teman-teman praktikan atas bantuan dan dukungan yang diberikan, sehingga laporan ini dapat terselesaikan dengan baik.

Saya menyadari laporan ini masih jauh dari sempurna. Oleh karena itu, saya mohon maaf apabila terdapat kekurangan maupun kesalahan dalam penyusunan laporan ini. Harapan saya, semoga laporan ini dapat memberikan manfaat serta menambah pengetahuan bagi pembaca sekalian.

DAFTAR ISI

KATA PENGANTAR	ii
DAFTAR ISI.....	iii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Tujuan.....	1
1.3 Manfaat.....	2
1.4 Instruksi Tugas	2
BAB II PEMBAHASAN	4
2.1 Kode Program	4
<u>2.1.1</u> Class Data : Akun.java	4
<u>2.1.2</u> Class Utama : Registrasi.java (tugasAlproPekan7_NIM.java)	5
2.2 Output:.....	6
3.3 Flowchart.....	10
3.4 Penjelasan Singkat.....	10
BAB III PENUTUP	12
3.1 Kesimpulan.....	12
DAFTAR PUSTAKA	13

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pada praktikum sebelumnya, kita telah mempelajari dasar-dasar pemrograman Java seperti pemakaian String, pembuatan classs data/class utama, dan lain sebagainya. Pemahaman dasar tersebut memberikan fondasi awal untuk mengembangkan program yang lebih terstruktur dan fungsional.

Sebagai bentuk penerapan mandiri dari materi yang sudah dipelajari, pada tugas praktikum kali ini mahasiswa diminta mengembangkan sendiri sebuah program yang memanfaatkan konsep-konsep penting dalam pemrograman berorientasi objek. Program yang dibuat harus memanfaatkan *class* dan *object*, menerapkan *method* seperti *setter*, *getter*, dan *helper method* untuk validasi, serta menggunakan logika percabangan untuk menentukan alur program. Selain itu, mahasiswa juga diharuskan menerapkan beberapa operasi dan manipulasi String, seperti *.length()*, *.toUpperCase()*, *.toLowerCase()*, serta penggabungan *string*.

Melalui tugas ini, mahasiswa diharapkan mampu mengeksplorasi dan mengembangkan kemampuan pemrograman secara mandiri, mulai dari merancang struktur class, mengelola data menggunakan method, hingga mengimplementasikan logika dan operasi string untuk menyelesaikan masalah sederhana. Dengan demikian, tugas ini membantu mahasiswa membangun pemahaman yang lebih kuat dan terarah terhadap konsep dasar pemrograman berorientasi objek.

1.2 Tujuan

Tujuan dilaksanakannya praktikum ini adalah sebagai berikut:

- a. Mempelajari cara kerja dan penggunaan method seperti *getter*, *setter*, dan *helper*.

- b. Memahami cara membuat class serta object dalam pemrograman Bahasa Java.
- c. Memahami pengimplementasian berbagai operasi dan manipulasi *String* seperti *.length()*, *.toUpperCase()*, *.toLowerCase()*, serta penggabungan *string*.
- d. Mengembangkan kemampuan logika pemrograman agar dapat membuat kode secara lebih efisien dan terstruktur

1.3 Manfaat

- a. Dapat memahami struktur dasar pembuatan object dan class sehingga mampu membuat program yang lebih terstruktur.
- b. Memperoleh pengalaman dalam mengelola data menggunakan method
- c. Melatih keterampilan dalam merangkai kode yang efektif dan efisien.

1.4 Instruksi Tugas

1. 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. Struktur Program:
 - 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 (*public boolean isPasswordValid()*): Mengembalikan true jika panjang password 8 karakter atau lebih. dan *public boolean isEmailValid()*: Mengembalikan true jika email mengandung karakter @ dan “.”).
 - Class Utama: Registrasi.java: Class ini memiliki main method. Bertugas mengambil input, membuat object Akun, dan memanggil setter. Meminta input berikut dari pengguna: Username (String),

Password (String), Email (String), dan PIN Angka (int, misal 6 digit). Lalu Melakukan pengecekan if-else dengan memanggil method *isPasswordValid()* dan *isEmailValid()* dari object Akun.

3. 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).
- Simpan file Java dengan nama: tugasAlproPekan7_NIM.java (untuk **class utama**) dan Akun.java (untuk **class data**).
- Buat laporan lengkap dalam format Word (.docx), kemudian konversi ke PDF dan unggah ke repository GitHub.

BAB II

PEMBAHASAN

2.1 Kode Program

2.1.1 Class Data : Akun.java (saya membuat class dengan tambahan NIM dibelakangnya)

```
package pekan7_2511533019;

public class Akun_2511533019 {

    public static void main(String[] args) {}
        // TODO Auto-generated method stub

        private String username,password,email;
        private int pinAngka;
        //membuat mutator (setter)
        public void setUsername (String username) {
            this.username=username;
        }
        public void setPassword (String password) {
            this.password=password;
        }
        public void setEmail (String email) {
            this.email=email;
        }
        public void setPinAngka (int pinAngka) {
            this.pinAngka=pinAngka;
        }
        //membuat accesor (getter)
        public String getUsername() {
            return username;
        }
        public String getPassword() {
            return password;
        }
        public String getEmail() {
            return email;
        }
        public int getPinAngka() {
            return pinAngka;
        }
        //membuat helper method
        public boolean isPasswordValid() {
            return password.length()>=8;
        }
        public boolean isEmailValid() {
            return email.contains "@" && email.contains ".";
        }
}
```

```
}
```

2.1.2 Class Utama : Registrasi.java (saya membuat nama class sesuai ketentuan yaitu tugasAlproPekan7_NIM.java)

```
package pekan7_2511533019;

import java.util.Scanner;

public class tugasAlproPekan7_2511533019 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner input= new Scanner(System.in);
        System.out.println("== REGISTRASI AKUN BARU ==");
        System.out.print("Masukkan Username : ");
        String x = input.nextLine();
        System.out.print("Masukkan Password : ");
        String y= input.nextLine();
        System.out.print("Masukkan Email : ");
        String z= input.nextLine();
        System.out.print("Masukkan PIN (4 digit): ");
        int w= input.nextInt();
        Akun_2511533019 a= new Akun_2511533019();
        a.setUsername(x);
        a.setPassword(y);
        a.setEmail(z);
        a.setPinAngka(w);

        if (!a.isPasswordValid()) {
            System.out.println("Password harus min 8 karakter!!!");
            System.out.println("Silahkan coba lagi.");
        }else if (!a.isEmailValid()) {
            System.out.println("Email tidak valid! wajib mengandung
'@' dan '.'!!!");
            System.out.println("Silahkan coba lagi.");
        }else {
            System.out.println();
            System.out.println("--- REGISTRASI BERHASIL ---");
            System.out.println("Akun untuk \"username\" telah
berhasil dibuat");
            System.out.println();
            System.out.println("--- Detail Akun ---");
            System.out.println("Username (Lowercase) :
"+x.toLowerCase());
            System.out.println("Email (Uppercase)      :
"+z.toUpperCase());
            System.out.println("ID Pengguna (Gabungan):"+x+w);
            System.out.println();
            System.out.println("--- Uji Tipe Data (PIN Anda: "+w+
") ---");
        }
    }
}
```

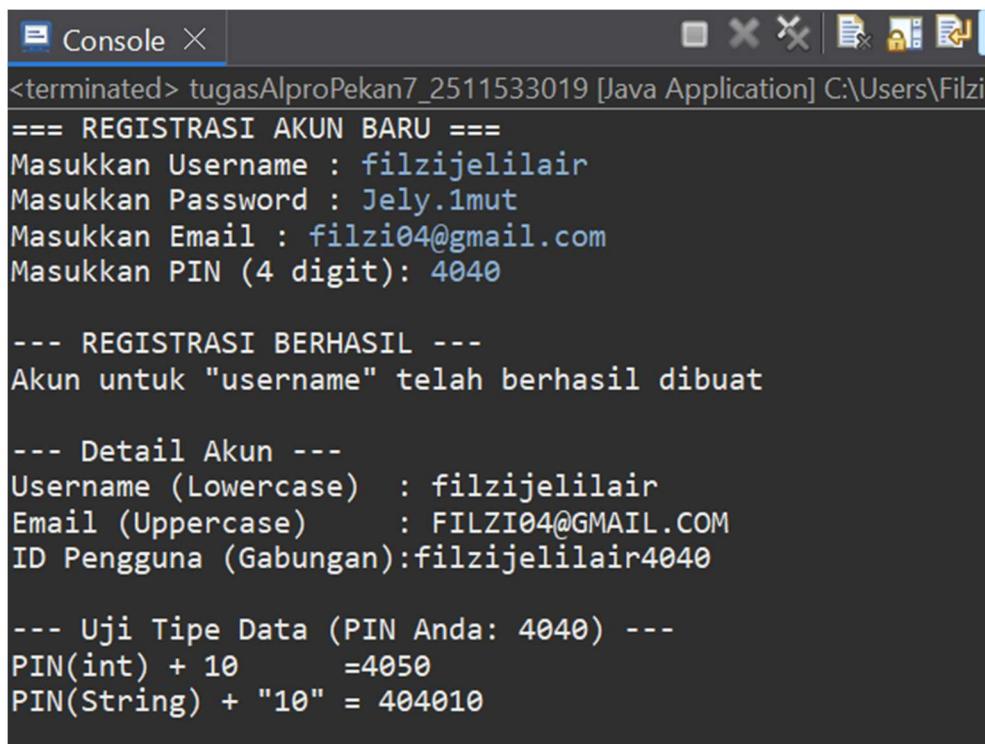
```

        System.out.println("PIN(int) + 10      =" +(w+10));
        System.out.println("PIN(String) + \"10\" = "+w+"10");
    }
    input.close();
}
}

```

2.2 Output:

1. Jika berhasil validasi :



The screenshot shows a Java application window titled "Console". The output text is as follows:

```

<terminated> tugasAlproPekan7_2511533019 [Java Application] C:\Users\Filzi
== REGISTRASI AKUN BARU ==
Masukkan Username : filzijelilair
Masukkan Password : Jely.1mut
Masukkan Email : filzi04@gmail.com
Masukkan PIN (4 digit): 4040

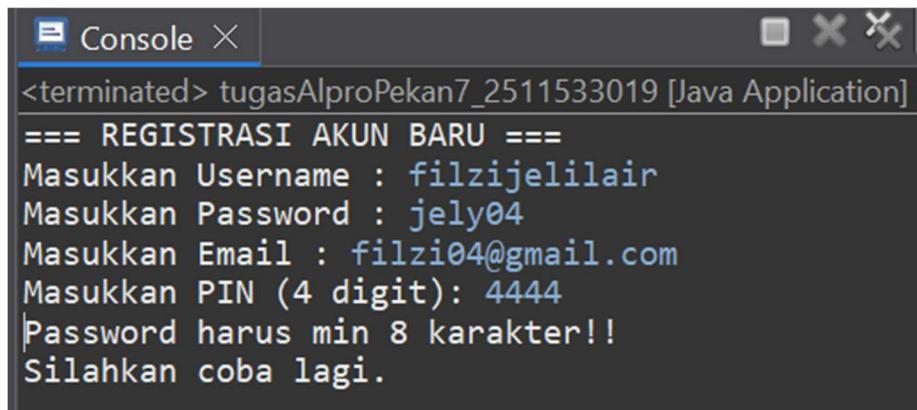
--- REGISTRASI BERHASIL ---
Akun untuk "username" telah berhasil dibuat

--- Detail Akun ---
Username (Lowercase) : filzijelilair
Email (Uppercase) : FILZI04@GMAIL.COM
ID Pengguna (Gabungan):filzijelilair4040

--- Uji Tipe Data (PIN Anda: 4040) ---
PIN(int) + 10      =4050
PIN(String) + "10" = 404010

```

2. Jika gagal validasi :



The screenshot shows a Java application window titled "Console". The output text is as follows:

```

<terminated> tugasAlproPekan7_2511533019 [Java Application]
== REGISTRASI AKUN BARU ==
Masukkan Username : filzijelilair
Masukkan Password : jely04
Masukkan Email : filzi04@gmail.com
Masukkan PIN (4 digit): 4444
Password harus min 8 karakter!!
Silahkan coba lagi.

```

The screenshot shows a Java application window titled "Console". The application is titled "REGISTRASI AKUN BARU". It prompts the user to enter a username ("Masukkan Username :"), password ("Masukkan Password :"), email ("Masukkan Email :"), and a PIN (4 digit) ("Masukkan PIN (4 digit):"). The user has entered "filzijelilair" for the username, "jely1mut" for the password, "filzi04@gmail.com" for the email, and "4440" for the PIN. An error message is displayed: "Email tidak valid! wajib mengandung '@' dan '.'!!!" (Email is invalid! It must contain '@' and '.'!!). Below the error message, it says "Silahkan coba lagi." (Please try again).

2.2 Pseudocode

Judul Program: Program Registrasi Akun

{program membuat/registrasi dan memvalidasi data akun baru}

Kamus

```
Var input : Scanner;  
Var username, password, email : String;  
Var pinAngka : integer;
```

Pseudocode

1. setUsername
2. this.username ← username(input)
3. setPassword
4. this.password ← password(input)
5. setEmail
6. this.email ← email(input)
7. setPinAngka
8. this.pinAngka ← pinAngka(input)
9. getUsername() String
10. return username
11. getPassword() String
12. return password
13. getEmail() String
14. return email

```
15. getPinAngka() integer  
16.    return pinAngka  
17. isPasswordValid()  
18.    if password.length ≥ 8  
19.        Return ← true  
20.    Else  
21.        Return ← false  
22. isEmailValid()  
23.    if email contains ('@' dan '.')  
24.        Return ← true  
25.    Else  
26.        Return ← false  
27. End  
28. Lanjut ke class utama
```

Kamus

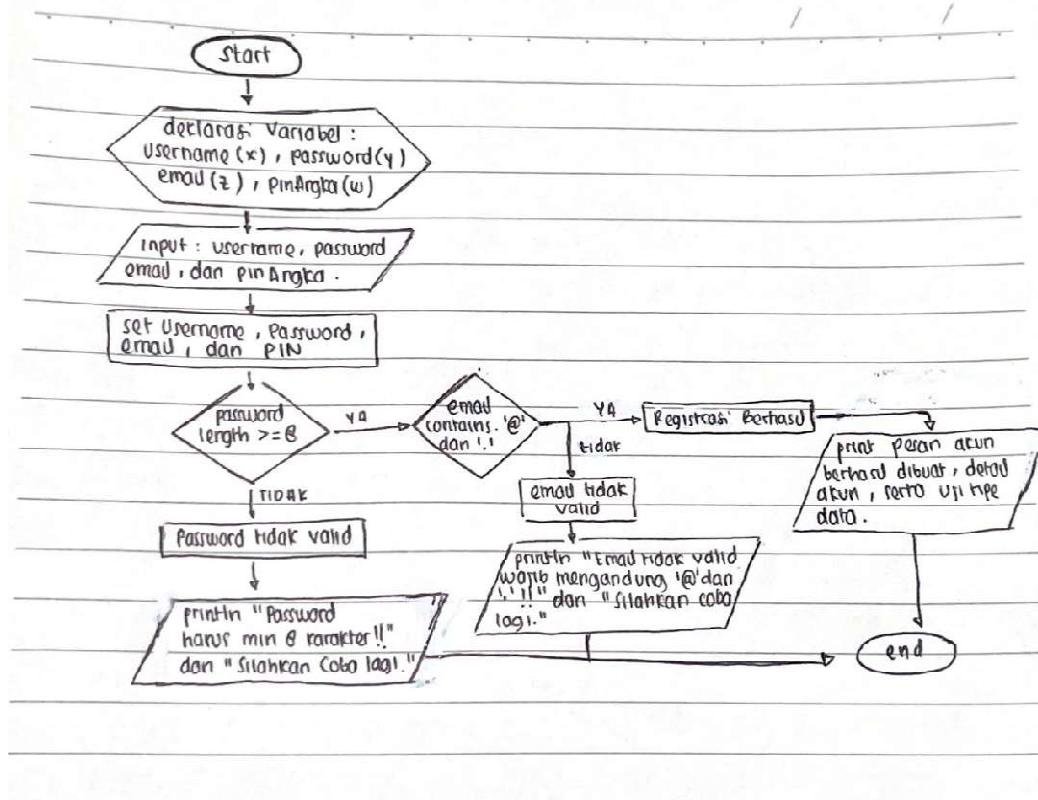
Var input : Scanner;
Var x, y, z : String;
Var w : integer;
Var a : objek akun

Pseudocode

1. Cetak ("== REGISTRASI AKUN BARU ==")
2. Input x ← username
3. Input y ← password
4. Input z ← email
5. Input w ← pinAngka
6. Buat object
7. a ← Akun_2511533019
8. a.setUsernam(x)
9. a.setPassword(y)

10. a.setEmail(z)
11. a.setPinAngka(w)
12. If a.isPasswordValid() = false, then
 13. Cetak (“Password harus min 8 karakter!!”)
 14. Cetak (“Silahkan coba lagi.”)
15. Else if a.isEmailValid() = false, then
 16. Cetak (“Password harus min 8 karakter!!”)
 17. Cetak (“Silahkan coba lagi.”)
18. Stop process
19. Else
 20. Cetak (“--- REGISTRASI BERHASIL ---”)
 21. Cetak (“Akun untuk \”username\” telah berhasil dibuat”)
 22. Cetak (“--- Detail Akun ---”)
 23. Cetak Username Lowercase
 24. Cetak Email Uppercase
 25. Cetak (“ID Pengguna (Gabungan):”username + pinAngka)
 26. Cetak (“--- Uji Tipe Data (PIN Anda: ”+pinAngka+) ---”)
 27. Cetak pinAngka+10 (integer)
 28. Cetak pinAngka+10 (String)
29. Input.close()
30. End

3.3 Flowchart



3.4 Penjelasan Singkat

Program pada tugas praktikum kali ini dibuat dengan menggunakan 2 class, yaitu **class data** (Akun_2511533019) dan **class utama** (tugasAlproPekan7_2511533019).

Pada class data, semua informasi akun seperti username, password, email, dan PIN disimpan sebagai *private* agar lebih aman dan hanya bisa diakses melalui *setter* dan *getter*. Class ini juga memiliki dua *helper* method, yaitu *isPasswordValid()* yang berfungsi untuk mengecek/memastikan apakah password yang diinput ≥ 8 karakter, dan *isEmailValid()* untuk mengecek/memastikan email apakah sudah mengandung "@" dan ".". Jadi, class data ini berfungsi sebagai tempat menyimpan data sekaligus melakukan pengecekan dasar sebelum datanya dipakai di program/class utama.

Di sisi lain, class utama adalah bagian yang berhubungan langsung dengan user. Dimana program meminta input berupa username, password, email, dan PIN, lalu memasukkannya ke objek akun yang sudah dibuat. Setelah itu, program menjalankan proses validasi dengan memanggil dua *helper* method dari class data (Akun_2511533019). Jika password atau email tidak sesuai aturan, program akan menampilkan pesan kesalahan. Namun jika semuanya valid, program menampilkan informasi akun yang sudah dibuat, seperti username dalam bentuk *lowercase*, email dalam bentuk *uppercase*, serta ID gabungan antara username dan PIN seperti yang terlihat pada gambar output diatas.

Program ini juga menunjukkan perbedaan antara penjumlahan angka dan penggabungan string melalui contoh PIN + 10 dan PIN + "10". Secara keseluruhan, program ini menerapkan materi *class & object, method, validasi, percabangan*, serta manipulasi *string* dalam bentuk kasus sederhana yaitu studi kasus pembuatan akun.

BAB III

PENUTUP

3.1 Kesimpulan

Kesimpulan yang dapat diambil dari praktikum yang telah dilakukan kali ini, ialah penggunaan *class* dan *object* sangat membantu dalam memisahkan antara data dan logika program. Class data (Akun_2511533019) berfungsi sebagai tempat penyimpanan data sekaligus melakukan validasi sederhana melalui *helper* method, sedangkan class utama bertugas menangani alur input-proses-output. Melalui program registrasi akun ini, konsep-konsep penting seperti *encapsulation*, *setter/getter*, *helper* method, serta percabangan dapat diterapkan secara nyata.

Selain itu, praktikum ini juga memperlihatkan bagaimana manipulasi *string* (seperti *lowercase*, *uppercase*, dan penggabungan *string*) digunakan dalam menghasilkan output yang lebih informatif.

Dengan menyelesaikan praktikum ini, diharapkan kedepannya semakin memahami cara membangun program berbasis objek yang lebih terstruktur, mudah dipelihara, dan sesuai kebutuhan.

DAFTAR PUSTAKA

- [1] Wahyudi, “*String dalam Java*” PowerPoint slides, Mata Kuliah Algoritma dan pemrograman, Universitas Andalas, Padang, 2025.
- [2] Oracle, “*The Java™ Tutorials Object-Oriented Programming Concepts*,” Oracle Java Documentation. [Online].
Tersedia: <https://docs.oracle.com/javase/tutorial/java/concepts/>
- [3] Oracle, “*The Java™ Tutorials Strings*,” Oracle Documentation.
Tersedia: [Strings \(The Java™ Tutorials > Learning the Java Language > Numbers and Strings\)](https://docs.oracle.com/javase/tutorial/java/nutsandbolts/strings.html)
- [3] H. M. Deitel and P. J. Deitel, Java: How to Program, 10th ed. Pearson, 2016.
Tersedia: <https://www.rmhjaved.com/oop/Java%20-%20How%20to%20Program%2010th%20Ed%20-%20Early%20Objects%20Version%20-%20Deitel.pdf>