

**TUGAS PRAKTIKUM JAVA PEKAN 7**  
**OBJECT-ORIENTED PROGRAMMING DAN MANIPULASI STRING**

**O**

**L**

**E**

**H**

**MUHAMMAD HANS NAFIS**

**2511532027**

Buatlah flowchart, pseudocode, program Java, serta tampilkan output programnya dari soal berikut !

#### 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.
2. Class Utama: Registrasi.java
  - 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 "."
  - 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.

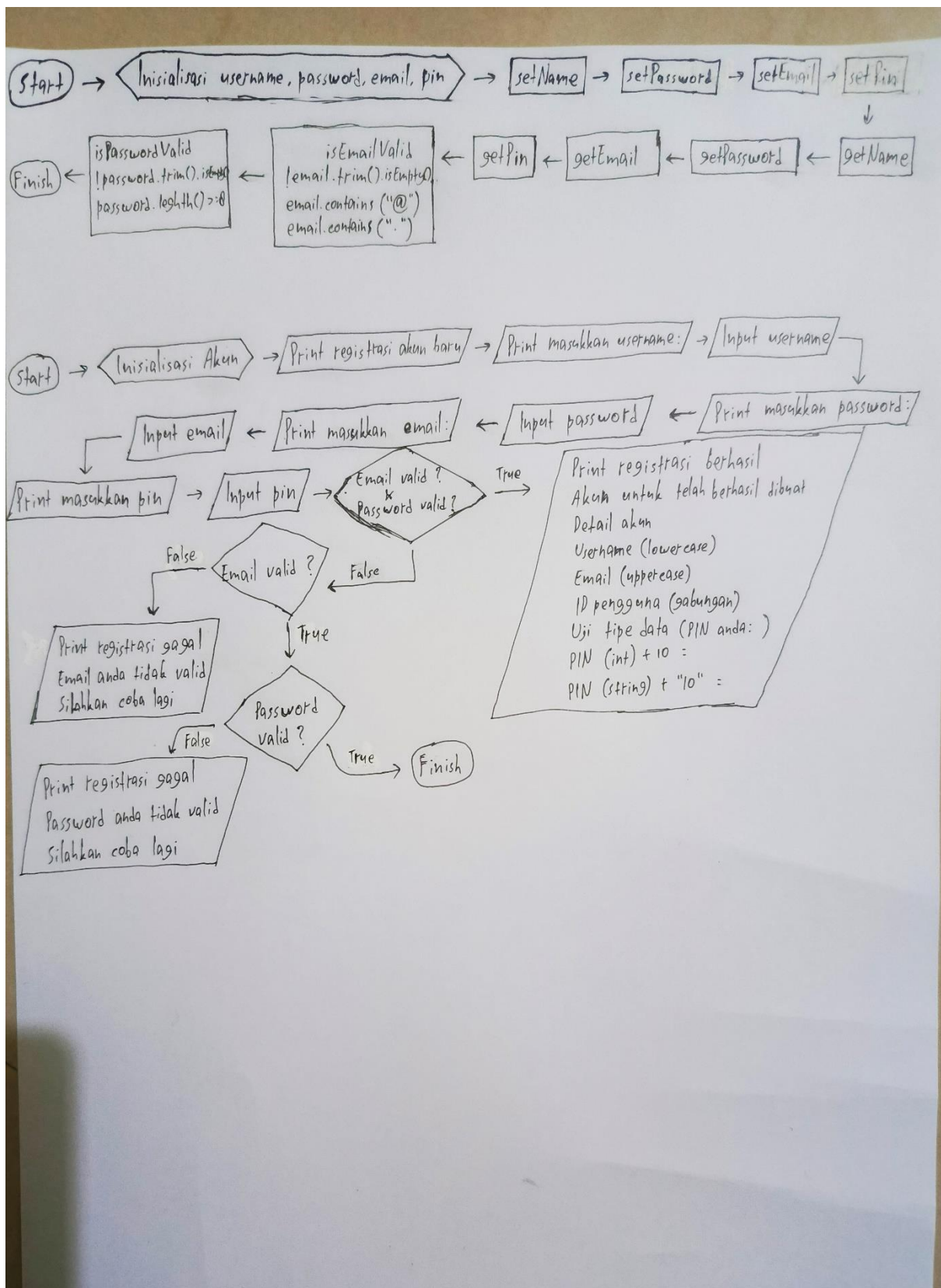
```
===== REGISTRASI AKUN BARU =====
Masukkan Username: fufufafa67
Masukkan Password: 6767676767
Masukkan Email: fufufafa67@wapres.paman
Masukkan PIN (6 digit): 676767

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

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

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

## Flowchart



## Pseudocode

### Judul

Sistem Registrasi Kredensial Pengguna (Class Data)

### Deklarasi

string username, password, email

integer pin

### Algoritma

1. method setName(username\_input : String)
2.     set username = username\_input
3. end method
4. method setPassword(password\_input : String)
5.     set password = password\_input
6. end method
7. method setEmail(email\_input : String)
8.     set email = email\_input
9. end method
10. method setPin(pin\_input : Integer)
11.     set pin = pin\_input
12. end method
13. method getName() returns String
14.     return username
15. end method
16. method getPassword() returns String
17.     return password
18. end method
19. method getEmail() returns String
20.     return email
21. end method
22. method getPin() returns Integer
23.     return pin
24. end method
25. method isValidEmail(check\_email : String) returns Boolean
26.     if check\_email is not null
27.         and check\_email.trim() is not empty
28.         and check\_email contains "@"
29.         and check\_email contains "."
30.     then
31.         return true
32.     else
33.         return false
34.     end if
35. end method
36. method isValidPassword(check\_password : String) returns Boolean
37.     if check\_password is not null
38.         and check\_password.trim() is not empty
39.         and length(check\_password) >= 8
40.     then
41.         return true
42.     else

43.       return false 44.       end if 45.       end method
--

<b>Judul</b>
--------------

Sistem Registrasi Kredensial Pengguna (Class Utama)
---

<b>Deklarasi</b>
------------------

Sistem Registrasi Kredensial Pengguna (Class Data)
--

<b>Algoritma</b>
------------------

1. print "registrasi akun baru" 2. print "masukkan username: " 3. print "masukkan password: " 4. print "masukkan pin: " 5. if (email valid & password valid) then 6. print "registrasi berhasil" 7. print "akun untuk berhasil dibuat" 8. print "detail akun" 9. print "username (lowecase):" 10. print "email (uppercase):" 11. print "ID pengguna (Gabungan):" 12. print "uji tipe data (PIN anda: )" 13. print "PIN (int) + 10 = " 14. print "PIN (string) + "10" = " 15. else if (email tidak valid) then 16. print "registrasi gagal" 17. print "email anda tidak valid" 18. print "silahkan coba lagi" 19. else if (password tidak valid) then 20. print "registrasi gagal" 21. print "password anda tidak valid" 22. print "silahkan coba lagi" 23. else 24. endif
--

## Kode Program Java

```
public class Akun {
    private String username, password, email;
    private int pin;

    public void setName (String username) {
        this.username = username;
    }

    public void setPassword (String password) {
        this.password = password;
    }

    public void setEmail (String email) {
        this.email = email ;
    }

    public void setPin (int pin ) {
        this.pin = pin;
    }

    public String getName() {
        return username;
    }

    public String getPassword() {
        return password;
    }

    public String getEmail() {
        return email;
    }

    public int getPin() {
        return pin;
    }

    public boolean isEmailValid(String email) {
        return email != null
            && !email.trim().isEmpty()
            && email.contains("@")
            && email.contains(".");
    }

    public boolean isPasswordValid(String password) {
        return password != null
            && !password.trim().isEmpty()
            && password.length() >= 8;
    }
}
```

Kelas Akun pada kode di atas berfungsi sebagai *blueprint* untuk merepresentasikan sebuah akun dengan empat atribut utama: username, password, email, dan pin. Semua atribut bersifat private, sehingga hanya dapat diakses melalui metode *setter* dan *getter*. Pendekatan ini merupakan contoh dari konsep enkapsulasi dalam OOP, yang bertujuan menjaga

keamanan data agar tidak diakses atau dimodifikasi secara langsung dari luar kelas. Metode seperti setName, setPassword, setEmail, dan setPin memungkinkan objek lain mengisi nilai atribut, sedangkan metode getName, getPassword, getEmail, dan getPin memungkinkan pengambilan nilai atribut tersebut.

Selain menyimpan data akun, kelas ini juga menyediakan fitur validasi melalui dua metode: isEmailValid dan isPasswordValid. Metode isEmailValid mengecek apakah email tidak kosong, tidak null, dan mengandung simbol "@" serta titik, yang merupakan syarat minimal format email. Sementara itu, isPasswordValid memastikan password tidak kosong dan memiliki panjang minimal delapan karakter, sebagai standar keamanan dasar. Kedua metode ini biasanya digunakan sebelum data disimpan, untuk memastikan informasi akun sudah memenuhi aturan valid yang diperlukan.

Secara keseluruhan, kelas Akun ini sudah memiliki struktur dasar yang rapi untuk membuat dan memvalidasi data akun. Namun, dalam penggunaan nyata, validasi dapat dibuat lebih ketat (misalnya format email dengan regex atau syarat password lebih kompleks). Meskipun begitu, kode ini sudah cukup baik sebagai contoh implementasi enkapsulasi, setter/getter, serta validasi sederhana pada Java.

```
import java.util.Scanner;

public class TugasAlproPekan7_2511532027 {
    public static void main (String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("==== REGISTRASI AKUN BARU =====");
        System.out.print("Masukkan Username: ");
        String w = input.nextLine();
        System.out.print("Masukkan Password: ");
        String x = input.nextLine();
        System.out.print("Masukkan Email: ");
        String y = input.nextLine();
        System.out.print("Masukkan PIN (8 digit): ");
        int z = input.nextInt();
        Akun a = new Akun();
        a.setName(w);
        a.setPassword(x);
        a.setEmail(y);
        a.setPin(z);
        if (a.isEmailValid(y) && a.isPasswordValid(x)) {
            System.out.println("");
            System.out.println("--- REGISTRASI BERHASIL ---");
            System.out.println("Akun untuk \"" + w + "\" telah berhasil dibuat.");
            System.out.println("");
            System.out.println("--- Detail Akun ---");
            System.out.println("Username (Lowercase): " + a.getName().toLowerCase());
            System.out.println("Email (Uppercase): " + a.getEmail().toUpperCase());
        }
    }
}
```

```

        System.out.println("ID Pengguna (Gabungan): " + a.getName() +
a.getPin());
        System.out.println("");
        System.out.println("---" + " Uji Tipe Data (PIN Anda: " +
a.getPin() + " )" + " ---");
        System.out.println("PIN (int) + 10 = " + z);
        System.out.println("PIN (String) + \"\" + 10 + \"\" = " +
a.getPin() + "10");
    }else if(!a.isEmailValid(y)) {
        System.out.println("");
        System.out.println("---- REGISTRASI GAGAL ----");
        System.out.println("Email Anda \"\" + a.getEmail() + \"\" tidak
valid (harus mengandung '@' dan '.').");
        System.out.println("Silakan coba lagi.");
    }else if (!a.isPasswordValid(x)) {
        System.out.println("");
        System.out.println("---- REGISTRASI GAGAL ----");
        System.out.println("Password Anda \"\" + a.getPassword() + \"\"
tidak valid (harus terdiri dari 8 karakter atau lebih).");
        System.out.println("Silakan coba lagi.");
    }

    input.close();
}
}

```

Program Java di atas adalah simulasi registrasi akun baru yang melibatkan input dari pengguna, validasi data, serta penggunaan sebuah kelas bernama Akun. Program dimulai dengan meminta pengguna memasukkan username, password, email, dan PIN. Semua data tersebut kemudian disimpan ke dalam sebuah objek Akun melalui method setter (setName, setPassword, setEmail, setPin). Setelah itu, program memeriksa apakah email dan password valid menggunakan method isEmailValid() dan isPasswordValid() dari kelas Akun. Validasi email kemungkinan mengecek keberadaan karakter '@' dan '.', sedangkan validasi password mengecek panjang minimal 8 karakter.

Apabila kedua validasi tersebut berhasil, program menampilkan pesan bahwa registrasi berhasil, kemudian menampilkan detail akun dengan beberapa manipulasi string: username ditampilkan dalam huruf kecil, email dalam huruf kapital, dan membuat ID pengguna dari gabungan username dan PIN. Program juga menunjukkan perbedaan tipe data dengan menampilkan operasi matematis terhadap PIN (sebagai int) dan operasi penggabungan teks terhadap PIN (sebagai String). Ini membantu menunjukkan perbedaan antara penjumlahan numerik dan concatenation pada Java.

Sebaliknya, jika validasi gagal, program menampilkan pesan bahwa registrasi tidak berhasil disertai alasan gagalnya: apakah email tidak valid, atau password kurang panjang. Dengan demikian, keseluruhan program berfungsi sebagai latihan penggunaan kelas dan objek, input-output, validasi data, manipulasi string, dan pengkondisian dalam bahasa Java.



## Output Program

<terminated> TugasAlproPekan7\_2511532027 [Java Application] C:

```
===== REGISTRASI AKUN BARU =====
Masukkan Username: a
Masukkan Password: agdhftjfhjdjhj
Masukkan Email: a@b.c
Masukkan PIN (8 digit): 12345678
|
--- REGISTRASI BERHASIL ---
Akun untuk "a" telah berhasil dibuat.

--- Detail Akun ---
Username (Lowercase): a
Email (Uppercase): A@B.C
ID Pengguna (Gabungan): a12345678

--- Uji Tipe Data (PIN Anda: 12345678 ) ---
PIN (int) + 10 = 12345678
PIN (String) + "10" = 1234567810
```

```
===== REGISTRASI AKUN BARU =====
Masukkan Username: a
Masukkan Password: gghhhrherer
Masukkan Email: abc
Masukkan PIN (8 digit): 12345678
|
--- REGISTRASI GAGAL ---
Email Anda "abc" tidak valid (harus mengandung '@' dan '.').
Silakan coba lagi.
```

```
===== REGISTRASI AKUN BARU =====
Masukkan Username: a
Masukkan Password: a
Masukkan Email: a@b.c
Masukkan PIN (8 digit): 12345678
|
--- REGISTRASI GAGAL ---
Password Anda "a" tidak valid (harus terdiri dari 8 karakter atau lebih).
Silakan coba lagi.
```