

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMOGRAMAN
CONDITIONAL STATEMENT
PEKAN 4

Disusun Oleh :

KINAYA NOVRYA MANDA
(2511531016)

Dosen Pengampu :

DR. WAHYUDI, S.T, M.T

Asisten Praktikum :

MUHAMMAD ZAKI AL HAFIZ



DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Puji syukur penulis sampaikan kehadirat Allah SWT. Salawat dan salam disampaikan kepada Nabi Muhammad SAW. Karena thaufik dan hidayah-Nya, laporan praktikum Java ini dapat diselesaikan dengan baik dan tepat waktu. Laporan ini disusun sebagai salah satu bentuk pertanggungjawaban dari kegiatan praktikum yang telah dilaksanakan, sekaligus sebagai sarana untuk memperdalam pemahaman mengenai konsep dasar serta penerapan bahasa pemrograman Java.

Melalui praktikum ini, penulis memperoleh pengalaman langsung dalam memahami struktur kontrol yang memungkinkan program membuat keputusan dan menjalankan blok kode yang berbeda berdasarkan kondisi tertentu menggunakan Java. Diharapkan laporan ini dapat memberikan gambaran yang jelas mengenai materi yang telah dipelajari serta hasil dari percobaan yang dilakukan selama praktikum berlangsung.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna, baik dari segi isi maupun penyajiannya. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa yang akan datang.

Akhir kata, penulis mengucapkan terima kasih kepada dosen pengampu, asisten laboratorium, serta semua pihak yang telah memberikan bimbingan, arahan, dan dukungan sehingga laporan praktikum ini dapat tersusun. Semoga laporan ini dapat bermanfaat bagi pembaca, khususnya dalam memahami dasar-dasar pemrograman Java.

Padang, September 2025

Penulis

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR LAMPIRAN	iii
BAB 1 PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Tujuan Praktikum.....	2
1.3 Manfaat Praktikum	2
BAB 2 PEMBAHASAN	
2.1 Kode Program	3
2.2 Langkah Kerja	11
2.3 Analisis Hasil.....	19
BAB 3 KESIMPULAN	
3.1 Hasil Praktikum.....	21
3.2 Saran Pengembangan	24
DAFTAR KEPUSTAKAAN	25

DAFTAR LAMPIRAN

Kode Program 2.1 (Kode Program LatIf1).....	3
Kode Program 2.2 (Kode Program Ifelse1).....	4
Kode Program 2.3 (Kode Program MultiIf)	6
Kode Program 2.4 (Kode Program Nilai).....	8
Kode Program 2.5 (Kode Program NamaBulan)	9
Gambar 2.1 (Langkah 1 LatIf1)	11
Gambar 2.2 (Langkah 2 LatIf1)	11
Gambar 2.3 (Langkah 3 LatIf1)	11
Gambar 2.4 (Langkah 4 LatIf1)	12
Gambar 2.5 (Langkah 1 Ifelse1)	12
Gambar 2.6 (Langkah 2 Ifelse1)	13
Gambar 2.7 (Langkah 3 Ifelse1)	13
Gambar 2.8 (Langkah 4 Ifelse1)	13
Gambar 2.9 (Langkah 1 MultiIf).....	14
Gambar 2.10 (Langkah 2 MultiIf).....	14
Gambar 2.11 (Langkah 3 MultiIf).....	15
Gambar 2.12 (Langkah 1 Nilai)	15
Gambar 2.13 (Langkah 2 Nilai)	16
Gambar 2.14 (Langkah 3 Nilai)	16
Gambar 2.15 (Langkah 4 Nilai)	16
Gambar 2.16 (Langkah 5 Nilai)	17
Gambar 2.17 (Langkah 1 NamaBulan).....	17
Gambar 2.18 (Langkah 2 NamaBulan).....	18
Gambar 2.19 (Langkah 3 NamaBulan).....	18

Gambar 2.20 (Langkah 4 NamaBulan).....	19
Gambar 3.1 (Hasil Praktikum LatIf1).....	21
Gambar 3.2 (Hasil Praktikum Ifelse1)	21
Gambar 3.3 (Hasil Praktikum MultiIf)	22
Gambar 3.4 (Hasil Praktikum Nilai).....	23
Gambar 3.5 (Hasil Praktikum NamaBulan)	23

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pemrograman komputer tidak hanya berfokus pada perintah berurutan, tetapi juga pada pengambilan keputusan yang menentukan jalannya eksekusi program. Dalam hal ini, *conditional statement* menjadi salah satu konsep dasar yang sangat penting untuk dipahami. Dengan *conditional statement*, program dapat merespons kondisi tertentu dan memilih instruksi mana yang harus dijalankan, sehingga membuat program lebih interaktif dan sesuai dengan kebutuhan pengguna.

Pada bahasa pemrograman Java, *conditional statement* diwujudkan melalui struktur seperti *if*, *if-else*, *if-else-if*, dan *switch*. Setiap struktur memiliki fungsi dan penerapan yang berbeda sesuai dengan kompleksitas masalah yang dihadapi. Misalnya, penggunaan *if-else* cocok untuk kondisi sederhana, sementara *switch* lebih efektif digunakan untuk banyak pilihan kondisi yang lebih spesifik. Praktikum ini menjadi sarana bagi mahasiswa untuk memahami bagaimana cara mengimplementasikan percabangan logika dalam Java secara tepat.

Selain itu, penguasaan *conditional statement* juga menjadi pondasi penting untuk mempelajari materi pemrograman lanjutan. Konsep percabangan ini nantinya akan berkaitan erat dengan perulangan, fungsi, algoritma, dan struktur data. Dengan melakukan praktikum, mahasiswa tidak hanya belajar teori, tetapi juga melatih kemampuan logika, analisis, dan keterampilan pemrograman secara langsung. Hal ini akan membantu mahasiswa dalam menyusun solusi yang efisien serta membiasakan pola pikir sistematis dalam menyelesaikan masalah nyata melalui pemrograman.

1.2 Tujuan Praktikum

Tujuan dilakukannya praktikum ini adalah sebagai berikut :

1. Memahami konsep dasar conditional statement pada Java.
2. Mengetahui cara penggunaan *if*, *if-else*, *if-else-if*, dan *switch*.
3. Melatih keterampilan menyusun logika percabangan dalam pemrograman.
4. Mengimplementasikan *conditional statement* pada kasus sederhana.
5. Meningkatkan kemampuan berpikir logis dan sistematis dalam memecahkan masalah dengan program.

1.3 Manfaat Praktikum

Manfaat dilakukannya praktikum ini adalah sebagai berikut :

1. Memberikan pemahaman dasar tentang logika percabangan sebagai fondasi pemrograman.
2. Membekali mahasiswa untuk membuat program yang dinamis, interaktif, dan fleksibel.
3. Menjadi dasar untuk mempelajari materi lebih lanjut seperti *looping*, fungsi, dan algoritma.
4. Melatih keterampilan berpikir kritis dalam menentukan kondisi dan alur program.
5. Membantu mahasiswa menguasai praktik pemrograman Java yang dapat diterapkan pada proyek nyata.

BAB II

PEMBAHASAN

2.1 Kode Program

2.1.1 LatIf1

```

1 package Pekan4;
2 import java.util.Scanner;
3 public class latIf1 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         double IPK;
8         Scanner input=new Scanner (System.in);
9         System.out.print("Input IPK Anda = ");
10        IPK=input.nextDouble();
11        if (IPK>2.75) {
12            System.out.println("Anda Lulus Sangat Memuaskan dengan IPK "+IPK);
13        }
14    }
15 }
16 }
17

```

Kode Program 2.1 (Kode Program LatIf1)

Uraian Kode Program :

1. Deklarasi dan Input Data

- Program menggunakan *Scanner* untuk membaca input dari pengguna.
- Variabel IPK bertipe *double* dideklarasikan untuk menyimpan nilai IPK yang dimasukkan.
- Baris `System.out.print("Input IPK Anda = ");` berfungsi menampilkan pesan agar pengguna memasukkan nilai IPK.
- Nilai IPK kemudian dibaca dengan `input.nextDouble();` dan disimpan dalam variabel IPK.

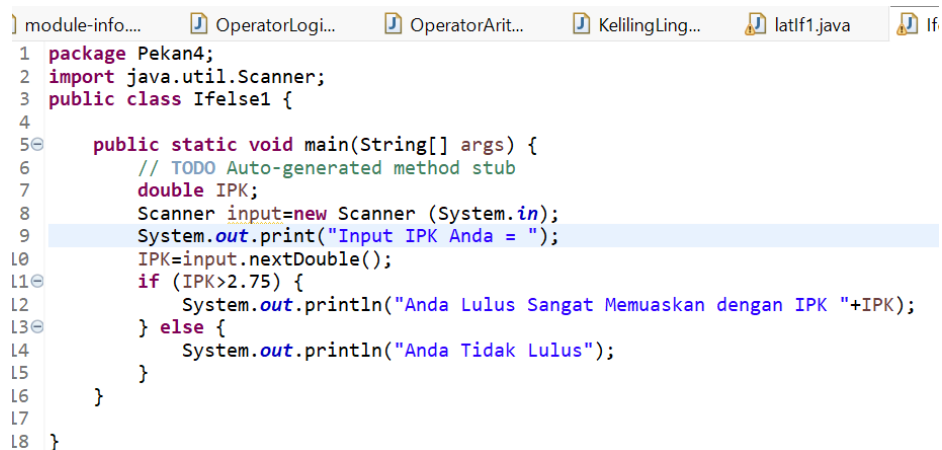
2. Percabangan (*Conditional Statement*)

- Program menggunakan pernyataan *if* untuk memeriksa kondisi nilai IPK.
- Kondisi yang dicek adalah *if* ($IPK > 2.75$). Artinya, program akan menjalankan perintah di dalam blok *if* hanya jika nilai IPK lebih besar dari 2.75.

3. Output Program

- Jika kondisi benar (IPK lebih besar dari 2.75), maka program akan menampilkan pesan: "Anda Lulus Sangat Memuaskan dengan IPK <nilai IPK>".
- Jika kondisi salah (IPK kurang dari atau sama dengan 2.75), program tidak menampilkan pesan apa pun karena tidak ada blok *else* yang ditulis.

2.1.2 Ifelse1



```

1 package Pekan4;
2 import java.util.Scanner;
3 public class Ifelse1 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         double IPK;
8         Scanner input=new Scanner (System.in);
9         System.out.print("Input IPK Anda = ");
10        IPK=input.nextDouble();
11        if (IPK>2.75) {
12            System.out.println("Anda Lulus Sangat Memuaskan dengan IPK "+IPK);
13        } else {
14            System.out.println("Anda Tidak Lulus");
15        }
16    }
17
18 }

```

Kode Program 2.2 (Kode Program Ifelse1)

Uraian Kode Program :

1. Deklarasi dan Input Data

- Program menggunakan *Scanner* untuk membaca input dari pengguna.
- Variabel IPK bertipe *double* dideklarasikan untuk menyimpan nilai IPK yang dimasukkan.
- Baris *System.out.print("Input IPK Anda = ");* berfungsi menampilkan pesan agar pengguna memasukkan nilai IPK.
- Nilai IPK kemudian dibaca menggunakan *input.nextDouble();* dan disimpan ke variabel IPK.

2. Percabangan (*If-Else Statement*)

- Program menggunakan pernyataan *if-else* untuk menentukan hasil berdasarkan nilai IPK.
- Kondisi yang dicek adalah *if* ($IPK > 2.75$). Jika kondisi ini benar, maka program akan mengeksekusi blok perintah dalam *if*.
- Jika kondisi salah ($IPK \leq 2.75$), program akan mengeksekusi perintah dalam blok *else*.

3. Output Program

- Jika nilai IPK lebih besar dari 2.75, maka program menampilkan pesan: "Anda Lulus Sangat Memuaskan dengan IPK <nilai IPK>".
- Jika nilai IPK kurang dari atau sama dengan 2.75, program akan menampilkan pesan: "Anda Tidak Lulus".

2.1.3 Multilf

```

1 package Pekan4;
2 import java.util.Scanner;
3 public class multiIf {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         int umur;
8         char sim;
9         Scanner a= new Scanner(System.in);
10        System.out.print("Input umur anda: ");
11        umur= a.nextInt();
12        System.out.print("Apakah Anda Sudah Punya Sim C: ");
13        sim=a.next().charAt(0);
14        a.close();
15        if((umur >= 17)&&(sim=='y')) {
16            System.out.println("Anda Sudah dewasa dan boleh bawa motor");
17        }
18        if((umur >= 17)&&(sim!='y')) {
19            System.out.println("Anda Sudah dewasa tetapi tidak boleh bawa motor");
20        }
21        if((umur < 17)&&(sim=='y')) {
22            System.out.println("Anda Belum Cukup Umur bawa motor");
23        }
24        if((umur < 17)&&(sim=='y')) {
25            System.out.println("Anda Belum Cukup Umur punya SIM");
26        }
27    }
28
29 }

```

Kode Program 2.3 (Kode Program Multilf)

Uraian Kode Program :

1. Deklarasi Variabel dan Input Data

- Program mendeklarasikan dua variabel: umur bertipe *int* dan sim bertipe *char*.
- *Scanner* digunakan untuk membaca input dari pengguna.
- Program meminta input umur melalui *System.out.print("Input umur anda: ");* lalu menyimpan hasilnya pada variabel umur.
- Program kemudian menanyakan apakah pengguna sudah memiliki SIM C dengan *System.out.print("Apakah Anda Sudah Punya Sim C: ");*.
- Input SIM disimpan ke variabel sim dengan hanya mengambil 1 karakter (*charAt(0)*).

2. Logika Percabangan (*Multi If*)

- Program menggunakan beberapa kondisi *if* untuk menentukan output berdasarkan kombinasi umur dan kepemilikan SIM.
- Jika umur ≥ 17 dan sim == 'y', maka tampil pesan: "Anda Sudah dewasa dan boleh bawa motor".
- Jika umur ≥ 17 dan sim != 'y', maka tampil pesan: "Anda Sudah dewasa tetapi tidak boleh bawa motor".
- Jika umur < 17 dan sim != 'y', maka tampil pesan: "Anda Belum Cukup Umur bawa motor".
- Jika umur < 17 dan sim == 'y', maka tampil pesan: "Anda Belum Cukup Umur punya SIM".

3. Output Program

- Output yang ditampilkan bergantung pada kombinasi input umur dan kepemilikan SIM:

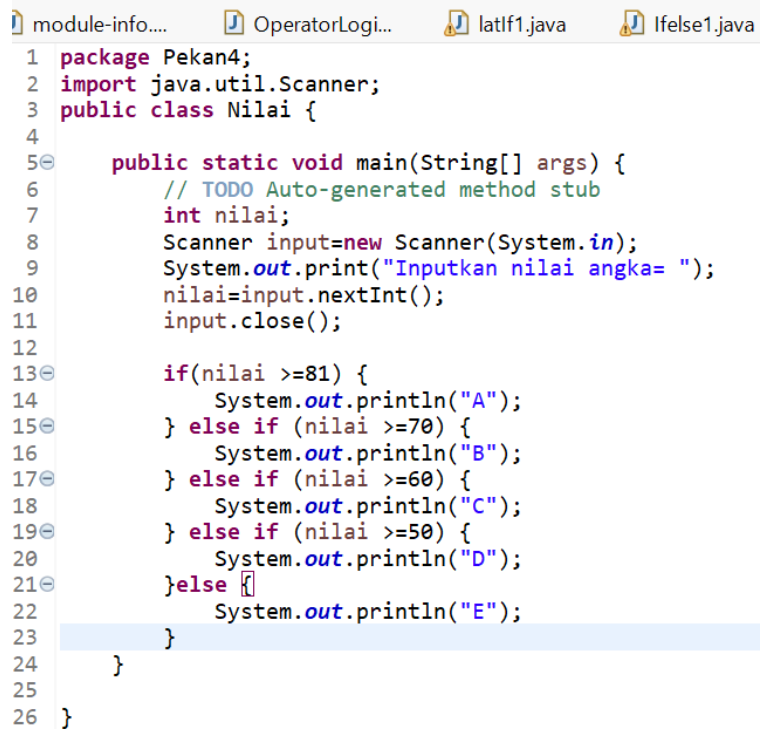
Umur ≥ 17 dan sudah punya SIM \rightarrow boleh bawa motor.

Umur ≥ 17 tapi tidak punya SIM \rightarrow tidak boleh bawa motor.

Umur < 17 dan tidak punya SIM \rightarrow belum cukup umur untuk bawa motor.

Umur < 17 tapi sudah punya SIM \rightarrow belum cukup umur punya SIM.

2.1.4 Nilai



```

1 package Pekan4;
2 import java.util.Scanner;
3 public class Nilai {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         int nilai;
8         Scanner input=new Scanner(System.in);
9         System.out.print("Inputkan nilai angka= ");
10        nilai=input.nextInt();
11        input.close();
12
13        if(nilai >=81) {
14            System.out.println("A");
15        } else if (nilai >=70) {
16            System.out.println("B");
17        } else if (nilai >=60) {
18            System.out.println("C");
19        } else if (nilai >=50) {
20            System.out.println("D");
21        } else {
22            System.out.println("E");
23        }
24    }
25 }
26 }

```

Kode Program 2.4 (Kode Program Nilai)

Uraian Kode Program :

1. Deklarasi Variabel dan Input Data

- Program mendeklarasikan variabel nilai bertipe *int* untuk menyimpan input angka.
- Menggunakan *Scanner* untuk membaca input dari pengguna.
- Baris *System.out.print("Inputkan nilai angka= ");* digunakan untuk meminta pengguna memasukkan nilai.
- Nilai yang dimasukkan disimpan ke variabel nilai dengan *input.nextInt()*. Setelah itu, scanner ditutup dengan *input.close()*.

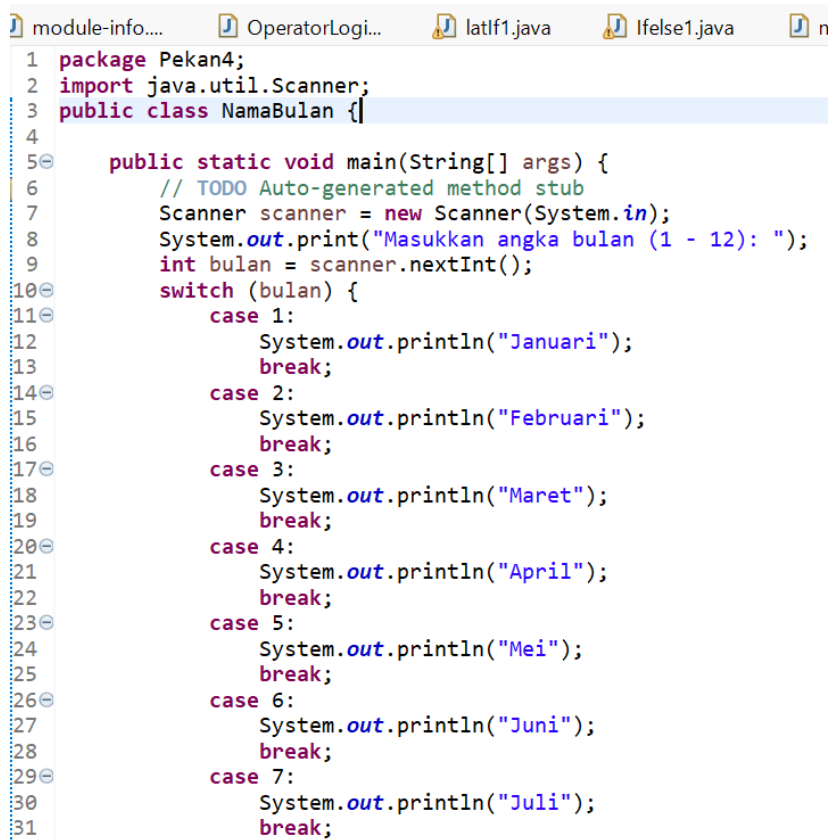
2. Percabangan

- Menggunakan *if-else if-else* untuk menentukan nilai huruf.
- $\geq 81 \rightarrow A, \geq 70 \rightarrow B, \geq 60 \rightarrow C, \geq 50 \rightarrow D$, selebihnya $\rightarrow E$.

3. Output

Program menampilkan nilai huruf sesuai angka yang dimasukkan.

2.1.5 NamaBulan



```

1 package Pekan4;
2 import java.util.Scanner;
3 public class NamaBulan {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         Scanner scanner = new Scanner(System.in);
8         System.out.print("Masukkan angka bulan (1 - 12): ");
9         int bulan = scanner.nextInt();
10        switch (bulan) {
11            case 1:
12                System.out.println("Januari");
13                break;
14            case 2:
15                System.out.println("Februari");
16                break;
17            case 3:
18                System.out.println("Maret");
19                break;
20            case 4:
21                System.out.println("April");
22                break;
23            case 5:
24                System.out.println("Mei");
25                break;
26            case 6:
27                System.out.println("Juni");
28                break;
29            case 7:
30                System.out.println("Juli");
31                break;

```

```

32 case 8:
33     System.out.println("Agustus");
34     break;
35 case 9:
36     System.out.println("September");
37     break;
38 case 10:
39     System.out.println("Oktober");
40     break;
41 case 11:
42     System.out.println("November");
43     break;
44 case 12:
45     System.out.println("Desember");
46     break;
47 default:
48     System.out.println("Angka tidak valid");
49 }
50 scanner.close();
51 }
52
53

```

Kode Program 2.5 (Kode Program NamaBulan)

Uraian Kode Program :

1. Input Data

- Program meminta pengguna memasukkan angka bulan (1–12).
- Angka yang dimasukkan disimpan pada variabel bulan.

2. Percabangan (*Switch-Case*)

- Program menggunakan *switch(bulan)* untuk mencocokkan angka dengan nama bulan.
- Setiap *case* mewakili satu bulan, misalnya *case 1* → Januari, *case 2* → Februari, dan seterusnya.

3. Output

- Program menampilkan nama bulan sesuai angka yang dimasukkan pengguna.

2.2 Langkah Kerja

2.2.1 LatIf1

1. Membuat kelas baru bernama *latIf1* dalam *package Pekan 4*.

Source folder: Browse...

Package: Browse...

☐ Enclosing type: Browse...

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass: Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?

☒ public static void main(String[] args)
☐ Constructors from superclass
☐ Inherited abstract methods

Gambar 2.1 (Langkah 1 LatIf1)

2. Program meminta pengguna memasukkan nilai IPK melalui keyboard.
 Nilai IPK tersebut disimpan dalam variabel IPK (*tipe double*).

```

5 public static void main(String[] args) {
6     // TODO Auto-generated method stub
7     double IPK;
8     Scanner input=new Scanner (System.in);
9     System.out.print("Input IPK Anda = ");
10    IPK=input.nextDouble();

```

Gambar 2.2 (Langkah 2 LatIf1)

3. Program memeriksa apakah nilai IPK > 2.75. Jika kondisi benar, maka perintah di dalam blok *if* akan dijalankan.

```

if (IPK>2.75) {

```


Gambar 2.3 (Langkah 3 LatIf1)

4. Jika syarat terpenuhi, program menampilkan pesan: "Anda Lulus Sangat Memuaskan dengan IPK <nilai>". Jika tidak, program tidak menampilkan apa pun.

```

12         System.out.println("Anda Lulus Sangat Memuaskan dengan IPK "+IPK);
13     }
14 }
15
16 }

```

Gambar 2.4 (Langkah 4 LatIf1)

2.2.2 Ifelse1

1. Membuat kelas baru bernama *Ifelse* dalam *package Pekan4*.

Source folder: Prakalpro_2025_B_2511531016/src Browse...

Package: Pekan4 Browse...

☐ Enclosing type: Pekan4.latIf1 Browse...

Name: Ifelse1

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?
☒ public static void main(String[] args)
☐ Constructors from superclass
☐ Inherited abstract methods

Gambar 2.5 (Langkah 1 Ifelse1)

2. Import library *Scanner* untuk membaca input. Buat objek *Scanner* input untuk mengambil data dari keyboard. Deklarasikan variabel *double* IPK. Program meminta pengguna memasukkan nilai IPK, lalu menyimpannya ke variabel IPK.

```
public static void main(String[] args) {
    // TODO Auto-generated method stub
    double IPK;
    Scanner input=new Scanner (System.in);
    System.out.print("Input IPK Anda = ");
    IPK=input.nextDouble();
}
```

Gambar 2.6 (Langkah 2 Ifelse1)

3. Program memeriksa kondisi *if* ($IPK > 2.75$). Jika kondisi salah, program mengeksekusi perintah di dalam blok *if*. Lalu menampilkan "*Anda Lulus Sangat Memuaskan dengan IPK <nilai>*".

```
11 if (IPK>2.75) {
12     System.out.println("Anda Lulus Sangat Memuaskan dengan IPK "+IPK);
}
```

Gambar 2.7 (Langkah 3 Ifelse1)

4. Program memeriksa kondisi *if* ($IPK > 2.75$). Jika kondisi benar, program mengeksekusi perintah di dalam blok *else*. Lalu menampilkan "*Anda tidak Lulus*".

```
13 } else {
14     System.out.println("Anda Tidak Lulus");
}
```

Gambar 2.8 (Langkah 4 Ifelse1)

2.2.3 MultiIf

1. Membuat kelas baru bernama *MultiIf* dalam *package Pekan4*.

Source folder: Prakalpro_2025_B_2511531016/src Browse...

Package: Pekan4 Browse...

☐ Enclosing type: Pekan4.latf1 Browse...

Name: multiIf

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?
☒ public static void main(String[] args)
☐ Constructors from superclass
☐ Inherited abstract methods

Gambar 2.9 (Langkah 1 MultiIf)

2. Import library Scanner. Deklarasi variabel `int umur` dan `char sim`. Buat objek Scanner `a` untuk membaca input dari keyboard. Program meminta input umur (`umur = a.nextInt()`) dan kepemilikan SIM C (`sim = a.next().charAt(0)`).

```

5 public static void main(String[] args) {
6     // TODO Auto-generated method stub
7     int umur;
8     char sim;
9     Scanner a= new Scanner(System.in);
10    System.out.print("Input umur anda: ");
11    umur= a.nextInt();
12    System.out.print("Apakah Anda Sudah Punya Sim C: ");
13    sim=a.next().charAt(0);
14    a.close();

```

Gambar 2.10 (Langkah 2 MultiIf)

3. Program melakukan proses pengecekan dengan percabangan *if* dan menampilkan hasil berupa pesan sesuai kondisi yang dipenuhi.

```

if((umur >= 17)&&(sim=='y')) {
    System.out.println("Anda Sudah dewasa dan boleh bawa motor");
}
if((umur >= 17)&&(sim!='y')) {
    System.out.println("Anda Sudah dewasa tetapi tidak boleh bawa motor");
}
if((umur < 17)&&(sim!='y')) {
    System.out.println("Anda Belum Cukup Umur bawa motor");
}
if((umur < 17)&&(sim=='y')) {
    System.out.println("Anda Belum Cukup Umur punya SIM");
}
}

```

Gambar 2.11 (Langkah 3 MultiIf)

2.2.4 Nilai

1. Membuat kelas baru bernama *Nilai* dalam *package Pekan4*.

Source folder: Prakalpro_2025_B_2511531016/src

Package: Pekan4

☐ Enclosing type: Pekan4.latIf1

Name: Nilai

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

☒ public static void main(String[] args)
☐ Constructors from superclass
☐ Inherited abstract methods

Gambar 2.12 (Langkah 1 Nilai)

2. Program diawali dengan mengimpor *library Scanner* dan mendeklarasikan variabel *nilai* untuk menyimpan input dari pengguna. Selanjutnya, program menampilkan pesan *"Inputkan nilai angka ="* dan menunggu pengguna memasukkan angka. Input ini disimpan dalam variabel *nilai*.

```

5 public static void main(String[] args) {
6     // TODO Auto-generated method stub
7     int nilai;
8     Scanner input=new Scanner(System.in);
9     System.out.print("Inputkan nilai angka= ");
0     nilai=input.nextInt();
1     input.close();

```

Gambar 2.13 (Langkah 2 Nilai)

3. Menginput dua nilai boolean (*true* atau *false*) dari keyboard menggunakan *Scanner*.

```

10 Scanner keyboard = new Scanner(System.in);
11 System.out.print("Input nilai boolean-1 (true / false) : ");
12 A1 = keyboard.nextBoolean();//input 10
13 System.out.print("Input nilai boolean-2 (true / false): ");
14 A2 = keyboard.nextBoolean();
15 keyboard.close();

```

Gambar 2.14 (Langkah 3 Nilai)

4. Setelah itu, program menggunakan struktur percabangan *if-else if-else* untuk mengecek nilai yang dimasukkan. Jika nilai ≥ 81 , maka program menampilkan "A". Jika tidak, tetapi nilai ≥ 70 , maka ditampilkan "B". Selanjutnya, jika nilai ≥ 60 , maka ditampilkan "C". Jika nilai ≥ 50 , maka ditampilkan "D".

```

13 if(nilai >=81) {
14     System.out.println("A");
15 } else if (nilai >=70) {
16     System.out.println("B");
17 } else if (nilai >=60) {
18     System.out.println("C");
19 } else if (nilai >=50) {
20     System.out.println("D");

```

Gambar 2.15 (Langkah 4 Nilai)

5. Apabila semua kondisi di atas tidak terpenuhi (nilai < 50), maka program menampilkan "E". Dengan demikian, program secara otomatis mengubah nilai angka yang dimasukkan pengguna menjadi huruf berdasarkan rentang yang telah ditentukan.

```
21 }else {
22     System.out.println("E");
```

Gambar 2.16 (Langkah 5 Nilai)

2.2.5 NamaBulan

1. Membuat kelas baru bernama *NamaBulan* dalam *package Pekan4*.

Source folder: Prakalpro_2025_B_2511531016/src Browse...

Package: Pekan4 Browse...

☐ Enclosing type: Pekan4.latIf1 Browse...

Name: NamaBulan

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?
☒ public static void main(String[] args)
☐ Constructors from superclass
☐ Inherited abstract methods

Gambar 2.17 (Langkah 1 NamaBulan)

2. Program mendeklarasikan *class NamaBulan*, mengimpor *Scanner*, lalu meminta input angka dari pengguna melalui perintah Masukkan angka bulan (1 - 12). Input ini disimpan dalam variabel bulan.

```

5 public static void main(String[] args) {
6     // TODO Auto-generated method stub
7     Scanner scanner = new Scanner(System.in);
8     System.out.print("Masukkan angka bulan (1 - 12): ");
9     int bulan = scanner.nextInt();

```

Gambar 2.18 (Langkah 2 NamaBulan)

3. Program menggunakan struktur *switch* untuk mengecek nilai variabel bulan. Jika pengguna memasukkan angka tertentu, program akan mencocokkannya dengan *case* yang sesuai. Misalnya, jika bulan = 1, maka program menampilkan "Januari", jika bulan = 2 menampilkan "Februari", dan seterusnya.

```

10 switch (bulan) {
11     case 1:
12         System.out.println("Januari");
13         break;
14     case 2:
15         System.out.println("Februari");
16         break;
17     case 3:
18         System.out.println("Maret");
19         break;
20     case 4:
21         System.out.println("April");
22         break;
23     case 5:
24         System.out.println("Mei");
25         break;
26     case 6:
27         System.out.println("Juni");
28         break;
29     case 7:
30         System.out.println("Juli");
31         break;
32     case 8:
33         System.out.println("Agustus");
34         break;
35     case 9:
36         System.out.println("September");
37         break;
38     case 10:
39         System.out.println("Oktober");
40         break;
41
42     case 11:
43         System.out.println("November");
44         break;
45     case 12:
46         System.out.println("Desember");
47         break;

```

Gambar 2.19 (Langkah 3 NamaBulan)

4. Default akan dijalankan jika tidak ada *case* yang cocok dengan nilai yang sedang diperiksa. `System.out.println("Angka tidak valid");` akan memberi tahu pengguna bahwa input yang dimasukkan tidak sesuai dengan pilihan yang diharapkan. Lalu tutup objek *Scanner*.

```

47     default:
48         System.out.println("Angka tidak valid");
49     }
50     scanner.close();
51 }

```

Gambar 2.20 (Langkah 4 NamaBulan)

2.3 Analisis Hasil

Analisis hasil dari kelima kode program menunjukkan bahwa konsep *conditional statement* atau pernyataan bersyarat merupakan salah satu elemen fundamental dalam pemrograman Java yang berfungsi untuk mengendalikan alur eksekusi berdasarkan kondisi tertentu.

Pada program *multiIf*, penggunaan *if-else* majemuk dengan operator logika & memperlihatkan bagaimana komputer mampu mengambil keputusan berdasarkan lebih dari satu kondisi, yakni usia dan kepemilikan SIM. Hal ini sesuai dengan teori yang dijelaskan oleh William Stallings (2012), bahwa kontrol alur bersyarat memungkinkan sistem untuk melakukan pemrosesan yang adaptif sesuai data masukan. Program Nilai menggunakan *if-else if ladder* untuk mengklasifikasikan nilai menjadi kategori A sampai E, yang menggambarkan penerapan pengambilan keputusan bertingkat. Menurut Deitel & Deitel (2015), struktur ini efektif digunakan ketika terdapat beberapa kondisi yang saling eksklusif.

Selanjutnya, program *latIf1* menampilkan penggunaan *if statement* sederhana untuk memeriksa apakah IPK mahasiswa lebih dari 2.75. Hal ini menegaskan pernyataan Barnes & Kolling (2012), bahwa *if statement* adalah

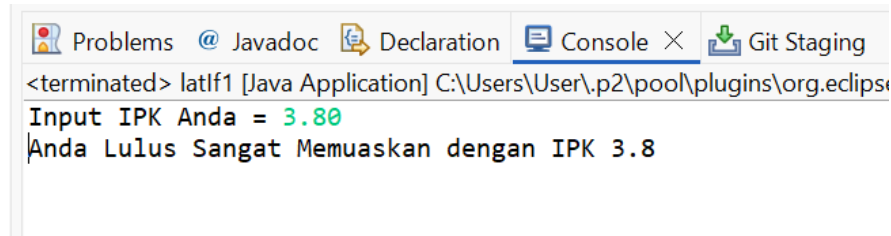
bentuk paling dasar dari seleksi logika yang digunakan untuk menentukan jalannya program. Program *Ifelse1* mengembangkan konsep tersebut dengan menambahkan *else* sehingga dapat menangani dua kemungkinan kondisi (lulus atau tidak lulus), yang sesuai dengan teori Savitch (2014) bahwa *if-else* diperlukan untuk memberikan alternatif jalur eksekusi. Program terakhir, *NamaBulan*, menggunakan *switch-case* untuk menentukan nama bulan berdasarkan input angka. Menurut Schildt (2018), *switch-case* sangat cocok digunakan pada kasus dengan banyak kemungkinan nilai diskrit karena lebih ringkas dan mudah dibaca dibandingkan *if-else ladder*.

Dari keseluruhan analisis, dapat disimpulkan bahwa kelima program berhasil menunjukkan perbedaan mendasar antara berbagai bentuk *conditional statement* di Java. Masing-masing memiliki fungsi spesifik: *if* untuk kondisi tunggal, *if-else* untuk dua kondisi, *if-else if ladder* untuk kondisi bertingkat, *multif* untuk kondisi kompleks dengan operator logika, serta *switch-case* untuk banyak pilihan diskrit. Teori dan referensi yang digunakan mendukung bahwa pemahaman dan penerapan struktur kontrol ini merupakan langkah penting dalam penguasaan logika pemrograman yang efisien dan terstruktur.

BAB III KESIMPULAN

3.1 Hasil Praktikum

3.1.1 LatIf1

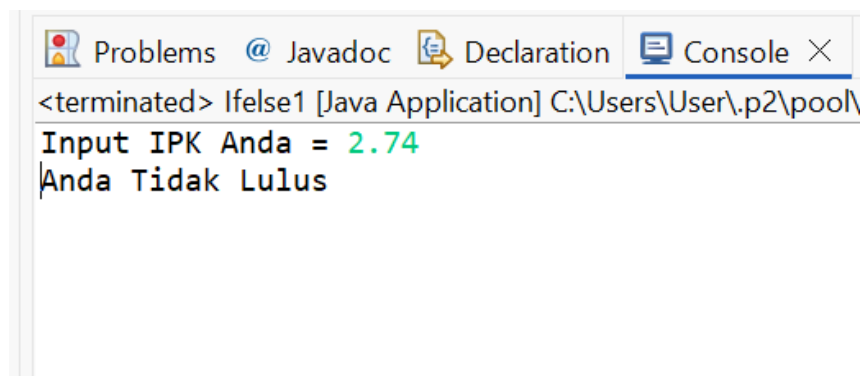


```
<terminated> latIf1 [Java Application] C:\Users\User\.p2\pool\plugins\org.eclipse...
Input IPK Anda = 3.80
Anda Lulus Sangat Memuaskan dengan IPK 3.8
```

Gambar 3.1 (Hasil Praktikum LatIf1)

Pada praktikum tersebut, pengguna memasukkan nilai IPK sebesar 3.80. Program kemudian menampilkan kembali input tersebut sebagai konfirmasi dan melakukan pengecekan terhadap kategori IPK. Hasil akhirnya ditampilkan dalam bentuk pesan di layar: *“Anda Lulus Sangat Memuaskan dengan IPK 3.8”*. Hal ini menunjukkan bahwa program berhasil membaca input, memproses kondisi, dan menampilkan hasil sesuai logika yang telah ditentukan.

3.1.2 Ifelse1

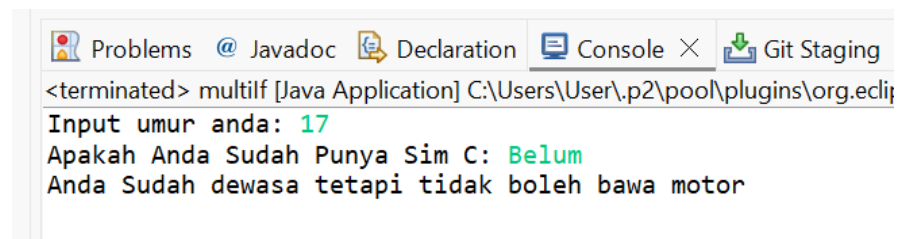


```
<terminated> Ifelse1 [Java Application] C:\Users\User\.p2\pool\...
Input IPK Anda = 2.74
Anda Tidak Lulus
```

Gambar 3.2 (Hasil Praktikum Ifelse1)

Pada praktikum tersebut, pengguna memasukkan nilai IPK sebesar 2.74. Program kemudian menampilkan kembali input tersebut sebagai konfirmasi dan melakukan pengecekan terhadap kategori kelulusan. Karena nilai 2.74 berada di bawah batas minimal kelulusan yang ditentukan, program memberikan output berupa pesan “*Anda Tidak Lulus*”. Hal ini menunjukkan bahwa program berhasil menjalankan logika kondisi *if-else* untuk menentukan status kelulusan berdasarkan nilai IPK yang diinputkan.

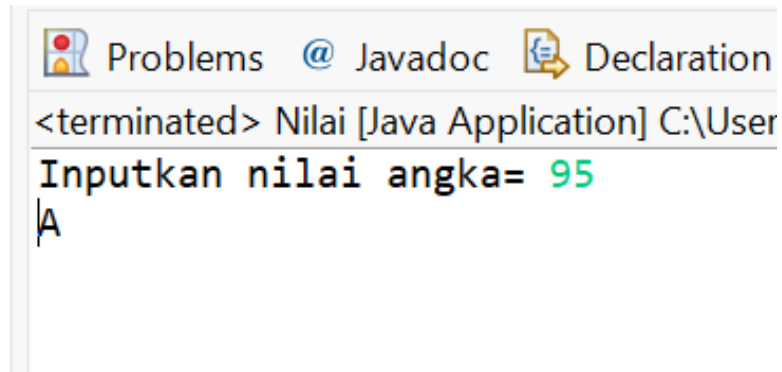
3.1.3 MultIf



Gambar 3.3 (Hasil Praktikum MultIf)

Program menerima input umur dan status kepemilikan SIM C. Jika umur sudah mencapai atau melewati batas dewasa (misalnya 17 tahun), tapi belum memiliki SIM C, maka program akan menampilkan pesan bahwa meskipun sudah dewasa, pengendara tidak boleh membawa motor tanpa SIM C.

3.1.4 Nilai



```

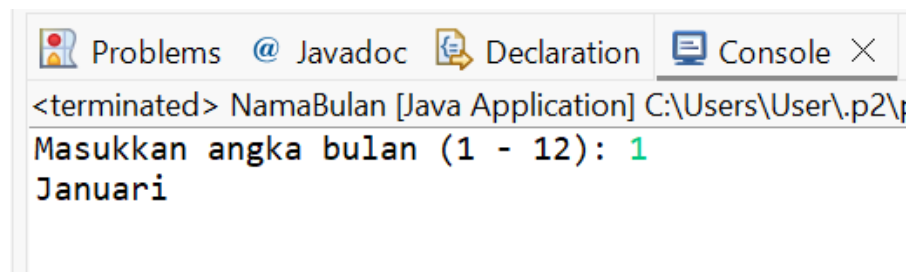
Problems @ Javadoc Declaration
<terminated> Nilai [Java Application] C:\User
Inputkan nilai angka= 95
A

```

Gambar 3.4 (Hasil Praktikum Nilai)

Hasil praktikum menunjukkan bahwa ketika input nilai angka adalah 95, program mengeluarkan hasil "A". Ini berarti nilai 95 masuk ke dalam kategori nilai terbaik atau sangat baik dalam penilaian yang diberikan oleh program.

3.1.5 NamaBulan



```

Problems @ Javadoc Declaration Console X
<terminated> NamaBulan [Java Application] C:\Users\User\p2\
Masukkan angka bulan (1 - 12): 1
Januari

```

Gambar 3.5 (Hasil Praktikum NamaBulan)

Hasil praktikum menunjukkan bahwa ketika pengguna memasukkan angka bulan 1, program memberikan output "Januari" yang merupakan nama bulan ke-1 dalam kalender. Ini berarti program berhasil mengonversi input angka bulan menjadi nama bulan yang sesuai.

3.2 Saran Pengembangan

Saran pengembangan dari kelima kode program di atas dapat diarahkan pada peningkatan fungsionalitas, fleksibilitas, dan interaktivitas agar lebih bermanfaat dalam praktik nyata. Program *multiIf* dapat dikembangkan dengan menambahkan validasi input serta kondisi tambahan, misalnya pemeriksaan kesehatan atau pelatihan mengemudi, sehingga lebih realistis. Program Nilai bisa diperluas dengan fitur input data beberapa mahasiswa sekaligus lalu menampilkan rekap nilai dalam bentuk tabel, bahkan disertai grafik sederhana untuk analisis visual.

Untuk program *latIf1* dan *Ifelse1*, pengembangan yang bisa dilakukan adalah dengan menambahkan kategori penilaian lebih detail, misalnya predikat “memuaskan” atau “*cumlaude*” berdasarkan IPK, serta menambahkan fitur penyimpanan hasil ke dalam file. Sementara itu, program NamaBulan dapat ditingkatkan dengan menambahkan informasi tambahan seperti jumlah hari dalam bulan atau penanda tahun kabisat khusus untuk bulan Februari.

Selain itu, semua program dapat digabungkan dengan antarmuka grafis sederhana menggunakan Java Swing atau JavaFX agar lebih menarik dan mudah digunakan. Dengan pengembangan ini, kelima program tidak hanya berfungsi sebagai latihan dasar *conditional statement*, tetapi juga memiliki aplikasi nyata yang lebih luas dan mendukung pemahaman mendalam tentang logika pemrograman.

DAFTAR PUSTAKA

- [1] W. Stallings, *Operating Systems: Internals and Design Principles*, 7th ed. Upper Saddle River, NJ: Prentice Hall, 2012.
- [2] P. J. Deitel and H. M. Deitel, *Java How to Program*, 10th ed. Upper Saddle River, NJ: Pearson, 2015.
- [3] D. J. Barnes and M. Kölling, *Objects First with Java: A Practical Introduction Using BlueJ*, 5th ed. Harlow: Pearson Education, 2012.
- [4] W. J. Savitch, *Absolute Java*, 5th ed. Boston: Addison-Wesley, 2014.
- [5] H. Schildt, *Java: The Complete Reference*, 9th ed. New York: McGraw-Hill, 2018.