

LAPORAN PRAKTIKUM KE-3
PROYEK PENGANTAR PEMROGRAMAN



Disusun Oleh :

Nama : Ariqah Naurah
NPM : G1A025039
Kelas : A1

Nama Asisten Dosen :

- | | |
|---------------------------|-------------|
| 1. Davi Sulaiman | (G1A022001) |
| 2. Julia Mayang Sari | (G1A022010) |
| 3. Evelyn Unike Aritonang | (G1A022024) |
| 4. Reksi Hendra | (G1A022032) |
| 5. Ade Irawan | (G1A022083) |
| 6. Azzahra Faranisa | (G1A023010) |
| 7. Lio Kusnata | (G1A023013) |
| 8. Ahsan Nurfauzi | (G1A024002) |
| 9. Gusti Revalion Putrada | (G1A024081) |
| 10. Lantera Meusanah | (G1A024105) |

Dosen Pengampu :

1. Funny Farady Coastera, S.Kom, M.T
2. Arie Vatesia, S.T., M.TI, Ph.D

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS BENGKULU

2025

LANDASAN TEORI

Algoritma merupakan suatu urutan langkah logis yang disusun secara sistematis untuk menyelesaikan suatu permasalahan. Dalam dunia pemrograman, algoritma menjadi dasar penting karena tanpa adanya alur yang jelas, program tidak dapat berjalan sesuai harapan. Sebelum diterjemahkan ke dalam bahasa pemrograman seperti Java, algoritma biasanya direpresentasikan terlebih dahulu dalam bentuk pseudocode ataupun flowchart. Representasi ini bertujuan agar langkah penyelesaian masalah dapat dipahami baik oleh manusia maupun oleh komputer. Menurut Poliban Press (2021), algoritma memiliki beberapa sifat utama, yakni harus memiliki input, menghasilkan output, bersifat jelas dan tidak ambigu, memiliki batas atau berhenti, serta harus efektif dalam menyelesaikan permasalahan.

Java sebagai bahasa pemrograman tingkat tinggi memiliki struktur dasar yang sama dengan bahasa pemrograman lain, yaitu input, proses, dan output. Input adalah data yang dimasukkan baik dari pengguna, sensor, maupun variabel yang telah ditentukan sebelumnya. Proses adalah tahapan logika yang mengolah data sesuai aturan yang telah dibuat, sedangkan output adalah hasil yang ditampilkan kepada pengguna berdasarkan kondisi yang diberikan. Misalnya pada kasus pemeriksaan ruangan, input berupa panjang dan lebar ruangan, proses berupa logika perbandingan dengan aturan minimum, sedangkan output berupa teks “Ruang kelas” atau “Bukan ruang kelas”. Hal ini sejalan dengan Irmayanti (2020) yang menjelaskan bahwa struktur dasar program harus mampu mengolah masukan menjadi keluaran yang sesuai.

Dalam Java, terdapat konsep variabel yang digunakan sebagai wadah untuk menyimpan data. Nilai variabel dapat berubah sesuai kebutuhan selama program berjalan. Tipe data mendefinisikan jenis nilai yang dapat disimpan dalam variabel, misalnya int untuk bilangan bulat, double untuk bilangan pecahan, char untuk karakter, string untuk teks, dan boolean untuk nilai benar atau salah. Selain itu terdapat juga konstanta, yaitu variabel yang nilainya tetap sepanjang eksekusi program. Pada kasus lampu pintar, variabel boolean seperti adaOrang menjadi penentu apakah lampu menyala atau mati. Dalam kasus sistem parkir, variabel jam dengan tipe data integer digunakan untuk menentukan waktu masuk kendaraan dan tarif yang berlaku.

Struktur kontrol atau control structure dalam Java adalah salah satu bagian paling penting karena berfungsi untuk mengendalikan alur program. Struktur kontrol memungkinkan program untuk mengambil keputusan sesuai kondisi tertentu. Menurut Universitas Terbuka (2017), terdapat beberapa bentuk struktur kontrol, yaitu if, if...else, else if, nested if, serta operator ternary. Struktur if digunakan untuk mengeksekusi kode jika kondisi bernilai benar. Jika terdapat dua kemungkinan hasil, maka digunakan if...else. Untuk menangani beberapa kondisi sekaligus secara berurutan digunakan else if, seperti pada sistem tarif parkir mall yang

membedakan tarif pagi, siang, dan malam. Nested if atau if bertingkat digunakan untuk pengecekan lebih kompleks, contohnya dalam kasus akses e-book, di mana sistem harus mengecek apakah seseorang anggota terlebih dahulu sebelum mengecek status pembayaran. Ternary operator adalah bentuk singkat dari if...else yang sering digunakan untuk kondisi sederhana, contohnya pada sistem lift rumah sakit yang menampilkan “Lift bisa jalan” jika jumlah orang kurang dari atau sama dengan delapan, dan menampilkan “Terlalu penuh” jika lebih dari delapan.

Operator relasional dan logika juga berperan penting dalam menyusun aturan program. Operator relasional seperti lebih besar (>), lebih kecil (<), sama dengan (==), lebih besar atau sama dengan (>=), serta lebih kecil atau sama dengan (<=) digunakan untuk membandingkan nilai antar variabel. Operator logika seperti AND (&&), OR (||), dan NOT (!) digunakan untuk menggabungkan beberapa kondisi sekaligus. Misalnya, dalam kasus akses jalan tol, aturan saldo minimal 20.000 dapat diuji menggunakan operator relasional >=. Apabila saldo mencukupi, maka output program adalah boleh masuk, sebaliknya jika saldo kurang maka output yang ditampilkan adalah tidak boleh masuk.

Selain itu, Java dikenal sebagai bahasa pemrograman berorientasi objek atau Object-Oriented Programming (OOP). OOP terdiri dari konsep kelas, objek, dan metode. Kelas merupakan cetak biru yang berisi definisi atribut dan perilaku, objek merupakan instance dari kelas, sementara metode mendefinisikan aksi yang dapat dilakukan objek. Walaupun soal-soal pada praktikum ini masih berada pada tahap dasar yaitu penggunaan variabel, tipe data, operator, serta struktur kontrol, prinsip OOP tetap penting sebagai dasar untuk pemahaman lebih lanjut dalam pengembangan aplikasi berskala besar (Irmayanti, 2020).

Dalam hal input dan output, Java menyediakan perintah System.out.println() untuk menampilkan data atau informasi ke layar. Input bisa berasal dari pengguna melalui scanner, dari sensor, atau sekadar dari variabel yang diinisialisasi di dalam program. Pada kasus-kasus soal yang diberikan, input berupa data numerik seperti panjang, lebar, jam, jumlah orang, serta data boolean seperti adaOrang, anggota, dan bayar. Data-data tersebut kemudian diproses sesuai aturan yang sudah ditentukan dan menghasilkan output berupa teks yang memberi informasi atau instruksi, misalnya “Ruang kelas”, “Lampu menyala”, “Tarif Rp 7000”, atau “Akses e-book dibuka”. Dengan demikian, teori mengenai algoritma, struktur program, variabel, tipe data, konstanta, struktur kontrol, operator, serta input-output dalam Java menjadi landasan utama yang relevan untuk memahami dan menyelesaikan soal praktikum yang diberikan.

Dalam praktik pemrograman, selain memahami algoritma, tipe data, dan struktur kontrol, dan pengujian juga memegang peranan penting agar program bekerja dengan baik.

SOAL DAN PEMBAHASAN

1. Ade merupakan seorang arsitek dan programmer yang mana ia akan membuat program sederhana untuk mengecek apakah sebuah ruangan bisa disebut “ruang kelas”.

Aturan:

Panjang ruangan minimal 5 meter.

Lebar ruangan minimal 4 meter.

Jika kedua syarat dipenuhi, hasilnya “Ruang kelas”. Jika tidak, “Bukan ruang kelas”.

Pernyataan:

Jika sebuah ruangan memiliki panjang = 6 dan lebar = 3, maka program akan mencetak “Ruang kelas”.

Apakah pernyataan ini benar atau salah?

Jawab: **Pertanyaan ini salah karena, Berdasarkan ketentuan yang diberikan, sebuah ruangan dapat dikategorikan sebagai “Ruang kelas” apabila panjang dan lebar ruangan memenuhi syarat minimal. Di soal tersebut diketahui aturannya bahwa:**

1. Syarat panjang minimal 5 meter, dan ruangan tersebut memiliki panjang 6 meter

2. Syarat lebar minimal 4 meter, sedangkan di ruangan tersebut memiliki lebar 3 meter.

Karena kedua syarat harus terpenuhi, tetapi pada soal ini syarat lebar tidak terpenuhi, maka ruangan di soal tersebut tidak dapat dikategorikan sebagai “Ruang kelas”.

2. Alicia seorang programmer di bidang IoT ia diminta oleh bu arie untuk membuat sistem lampu pintar.

Aturannya:

- Jika sensor mendeteksi orang masuk, lampu menyala.
- Jika tidak ada orang, lampu mati.

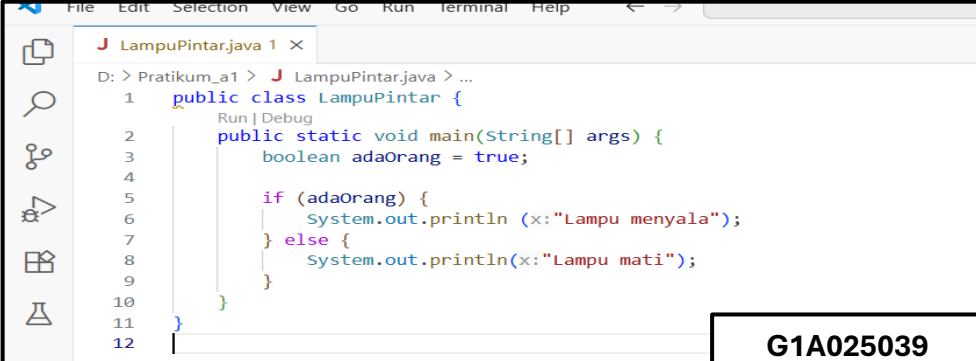
a. Buatlah kode program sesuai cerita di atas.

b. Jalankan program jika adaOrang = true.

c. Jalankan program jika adaOrang = false.

Jawab:

b. PrintScreen Source Code



```
1 public class LampuPintar {
2     public static void main(String[] args) {
3         boolean adaOrang = true;
4
5         if (adaOrang) {
6             System.out.println(x:"Lampu menyala");
7         } else {
8             System.out.println(x:"Lampu mati");
9         }
10    }
11 }
12
```

G1A025039

Gambar 2.1 Source Code Soal No.2 b

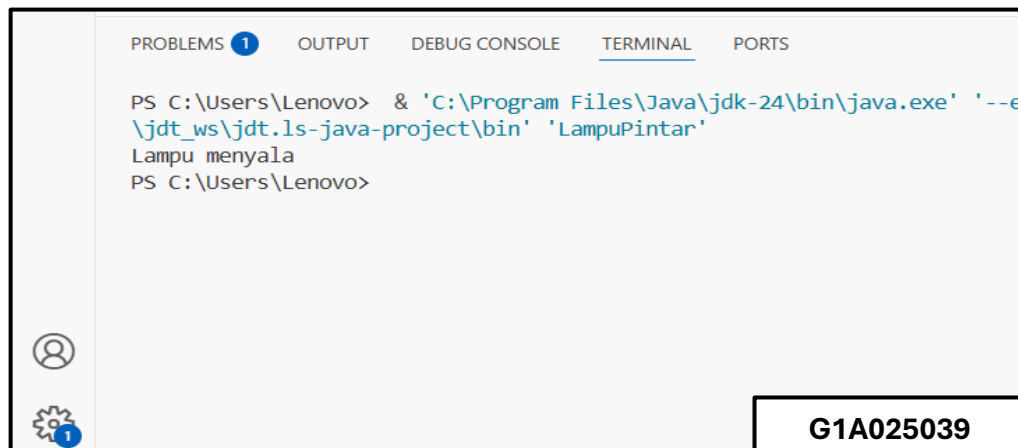
Source Code:

```
public class LampuPintar {  
    public static void main(String[] args) {  
        boolean adaOrang = true;  
  
        if (adaOrang) {  
            System.out.println("Lampu menyala");  
        } else {  
            System.out.println("Lampu mati");  
        }  
    }  
}
```

Penjelasan Source Code:

Pada baris kode `boolean adaOrang = true;`, dibuat sebuah variabel bernama `adaOrang` dengan tipe data `boolean`. Tipe data `boolean` hanya memiliki dua kemungkinan nilai, yaitu `true` (benar) atau `false` (salah). Pemberian nilai `true` pada variabel ini menandakan bahwa ada orang, sehingga nilai tersebut nantinya akan digunakan dalam percabangan `if` untuk menentukan apakah lampu akan menyala atau mati.

PrintScreen Output:

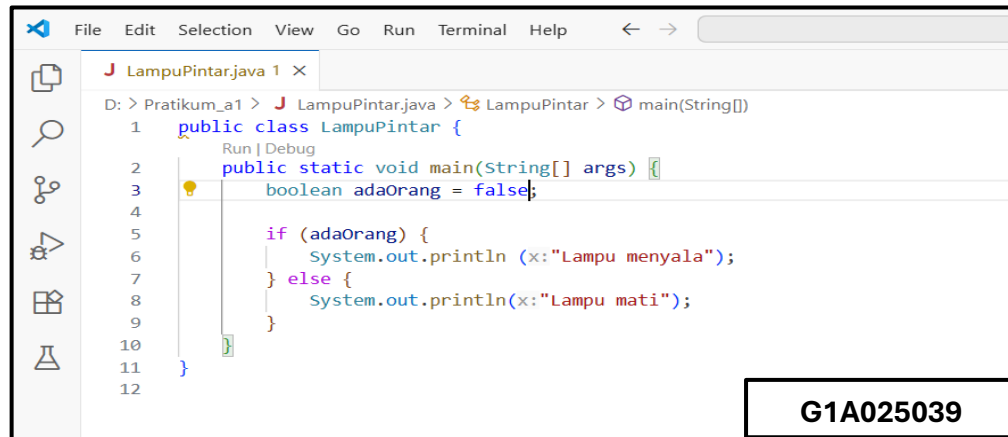


Gambar 2.2 Output Soal No.2 b

Penjelasan Output:

Ketika program `LampuPintar` dijalankan, hasil yang muncul di terminal adalah “Lampu menyala”. Hal ini terjadi karena pada program sebelumnya variabel `adaOrang` diberi nilai `true`, sehingga percabangan `if` akan dijalankan dan menampilkan pesan “Lampu menyala” pada layar. Output ini membuktikan bahwa logika program berjalan dengan benar, yaitu lampu akan menyala jika variabel `adaOrang` bernilai `true`.

c. .PrintScreen Source Code



Gambar 2.3 Source Code Soal No.2 c

Source Code:

```
public class LampuPintar {  
    public static void main(String[] args) {  
        boolean adaOrang = false;  
  
        if (adaOrang) {  
            System.out.println ("Lampu menyala");  
        } else {  
            System.out.println("Lampu mati");  
        }  
    }  
}
```

Penjelasan Source Code:

Pada program **LampuPintar** di atas, variabel **adaOrang** diisi dengan nilai **false**. Variabel ini bertipe **boolean** sehingga hanya dapat bernilai **true** atau **false**. Ketika program dijalankan, pernyataan **if (adaOrang)** akan diperiksa. Karena **adaOrang** bernilai **false**, kondisi pada **if** tidak terpenuhi sehingga baris kode di dalam blok **if** akan dilewati. Program kemudian masuk ke blok **else** dan menjalankan perintah **System.out.println("Lampu mati")**.

PrintScreen Output:



Gambar 2.4 Output Soal No.2 c

Penjelasan Output:

Ketika program dijalankan dengan nilai `adaOrang = false`, hasil yang muncul pada terminal adalah **"Lampu mati"**. Output ini sesuai dengan logika program, karena kondisi `if (adaOrang)` tidak terpenuhi akibat variabel bernilai `false`, sehingga program beralih ke blok `else` dan menampilkan pesan **"Lampu mati"**.

3. Bencoolen mall memiliki aturan tarif parkir yang berbeda sesuai waktu kendaraan masuk. Sistem parkir

otomatis harus mengikuti ketentuan berikut:

- Jika kendaraan masuk sebelum jam 12 siang, maka sistem akan menampilkan "Pagi, tarif Rp 5000".
- Jika kendaraan masuk antara jam 12 siang hingga sebelum jam 18 sore, maka sistem akan menampilkan "Siang, tarif Rp 7000".
- Jika kendaraan masuk jam 18 sore atau lebih, maka sistem akan menampilkan "Malam, tarif Rp 10000".

Cahaya merupakan programmer yang dipilih oleh pihak Bencoolen Mall, ia diminta untuk membuat

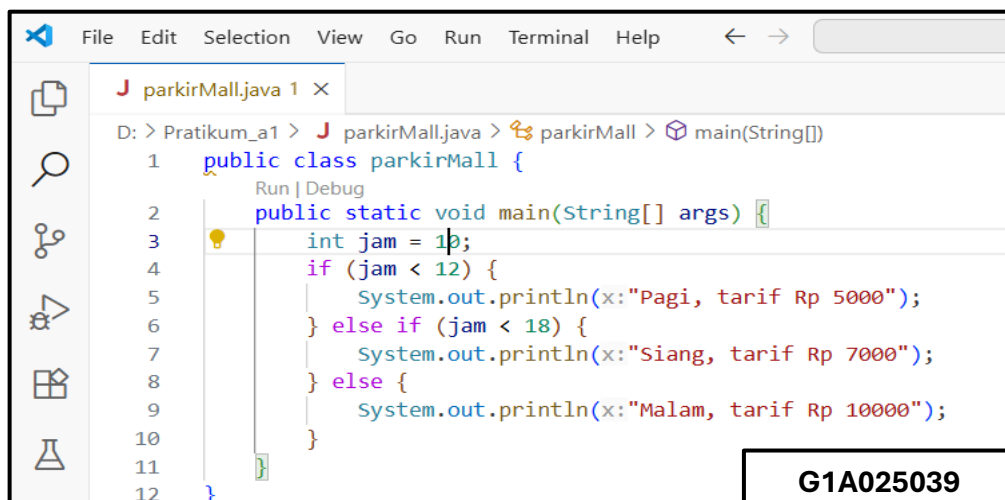
program berdasarkan aturan tersebut.

Pertanyaan:

- Buatlah kode program menggunakan `else if` untuk menerapkan aturan parkir mall tersebut.
- Jalankan program dengan input jam = 10 (kendaraan masuk pagi hari).
- Jalankan program dengan input jam = 20 (kendaraan masuk malam hari).

Jawab:

b.PrintScreen Source Code



```
File Edit Selection View Go Run Terminal Help
J parkirMall.java 1 X
D: > Pratikum_a1 > J parkirMall.java > parkirMall > main(String[])
1 public class parkirMall {
2     public static void main(String[] args) {
3         int jam = 10;
4         if (jam < 12) {
5             System.out.println(x:"Pagi, tarif Rp 5000");
6         } else if (jam < 18) {
7             System.out.println(x:"Siang, tarif Rp 7000");
8         } else {
9             System.out.println(x:"Malam, tarif Rp 10000");
10        }
11    }
12 }
```

G1A025039

Gambar 3.1 Source Code Soal No.3 b

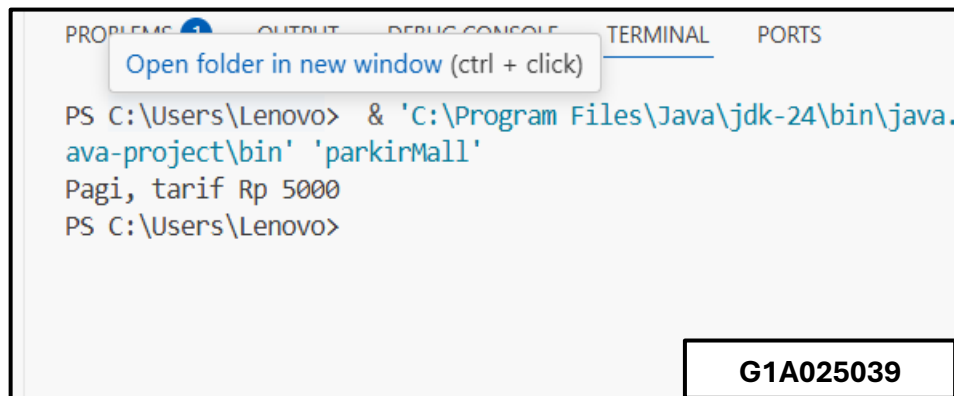
Source Code:

```
public class parkirMall {  
    public static void main(String[] args) {  
        int jam = 10;  
        if (jam < 12) {  
            System.out.println("Pagi, tarif Rp 5000");  
        } else if (jam < 18) {  
            System.out.println("Siang, tarif Rp 7000");  
        } else {  
            System.out.println("Malam, tarif Rp 10000");  
        }  
    }  
}
```

Penjelasan Source Code:

Pada program `parkirMall`, variabel `jam` bertipe data `int` dan diisi dengan nilai 10. Program ini menggunakan percabangan `if-else if-else` untuk menentukan tarif parkir berdasarkan nilai `jam`. Pertama, program memeriksa kondisi `if (jam < 12)`. Karena nilai `jam` adalah 10, maka kondisi tersebut bernilai benar sehingga program akan menjalankan perintah yang ada di dalam blok `if`.

PrintScreen Output:



Gambar 3.2 Output Soal No.3 b

Penjelasan Output:

Ketika program dijalankan dengan nilai `jam = 10`, output yang muncul pada terminal adalah `"Pagi, tarif Rp 5000"`. Output ini sesuai dengan logika program karena kondisi `if (jam < 12)` bernilai benar, sehingga program menjalankan perintah di dalam blok `if` dan tidak memeriksa kondisi lain. Hasil ini menunjukkan bahwa program dapat menentukan waktu parkir sebagai pagi hari dan menampilkan tarif yang sesuai, yaitu Rp 5000.

c. PrintScreen Source Code


```
File Edit Selection View Go Run Terminal Help
J parkirMall.java 1 x
D: > Pratikum_a1 > J parkirMall.java > parkirMall > main(String[])
1 public class parkirMall {
2     public static void main(String[] args) {
3         int jam = 20;
4         if (jam < 12) {
5             System.out.println(x:"Pagi, tarif Rp 5000");
6         } else if (jam < 18) {
7             System.out.println(x:"Siang, tarif Rp 7000");
8         } else {
9             System.out.println(x:"Malam, tarif Rp 10000");
10        }
11    }
12 }
```

G1A025039

Gambar 3.3 Source Code Soal No.3 c

Source Code:

```
public class parkirMall {
    public static void main(String[] args) {
        int jam = 20;
        if (jam < 12) {
            System.out.println("Pagi, tarif Rp 5000");
        } else if (jam < 18) {
            System.out.println("Siang, tarif Rp 7000");
        } else {
            System.out.println("Malam, tarif Rp 10000");
        }
    }
}
```

Penjelasan Source Code:

Variabel `jam` bertipe data `int` dan diberi nilai 20. Program kemudian memeriksa kondisi menggunakan `if-else if-else`. Pertama, kondisi `if (jam < 12)` dicek, tetapi bernilai salah karena 20 tidak lebih kecil dari 12. Selanjutnya, program memeriksa kondisi `else if (jam < 18)`, namun kondisi ini juga salah karena 20 tidak lebih kecil dari 18. Karena kedua kondisi tersebut tidak terpenuhi, maka program masuk ke bagian `else` dan memeriksa perintah di dalamnya. Dengan demikian, ketika `jam = 20`, program akan memilih blok `else` yang menunjukkan bahwa waktu parkir termasuk malam.

PrintScreen Output:

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Lenovo> & 'C:\Program Files\Java\jdk-24\bin\java-project\bin' 'parkirMall'
Malam, tarif Rp 10000
PS C:\Users\Lenovo>
```

G1A025039

Gambar 3.4 Output Soal No.3 c

Ketika program dijalankan dengan nilai jam = 20, output yang ditampilkan pada terminal adalah “Malam, tarif Rp 10000”. Hal ini terjadi karena kondisi if (jam < 12) dan else if (jam < 18) tidak terpenuhi, sehingga program masuk ke bagian else.

4. Rumah sakit Tiara Sella memiliki sebuah lift dengan sistem otomatis untuk mengecek jumlah orang di dalamnya. Sistem tersebut bekerja dengan aturan:

Jika jumlah orang ≤ 8 , maka layar menampilkan: “Lift bisa jalan”.

Jika jumlah orang > 8 , maka layar menampilkan: “Terlalu penuh, tunggu lift berikutnya”.

Mario diminta membuat program sederhana untuk menguji aturan tersebut.

Pertanyaan:

- Buatlah kode program menggunakan ternary operator untuk aturan lift tersebut.
- Uji program dengan jumlahOrang = 6 (bayangkan kondisi lift pagi hari saat belum terlalu ramai).
- Uji program dengan jumlahOrang = 10 (bayangkan kondisi lift siang hari saat penuh sesak).

Jawab:

b. PrintScreen Source Code



Gambar 4.1 Source Code Soal No.4 b

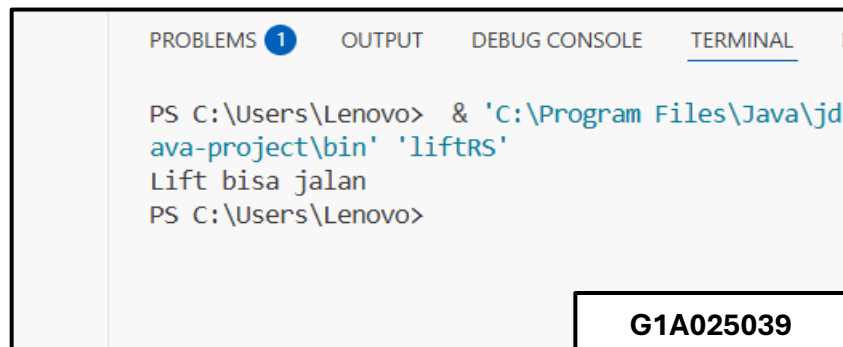
Source Code:

```
public class liftRS {  
    public static void main(String[] args) {  
        int jumlahOrang = 6; // Ubah angka ini untuk pengujian  
  
        String statusLift = (jumlahOrang <= 8) ? "Lift bisa jalan" : "Terlalu penuh, tunggu lift  
berikutnya";  
  
        System.out.println(statusLift);  
    }  
}
```

Penjelasan Source Code:

Variabel `jumlahOrang` bertipe data `int` dan diberi nilai `6`. Program kemudian menggunakan operator ternary untuk memeriksa apakah jumlah orang kurang dari atau sama dengan `8`. Jika kondisi `jumlahOrang <= 8` bernilai benar, maka variabel `statusLift` akan berisi teks `"Lift bisa jalan"`. Sebaliknya, jika kondisi bernilai salah, `statusLift` akan berisi teks `"Terlalu penuh, tunggu lift berikutnya"`.

PrintScreen Output:



```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\Lenovo> & 'C:\Program Files\Java\jdk-11.0.10\bin\java.exe' -cp 'C:\Program Files\Java\jdk-11.0.10\bin\java-project\bin' 'liftRS'
Lift bisa jalan
PS C:\Users\Lenovo>
```


G1A025039

Gambar 4.2 Output Soal No.4 b

Penjelasan Output:

Output program menampilkan teks `"Lift bisa jalan"`. Hal ini terjadi karena nilai variabel `jumlahOrang` adalah `6`, sedangkan syarat maksimum kapasitas lift adalah `8` orang. Kondisi `jumlahOrang <= 8` bernilai benar sehingga program memilih hasil `"Lift bisa jalan"` dan mencetaknya ke layar.

c. PrintScreen Source Code



```
droneAngin.java 1 liftRS.java 1 X
D:\Pratikum_a1> J liftRS.java > liftRS > main(String[])
1 public class liftRS {
2     public static void main(String[] args) {
3         int jumlahOrang = 10;
4
5         String statusLift = (jumlahOrang <= 8) ? "Lift bisa jalan" : "Terlalu penuh, tunggu lift berikutnya";
6
7         System.out.println(statusLift);
8     }
9 }
```

G1A025039

Gambar 4.3 Source Code Soal No.4 c

Source Code:

```
public class liftRS {
    public static void main(String[] args) {
        int jumlahOrang = 10; // Ubah angka ini untuk pengujian
```

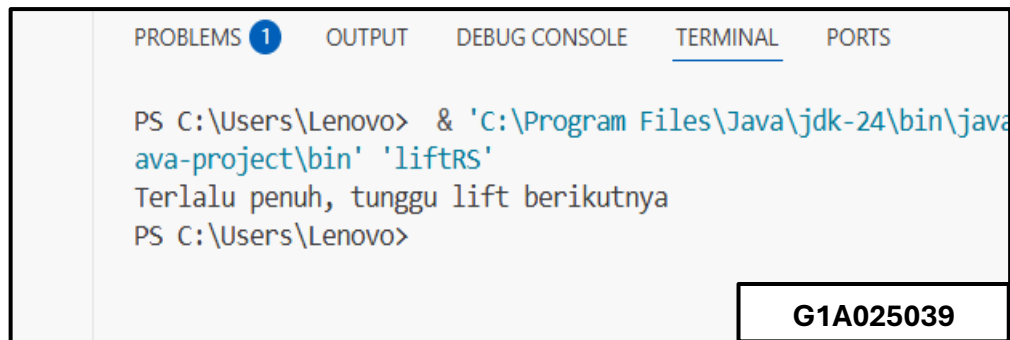
```
String statusLift = (jumlahOrang <= 8) ? "Lift bisa jalan" : "Terlalu penuh, tunggu lift berikutnya";
```

```
    System.out.println(statusLift);  
}  
}
```

Penjelasan Source Code:

Pada kode, variabel `jumlahOrang` diberi nilai 10. Program kemudian menggunakan operator ternary untuk mengecek kondisi, yaitu jika `jumlahOrang <= 8` maka lift bisa berjalan, sedangkan jika lebih dari 8 maka lift dinyatakan penuh. Karena nilai `jumlahOrang` adalah 10, kondisi `jumlahOrang <= 8` bernilai salah sehingga program menghasilkan output “Terlalu penuh, tunggu lift berikutnya”.

PrintScreen Output:



```
PS C:\Users\Lenovo> & 'C:\Program Files\Java\jdk-24\bin\java  
ava-project\bin' 'liftRS'  
Terlalu penuh, tunggu lift berikutnya  
PS C:\Users\Lenovo>
```

Gambar 4.4 Output Soal No.4 c

Penjelasan Output:

Program tersebut mencetak teks “Terlalu penuh, tunggu lift berikutnya” ke output. Hal ini terjadi karena pada source code sebelumnya variabel `jumlahOrang` diisi dengan nilai 10. Kondisi yang dicek adalah apakah `jumlahOrang <= 8`. Karena 10 lebih besar dari 8, kondisi bernilai salah sehingga program memilih bagian kedua dari operator ternary, yaitu pesan bahwa lift terlalu penuh.

5. Rifqi dan Zora adalah mahasiswa yang ingin mengakses koleksi e-book di perpustakaan digital kampus. Namun, sistem memiliki aturan akses sebagai berikut:

Jika mahasiswa sudah terdaftar sebagai anggota, maka dicek lagi:

Jika ia sudah membayar iuran, tampilkan: “Akses e-book dibuka”.

Jika ia belum membayar iuran, tampilkan: “Harap bayar iuran dulu”.

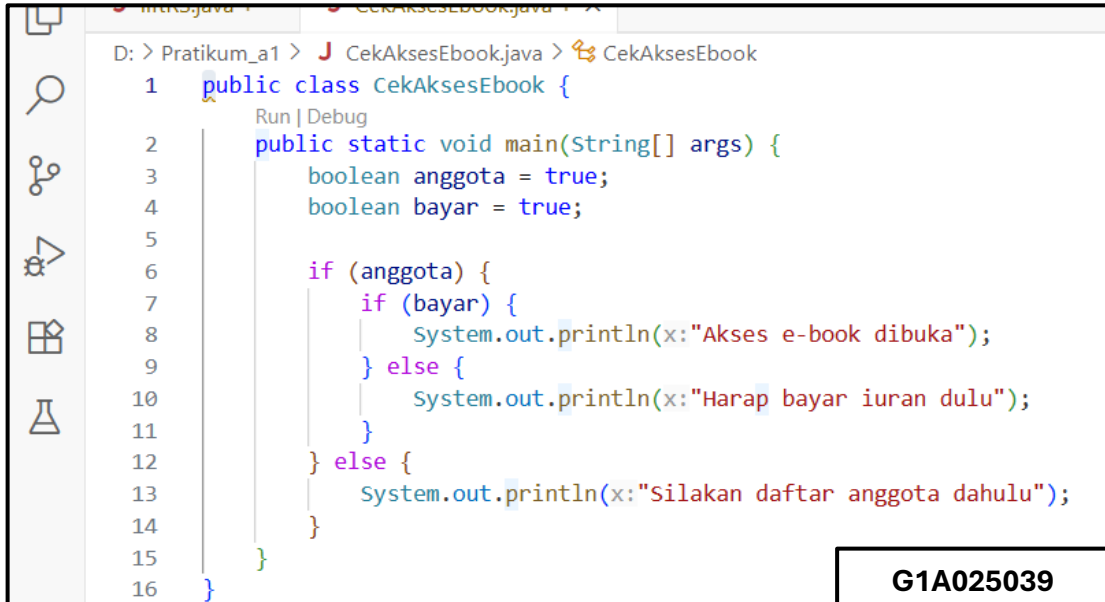
Jika mahasiswa bukan anggota, tampilkan: “Silakan daftar anggota dahulu”.

Mahasiswa tersebut kemudian meminta bantuan temannya yang belajar pemrograman untuk membuat simulasi aturan ini.

Pertanyaan:

- Buatlah kode program menggunakan nested if untuk aturan akses e-book di atas.
- Uji program dengan kondisi: anggota = true, bayar = true.
- Uji program dengan kondisi: anggota = true, bayar = false.

b. **PrintScreen Source Code**



```
1 public class CekAksesEbook {
2     public static void main(String[] args) {
3         boolean anggota = true;
4         boolean bayar = true;
5
6         if (anggota) {
7             if (bayar) {
8                 System.out.println(x:"Akses e-book dibuka");
9             } else {
10                System.out.println(x:"Harap bayar iuran dulu");
11            }
12        } else {
13            System.out.println(x:"Silakan daftar anggota dahulu");
14        }
15    }
16 }
```

G1A025039

Gambar 5.1 Source Code Soal No.5 b

Source Code:

```
public class CekAksesEbook {
    public static void main(String[] args) {
        boolean anggota = true;
        boolean bayar = true;

        if (anggota) {
            if (bayar) {
                System.out.println("Akses e-book dibuka");
            } else {
                System.out.println("Harap bayar iuran dulu");
            }
        } else {
            System.out.println("Silakan daftar anggota dahulu");
        }
    }
}
```

Penjelasan Source Code:

Pada kode yang ditampilkan, variabel anggota bernilai **true** yang berarti pengguna sudah terdaftar sebagai anggota, dan variabel bayar juga bernilai **true** yang berarti iuran sudah dibayar. Pertama, program memeriksa apakah pengguna adalah anggota. Karena nilainya **true**, maka pemeriksaan dilanjutkan ke kondisi berikutnya, yaitu status pembayaran. Karena nilai bayar juga **true**, program mengeksekusi perintah untuk mencetak pesan “Akses e-book dibuka”.

PrintScreen Output:

```
PS C:\Users\Lenovo> & 'C:\Program Files\Java\jdk-24\bin\java-project\bin' 'CekAksesEbook'
Akses e-book dibuka
PS C:\Users\Lenovo> 
```

G1A025039

Gambar 5.2 Output Soal No.5 b

Penjelasan Output:

Hasil output pada gambar menunjukkan teks “Akses e-book dibuka”. Output tersebut muncul karena pada program **CekAksesEbook** variabel anggota diset bernilai **true** dan variabel bayar juga bernilai **true**. Artinya pengguna sudah menjadi anggota sekaligus sudah membayar iuran,

c. PrintScreen Source Code

```
1 public class CekAksesEbook {
2     public static void main(String[] args) {
3         boolean anggota = true;
4         boolean bayar = false;
5
6         if (anggota) {
7             if (bayar) {
8                 System.out.println("Akses e-book dibuka");
9             } else {
10                System.out.println("Harap bayar iuran dulu");
11            }
12        } else {
13            System.out.println("Silakan daftar anggota dahulu");
14        }
15    }
16 }
```

G1A025039

Gambar 5.3 Source Code Soal No.5 c

Source Code:

```
public class CekAksesEbook {
    public static void main(String[] args) {
        boolean anggota = true;
        boolean bayar = false;

        if (anggota) {
```

```

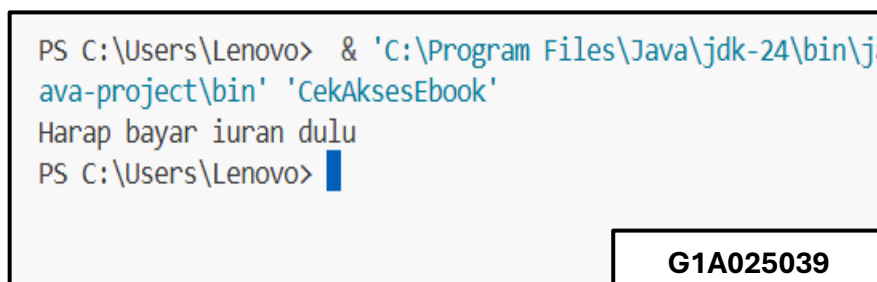
    if (bayar) {
        System.out.println("Akses e-book dibuka");
    } else {
        System.out.println("Harap bayar iuran dulu");
    }
} else {
    System.out.println("Silakan daftar anggota dahulu");
}
}
}
}

```

Penjelasan Source Code:

Pada kode ini, variabel anggota diberi nilai **true** sehingga pengguna dianggap sudah terdaftar sebagai anggota, sedangkan variabel bayar diberi nilai **false** yang berarti pengguna belum melakukan pembayaran iuran. Pertama, program memeriksa apakah pengguna adalah anggota. Karena nilainya **true**, program masuk ke blok **if** tersebut dan lanjut memeriksa status pembayaran. Pada pengecekan pembayaran, karena variabel bayar bernilai **false**, program tidak mencetak pesan “Akses e-book dibuka”, tetapi menampilkan pesan “Harap bayar iuran dulu”.

PrintScreen Output:



```

PS C:\Users\Lenovo> & 'C:\Program Files\Java\jdk-24\bin\java.exe' -cp 'C:\Users\Lenovo\Documents\Project\bin' 'CekAksesEbook'
Harap bayar iuran dulu
PS C:\Users\Lenovo>

```

Gambar 5.4 Output Soal No.5 c

Penjelasan Output:

Output pada gambar menampilkan teks “Harap bayar iuran dulu”. Hasil ini sesuai, di mana variabel anggota bernilai **true** sehingga pengguna sudah terdaftar sebagai anggota, tetapi variabel bayar bernilai **false** yang berarti pengguna belum melakukan pembayaran iuran. Karena kondisi pembayaran tidak terpenuhi, program mengeksekusi perintah pada blok **else** di dalam pengecekan kedua dan menampilkan pesan tersebut.

6. Sebuah drone harus memilih arah berdasarkan kondisi:

Jika angin datang dari utara (drone terbang ke selatan).

Jika angin datang dari barat (drone terbang ke timur).

Selain itu (drone hover atau diam di tempat).

Lengkapi kode berikut agar sesuai aturan:

```
String angin = "utara";

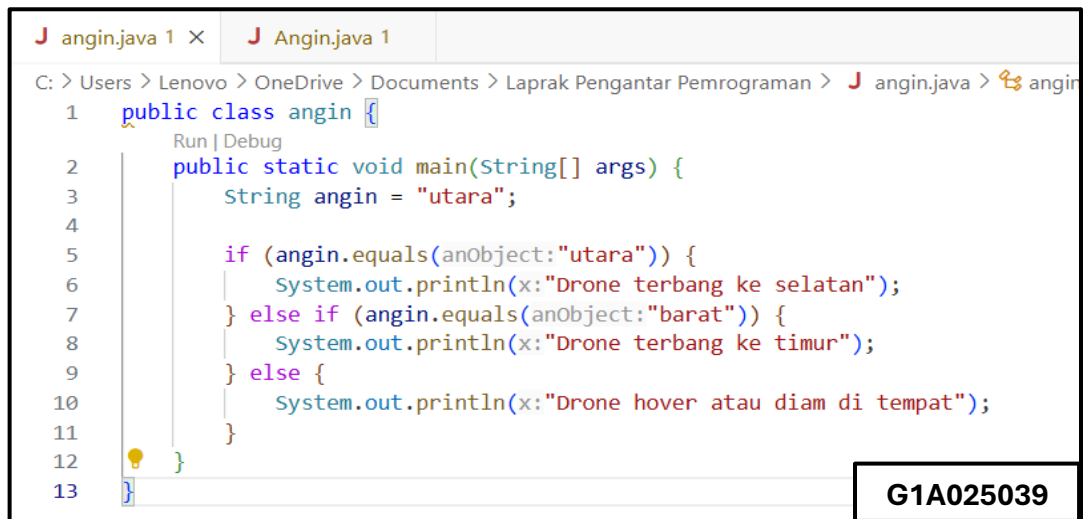
if (angin.equals("utara")) {
    System.out.println("Drone terbang ke selatan");
}
// lengkapi else if dan else di sini
```

Maka jawablah soal berikut

- Lengkapi bagian yang kosong.
- Uji dengan angin = "barat".
- Uji dengan angin = "selatan".

Jawab:

a. PrintScreen Source Code



```
1 public class angin {
2     public static void main(String[] args) {
3         String angin = "utara";
4
5         if (angin.equals("utara")) {
6             System.out.println("Drone terbang ke selatan");
7         } else if (angin.equals("barat")) {
8             System.out.println("Drone terbang ke timur");
9         } else {
10            System.out.println("Drone hover atau diam di tempat");
11        }
12    }
13 }
```

Gambar 6.1 Source Code Soal No.6 a

Source Code:

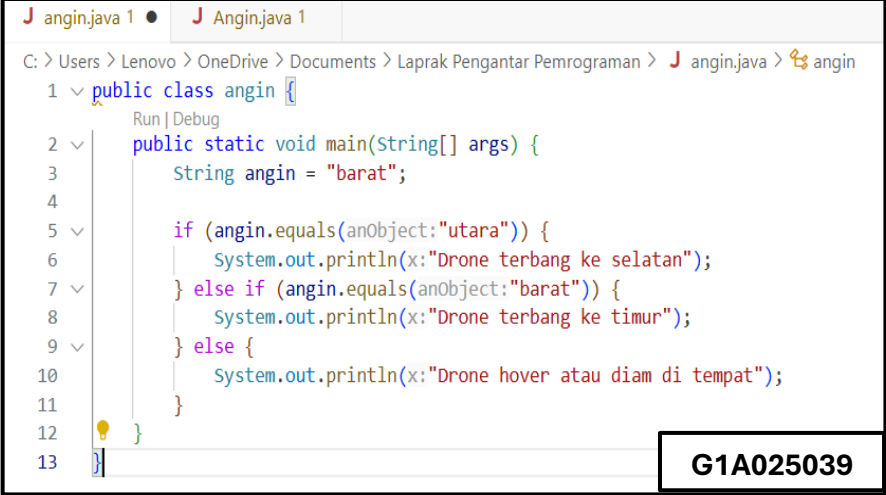
```
public static void main(String[] args) {
    String angin = "utara";

    if (angin.equals("utara")) {
        System.out.println("Drone terbang ke selatan");
    } else if (angin.equals("barat")) {
        System.out.println("Drone terbang ke timur");
    } else {
        System.out.println("Drone hover atau diam di tempat");
    }
}
```


Penjelasan Source Code:

Di dalam method main, variabel **angin** diisi dengan nilai "utara". Selanjutnya, program menggunakan struktur **if-else** untuk membandingkan isi variabel tersebut dengan beberapa kemungkinan nilai. Jika nilai **angin** sama dengan "utara", maka program akan mencetak pesan "Drone terbang ke selatan". Jika nilainya "barat", maka output yang ditampilkan adalah "Drone terbang ke timur". Jika nilainya bukan keduanya, maka program menampilkan pesan "Drone hover atau diam di tempat".

b.PrintScreen Source Code



```
1 public class angin {
2     public static void main(String[] args) {
3         String angin = "barat";
4
5         if (angin.equals(anObject:"utara")) {
6             System.out.println(x:"Drone terbang ke selatan");
7         } else if (angin.equals(anObject:"barat")) {
8             System.out.println(x:"Drone terbang ke timur");
9         } else {
10            System.out.println(x:"Drone hover atau diam di tempat");
11        }
12    }
13 }
```

G1A025039

Gambar 6.3 Source Code Soal No.6 b

Source Code:

```
public static void main(String[] args) {
    String angin = "barat";

    if (angin.equals("utara")) {
        System.out.println("Drone terbang ke selatan");
    } else if (angin.equals("barat")) {
        System.out.println("Drone terbang ke timur");
    } else {
        System.out.println("Drone hover atau diam di tempat");
    }
}
```

Penjelasan Source Code:

Program memeriksa arah angin berdasarkan **string** yang dimasukkan. Pertama, program akan membandingkan apakah nilai **Angin** sama dengan "utara". Karena nilainya "barat", maka

kondisi tersebut tidak terpenuhi. Selanjutnya, program masuk ke kondisi **else if** yang memeriksa apakah **angin** bernilai **"barat"**. Karena benar nilainya adalah **"barat"**, maka program akan menjalankan perintah di dalam blok tersebut dan mencetak teks **"Drone terbang ke timur"** ke layar.

PrintScreen Output:

```
PS C:\Users\Lenovo> & 'C:\Program Files\Java\jdk-24\bin\java-project\bin' 'droneAngin'
Drone terbang ke timur
PS C:\Users\Lenovo> |
```

G1A025039

Gambar 6.4 Output Soal No.6 b

Penjelasan Output:

Ketika program dijalankan, sistem melakukan pengecekan kondisi pertama **if (angin.equals("utara"))**, namun kondisi ini salah karena nilainya bukan **"utara"**. Selanjutnya program melanjutkan ke kondisi kedua **else if (angin.equals("barat"))**. Karena kondisi ini benar, maka perintah yang ada di dalam blok tersebut dijalankan, yaitu menampilkan teks **"Drone terbang ke timur"**.

c. PrintScreen Source Code

```
Explorer (Ctrl+Shift+E)  J Angin.java 1
C: > Users > Lenovo > OneDrive > Documents > Laprak Pengantar Pemrograman > J angin.java > J angin
1  public class angin {
2      Run | Debug
3      public static void main(String[] args) {
4          String angin = "selatan";
5
6          if (angin.equals(anObject:"utara")) {
7              System.out.println(x:"Drone terbang ke selatan");
8          } else if (angin.equals(anObject:"barat")) {
9              System.out.println(x:"Drone terbang ke timur");
10         } else {
11             System.out.println(x:"Drone hover atau diam di tempat");
12         }
13     }
|
```

G1A025039

Gambar 6.5 Source Code Soal No.6 c

Source Code:

```
public static void main(String[] args) {
    String angin = "selatan";

    if (angin.equals("utara")) {
        System.out.println("Drone terbang ke selatan");
    } else if (angin.equals("barat")) {
        System.out.println("Drone terbang ke timur");
    }
}
```

```

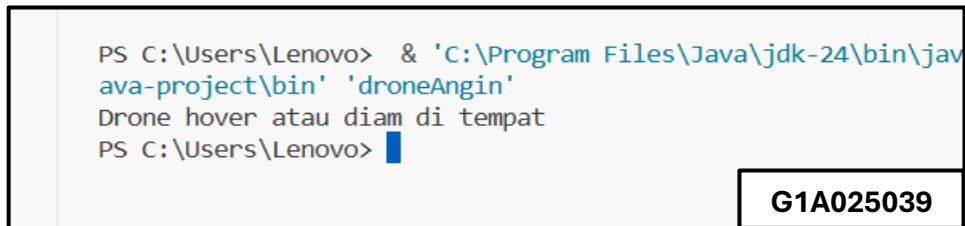
    } else {
        System.out.println("Drone hover atau diam di tempat");
    }
}
}
}

```

Penjelasan Source Code:

Variabel **angin** diisi dengan nilai "**selatan**". Nilai ini kemudian diuji menggunakan **if-else if-else**. Program pertama kali memeriksa apakah nilai variabel sama dengan "**utara**". Jika kondisi ini terpenuhi, drone akan diarahkan untuk terbang ke selatan. Karena nilai yang diberikan adalah "**selatan**", kondisi ini tidak terpenuhi. Selanjutnya, program memeriksa apakah nilai variabel sama dengan "**barat**". Jika benar, drone akan terbang ke timur. Namun, karena nilai yang diberikan bukan "**barat**", kondisi kedua juga tidak terpenuhi. Karena kedua syarat sebelumnya salah, program otomatis masuk ke bagian **else** yang berisi instruksi agar drone tidak bergerak, atau berada dalam kondisi hover.

PrintScreen Output:



```

PS C:\Users\Lenovo> & 'C:\Program Files\Java\jdk-24\bin\java-project\bin' 'droneAngin'
Drone hover atau diam di tempat
PS C:\Users\Lenovo>

```

Gambar 6.6 Output Soal No.6 c

Penjelasan Output:

Output yang ditampilkan pada program adalah "**Drone hover atau diam di tempat**". Hal ini terjadi karena pada kode sumber, variabel **Angin** diberi nilai "**selatan**". Saat program dijalankan, kondisi pertama memeriksa apakah **Angin** bernilai "**utara**", namun hasilnya salah. Kondisi kedua memeriksa apakah nilainya "**barat**", dan hasilnya juga salah. Karena kedua kondisi tersebut tidak terpenuhi

7. Melzaqi menulis program untuk mengecek apakah mobil boleh masuk jalan tol. Aturannya: Jika saldo e-Toll $\geq 20.000 \rightarrow$ boleh masuk. Jika tidak, tidak boleh masuk.

Kode Melzaqi:

```

public class Main {
    public static void main(String[] args) {
        int saldo = 15000;
        if (saldo >= 20000);
        {
            System.out.println("Boleh masuk tol");
        } else {
            System.out.println("Saldo tidak cukup");
        }
    }
}

```

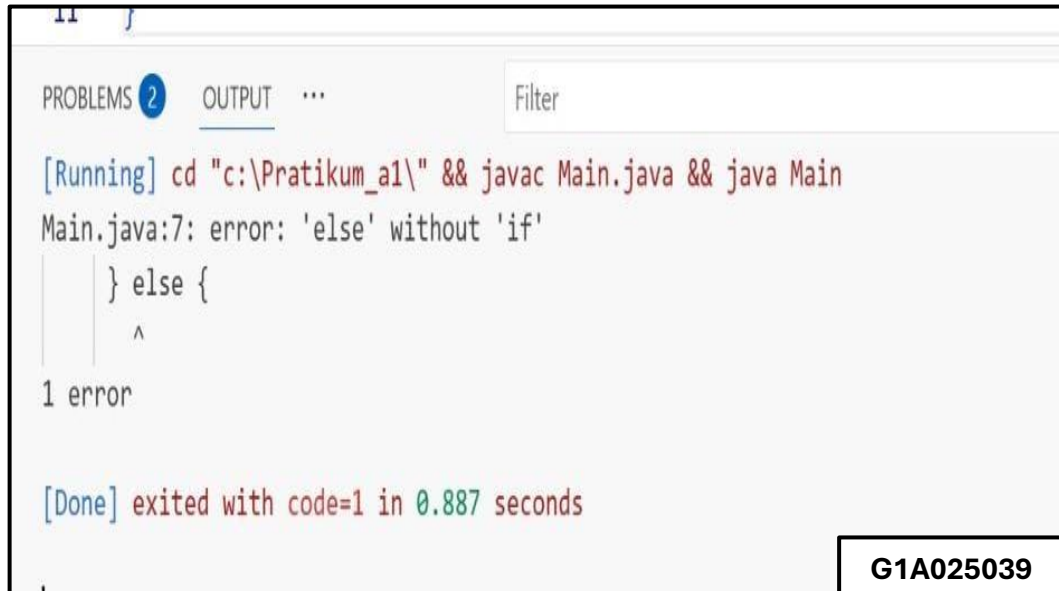
Maka,

a. Apa output yang didapatkan jika ada kesalahan tunjukkan dimana kesalahan pada kode program tersebut

b. Perbaiki kode program tersebut.

Jawab:

a.



The screenshot shows an IDE's output window with the following text:

```
PROBLEMS 2 OUTPUT ... Filter
[Running] cd "c:\Pratikum_a1\" && javac Main.java && java Main
Main.java:7: error: 'else' without 'if'
    } else {
      ^
1 error

[Done] exited with code=1 in 0.887 seconds
```

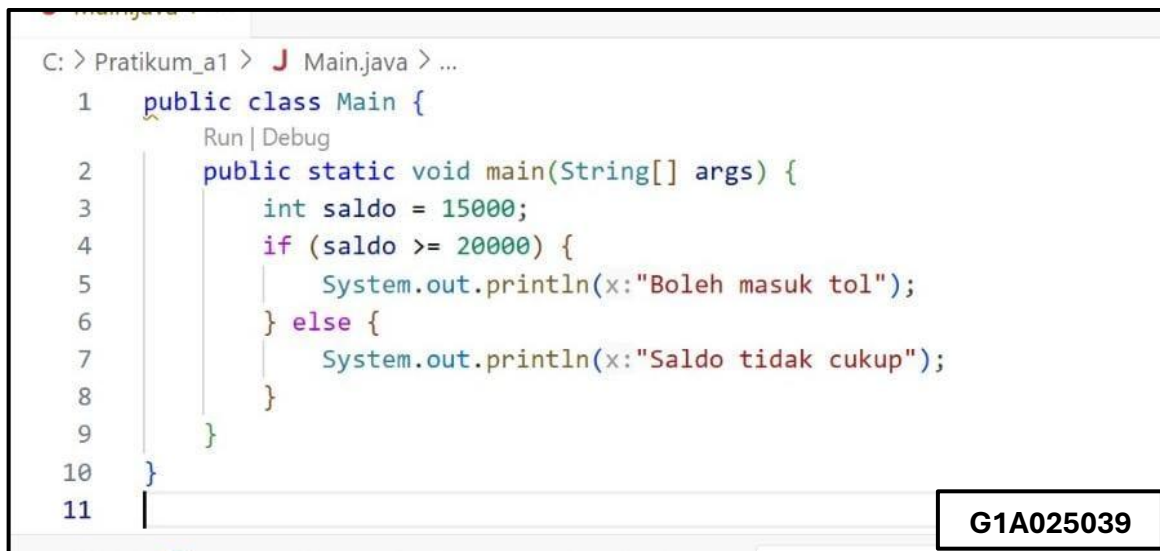
G1A025039

Gambar 7.1 Output Soal No.7 a

Penjelasan Output:

Kesalahan yang terjadi karena di bagian `if (saldo >= 20000);`, diakhirnya ditulis dengan tanda titik koma (;) di akhirnya. Karena itu, java menganggap pertanyaan sudah selesai disitu, sehingga blok kurung kurawal {...} berikutnya tidak berhubungan dengan `if`. Dan ketika program melanjutkan program nya ke bagian `else {...}`, java tidak menemukan pasangannya yaitu `if`, sehingga muncul eror di output kode tersebut.

b. PrintScreen Source Code



The screenshot shows the source code of Main.java in an IDE:

```
C: > Pratikum_a1 > J Main.java > ...
1 public class Main {
    Run | Debug
2     public static void main(String[] args) {
3         int saldo = 15000;
4         if (saldo >= 20000) {
5             System.out.println(x:"Boleh masuk tol");
6         } else {
7             System.out.println(x:"Saldo tidak cukup");
8         }
9     }
10 }
11
```

G1A025039

Gambar 7.2 Source Code Soal No.7 b

Source Code:

```
public class Main {  
    public static void main(String[] args) {  
        int saldo = 15000;  
        if (saldo >= 20000) {  
            System.out.println("Boleh masuk tol");  
        } else {  
            System.out.println("Saldo tidak cukup");  
        }  
    }  
}
```

Penjelasan Source Code:

Dibuat sebuah variabel bertipe `int` bernama `saldo` dengan nilai 15000. Selanjutnya terdapat struktur pengkondisian `if-else` yang digunakan untuk memeriksa nilai saldo tersebut. Pada bagian kondisi `if`, program mengecek apakah saldo lebih besar atau sama dengan 20000. Jika benar, maka akan ditampilkan pesan “Boleh masuk tol”. Namun, jika saldo kurang dari 20000, program akan mengeksekusi bagian `else` sehingga menampilkan pesan “Saldo tidak cukup”.

PrintScreen Output:



Gambar 7.3 Output Soal No.7 b

Penjelasan Output:

Saat program dijalankan, bagian kondisi `if (saldo >= 20000)` diperiksa. Karena nilai saldo (15000) lebih kecil dari 20000, kondisi tersebut bernilai `false`, sehingga perintah di dalam blok `if` tidak dijalankan. Program kemudian masuk ke blok `else` dan menampilkan pesan “Saldo tidak cukup”.

KESIMPULAN DAN SARAN

Kesimpulan

Berdasarkan serangkaian latihan soal pemrograman yang telah dikerjakan, dapat disimpulkan bahwa penerapan konsep dasar algoritma dan struktur kontrol merupakan aspek yang sangat penting dalam menyelesaikan masalah sehari-hari dengan bantuan komputer. Setiap soal menggambarkan kasus nyata yang diubah menjadi logika pemrograman melalui penggunaan variabel, tipe data, operator, dan percabangan.

Pada soal pertama, konsep logika digunakan untuk menentukan apakah suatu ruangan memenuhi kriteria sebagai ruang kelas. Hasil menunjukkan bahwa logika perbandingan dapat digunakan untuk memvalidasi syarat tertentu. Pada soal kedua, percabangan sederhana mampu menggambarkan sistem IoT berupa lampu pintar yang menyala atau mati sesuai keberadaan orang. Soal ketiga memberikan gambaran nyata bagaimana penggunaan *if...else if* bisa digunakan untuk menerapkan aturan tarif parkir berdasarkan waktu masuk kendaraan.

Lebih lanjut, pada soal keempat diperlihatkan pemanfaatan *ternary operator* untuk membuat program lebih ringkas dan efisien. Pada soal kelima, penggunaan *nested if* digunakan untuk mengecek kondisi bertingkat seperti status keanggotaan dan pembayaran iuran perpustakaan digital. Pada soal keenam, logika *if...else* digunakan untuk mengendalikan arah terbang drone berdasarkan arah angin, sedangkan pada soal ketujuh, pembelajaran mengenai kesalahan penulisan sintaks menegaskan pentingnya ketelitian dalam pemrograman agar program dapat berjalan sesuai dengan logika yang diinginkan.

Secara keseluruhan, latihan ini menekankan bahwa pemahaman terhadap dasar-dasar pemrograman tidak hanya penting untuk menulis kode, tetapi juga untuk memecahkan masalah nyata melalui pemodelan logika yang tepat. Dengan menguasai variabel, operator, dan percabangan, seorang programmer dapat membuat program sederhana yang bermanfaat dalam berbagai bidang, mulai dari arsitektur, IoT, transportasi, kesehatan, hingga sistem digital.

Saran

Untuk meningkatkan pemahaman dalam pemrograman, ada beberapa saran yang dapat dipertimbangkan. Pertama, sebaiknya tidak hanya memahami teori, tetapi juga sering berlatih menuliskan kode program secara langsung. Dengan banyak latihan, kesalahan maupun logika akan lebih mudah dikenali dan dihindari. Kedua, pemahaman terhadap aturan harus dikembangkan, karena inti dari pemrograman adalah menerjemahkan aturan tersebut menjadi algoritma dan instruksi yang dapat dipahami komputer. Selain itu, sebaiknya juga mempelajari lebih dari satu bahasa pemrograman agar dapat membandingkan cara penulisan yang berbeda, namun tetap berlandaskan logika yang sama.

DAFTAR PUSTAKA

Irmayanti, H. (2020). *Modul 1 Dasar Pemrograman Java* [Teaching resource]. Universitas Komputer Indonesia. Diunduh dari Repository UNIKOM

Lestari Pratiwi, E. (2020). *Konsep dasar algoritma dan pemrograman dengan bahasa Java*. Poliban Press.

https://press.poliban.ac.id/uploads/file/Buku_Algo_Pemrog_Java_9786237694458.pdf

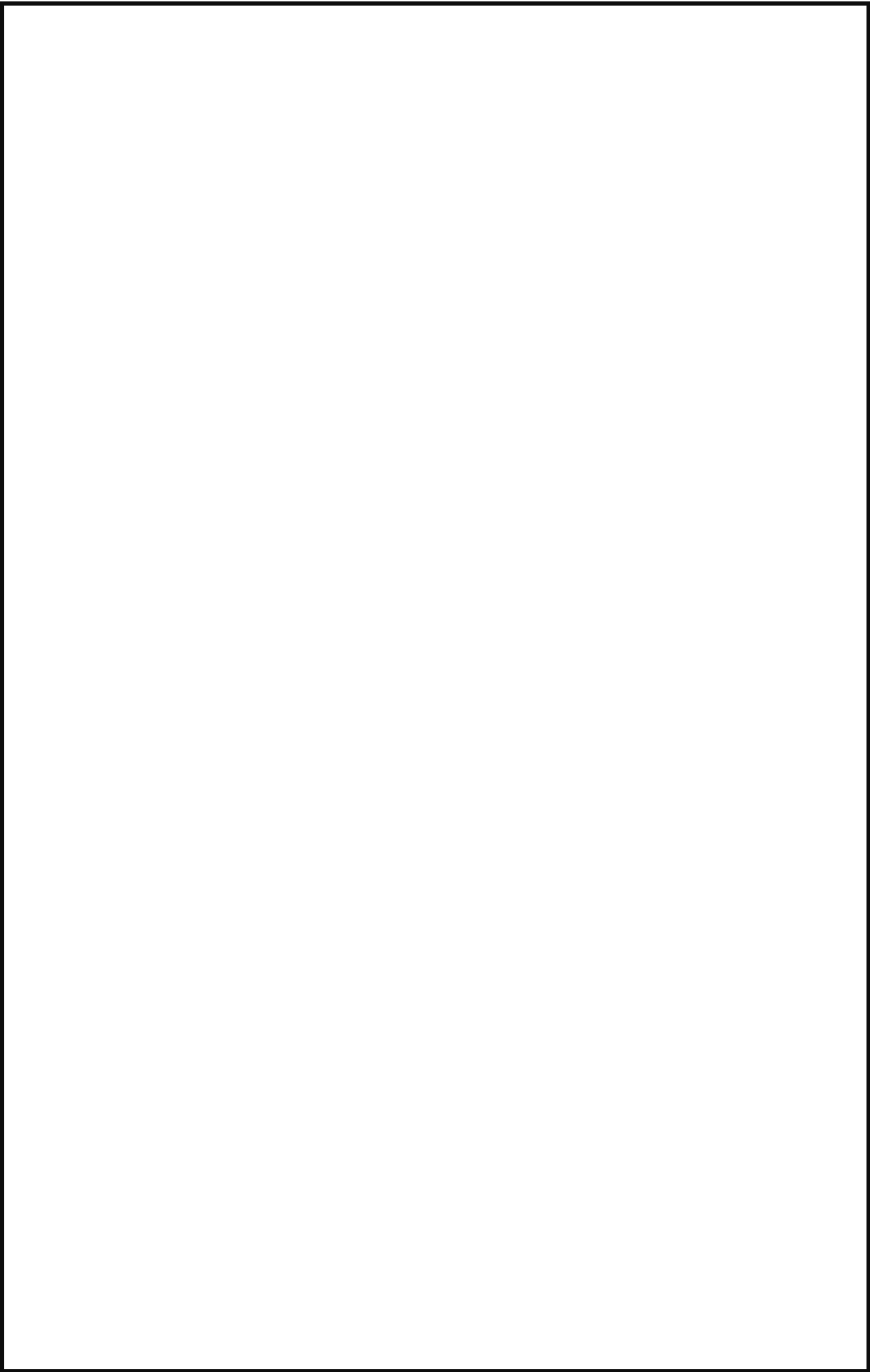
[Poliban Press](#)

Munir, R. (2012). *Pengantar ke algoritma* (Bab 1).

Institut Teknologi Bandung.

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Buku/Algoritma%20dan%20Pemrograman%20I%20%28Edisi%203%29/Bab-1%20Pengantar%20ke%20Algoritma.pdf>







**KEMENTERIAN PENDIDIKAN TINGGI, SAINS DAN
TEKNOLOGI
UNIVERSITAS BENGKULU
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA**

Jl. Wr. Supratman Kandang Limun, Bengkulu Bengkulu 38371 A

**LEMBAR ASISTENSI
PRAKTIKUM PROYEK PENGANTAR PEMROGRAMAN**

Nama Mahasiswa : Ariqah Naurah
NPM : G1A025039
Dosen : 1. Funny Farady Coastera, S.Kom, M.T
2. Arie Vatesia, S.T.,M.TI, Ph.D
Asisten : 1. Davi Sulaiman (G1A022001)
2. Julia Mayang Sari (G1A022010)
3. Evelyn Unike Aritonang (G1A022024)
4. Reksi Hendra (G1A022032)
5. Ade Irawan (G1A022083)
6. Azzahra Faranisa (G1A023010)
7. Lio Kusnata (G1A023013)
8. Ahsan Nurfauzi (G1A024002)
9. Gusti Revalion Putrada (G1A024081)
10. Lantera Meusanah (G1A024105)

Laporan Praktikum	Catatan dan Tanda Tangan
Laporan Praktikum ke - 3	