

LAPORAN PRAKTIKUM
PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN
“STATEMENT CONDISIONAL DI JAVA”



disusun Oleh:

ARDRA ATHAILLA FIRSY

2511533028

Dosen Pengampu:

Dr. WAHYUDI, S.T, M.T.

Asisten Praktikum:

JOVANTRI IMMANUEL GULO

DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya, laporan praktikum dengan judul “STATEMENT CONDISIONAL DI JAVA” dapat diselesaikan tepat waktu.

Laporan ini disusun untuk memenuhi salah satu tugas mata kuliah Praktikum Algoritma dan Pemrograman, sekaligus sebagai sarana pembelajaran dalam memahami konsep dasar pemrograman khususnya statement condisional pada Bahasa Java.

Pada kesempatan ini, penulis menyampaikan terimakasih kepada:

1. Bapak Dr. Wahyudi, S.T., M.T. selaku dosen pengampu mata kuliah Algoritma dan pemrograman yang telah memberikan bimbingan dan arahan.
2. Bang Jovantri Immanuel Gulo selaku asisten praktikum kelas A yang telah membantu pelaksanaan praktikum.
3. Teman-teman mahasiswa yang telah membantu dan memberi dukungan serta berdiskusi Bersama dalam menyelesaikan laporan ini.

Penulis menyadari bahwa laporan ini masih jauh dari kata sempurna. Oleh karena itu, kirtik dan saran yang membangun sangat diharapkan demi penyempurnaan laporan di masa mendatang.

Akhir kata, semoga laporan ini dapat memberikan manfaat bagi pembaca dan menambah wawasan mengenai pemrograman Java.

DAFTAR ISI

KATA PENGANTAR.....	2
DAFTAR ISI	3
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Praktikum	1
1.3 Manfaat Praktikum	1
BAB II PEMBAHASAN	2
2.1 Langkah Kerja Praktikum	2
2.2 Analisis Hasil dan Pembahasan.....	7
BAB III PENUTUP	10
3.1 Kesimpulan	10

DAFTAR

PUSTAKA

.....

BAB I PENDAHULUAN

1.1 Latar Belakang

Pemrograman merupakan salah satu dasar penting dalam bidang informatika. Salah satu konsep utama dalam pemrograman adalah struktur kontrol yang digunakan untuk mengatur kondisi tertentu. Struktur kontrol seperti if, if-else, if-else-if, dan switch-case sangat dibutuhkan agar program dapat mengambil Keputusan yang berbeda sesuai dengan input pengguna.

Melalui praktikum ini, mahasiswa diajak untuk memahami serta menerapkan penggunaan struktur kontrol dalam menyelesaikan permasalahan sederhana, misalnya menentukan kelulusan berdasarkan IPK, menilai prestasi berdasarkan angka, menampilkan nama bulan sesuai input angka, serta memeriksa syarat berkendara berdasarkan umur dan kepemilikan SIM.

1.2 Tujuan Praktikum

1. Memahami konsep dasar percabangan (*decision making*) dalam bahasa Java.
2. Mampu mengimplementasikan if, if-else, if-else-if, dan switch-case pada program sederhana.
3. Melatih logika pemrograman dalam menyelesaikan masalah berdasarkan kondisi bervariasi.
4. Membiasakan diri menggunakan *scanner* untuk menerima input dari pengguna pada program Java.

1.3 Manfaat Praktikum

1. Mahasiswa dapat berpikir logis dalam Menyusun alur program yang melibatkan pengambilan Keputusan.
2. Mahasiswa terbiasa menggunakan struktur kontrol untuk membuat program lebih dinamis dan interaktif.
3. Menjadi bekal dasar untuk mengerjakan pemrograman yang lebih kompleks di pertemuan berikutnya.
4. Memberikan pemahaman bahwa logika kondisi dalam program dapat diaplikasikan pada permasalahan sehari-hari.

BAB II PEMBAHASAN

2.1 Langkah Kerja Praktikum

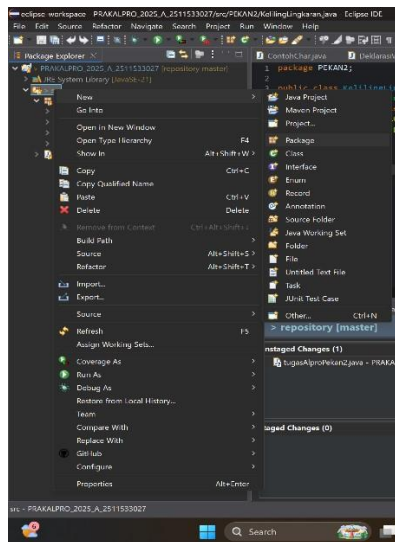
1. Persiapan Awal

- a) Buka aplikasi Eclipse IDE di laptop atau di komputer.



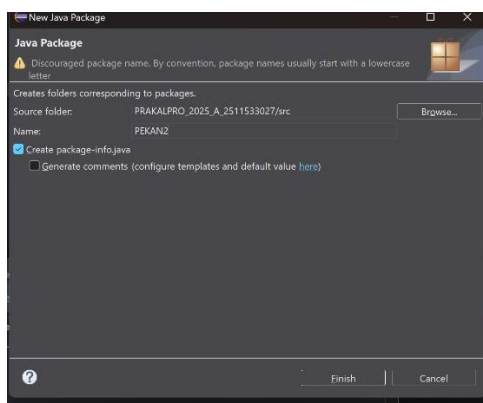
2. Membuat Package

- a) Klik kanan pada folder **src**, lalu pilih **New**, terakhir klik **Package**.



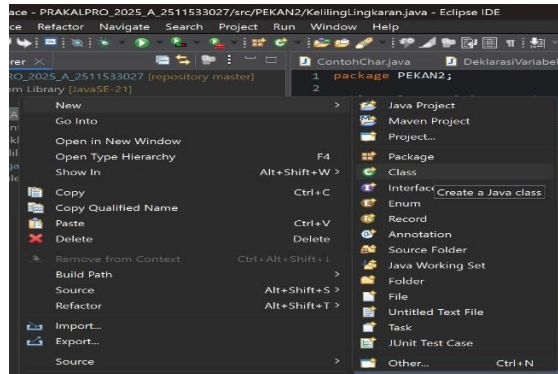
b)

Beri nama package **PEKAN4**.



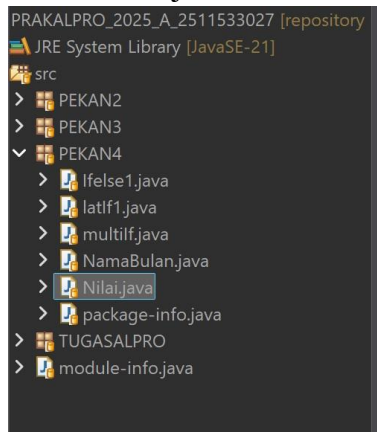
3. Membuat Class Baru untuk Setiap Percobaan

- a) Klik kanan pada package **PEKAN4**, lalu pilih **New**, terakhir klik **Class**.



b) Buat beberapa class sesuai percobaan.

- Ifelse1.java
- latIf1.java
- multiIf.java
- NamaBulan.java
- Nilai.java



c) Saat membuat class, centang opsi **public static void main(String[] args)** agar otomatis ada fungsi **main**.

4. Menulis Kode Program

a) Ifelse1.java

- Import library java.util.Scanner untuk membaca input dari pengguna.
- Deklarasikan variabel IPK bertipe double.
- Buat objek Scanner untuk menerima input.
- Tampilkan pesan: “Input IPK Anda = “ pada layar.
- Terima input berupa bilangan decimal dengan input.nextDouble().
- Tutup objek Scanner dengan input.close().
- Gunakan struktur if-else:
 - Jika IPK >= 2,75 tampilkan pesan “Anda Lulus Sangat Memuaskan dengan IPK...”.

- Jika tidak, tampilkan pesan “Anda Tidak Lulus”.

```

1 package PEKAN4;
2
3 import java.util.Scanner;
4
5 public class IfElse1 {
6
7     public static void main(String[] args) {
8         double IPK;
9         Scanner input=new Scanner (System.in);
10        System.out.print("Input IPK Anda = ");
11        IPK=input.nextDouble();
12        input.close();
13        if (IPK>=2.75) {
14            System.out.println("Anda Lulus Sangat Memuaskan dengan IPK "+IPK);
15        } else {
16            System.out.println("Anda Tidak Lulus");
17        }
18    }
19 }
20
21 }
22

```

b) latIf1.java

- Import java.util.Scanner.
- Deklarasikan variabel IPK bertipe double.
- Buat objek Scanner untuk input.
- Minta pengguna memasukkan nilai IPK dengan perintah System.out.print.
- Ambil input berupa angka decimal menggunakan nextDouble().
- Tutup objek Scanner.
- Gunakan struktur if (tanpa else):
 - Jika IPK >= 2,75 tampilkan “Anda Lulus Sangat Memuaskan dengan IPK...”.
 - Jika tidak memenuhi syarat program tidak menampilkan apa-apa.

```

1 package PEKAN4;
2
3 import java.util.Scanner;
4
5 public class latIf1 {
6
7     public static void main(String[] args) {
8         double IPK;
9         Scanner input=new Scanner (System.in);
10        System.out.print("Input IPK Anda = ");
11        IPK=input.nextDouble();
12        input.close();
13        if (IPK>=2.75) {
14            System.out.println("Anda Lulus Sangat Memuaskan dengan IPK "+IPK);
15        }
16    }
17 }
18
19 }

```

c) multiIf.java

- Import java.util.Scanner.
- Deklarasikan variabel umur bertipe int dan sim bertipe char.
- Buat objek Scanner.
- Minta pengguna memasukkan umur nextInt().
- Minta input apakah sudah punya SIM dengan jawaban ‘y’ untuk ya.
- Tutup objek Scanner.

- Gunakan beberapa kondisi if:
 - Jika umur >= 17 dan sim=='y' □ tampilkan “Anda sudah dewasa dan boleh bawa motor”.
 - Jika umur >= 17 dan sim!='y' □ tampilkan “Anda sudah dewasa tetapi tidak boleh bawa motor”.
 - Jika umur < 17 dan sim!='y' □ tampilkan “Anda belum cukup umur bawa motor”.
 - Jika umur < 17 dan sim=='y' □ tampilkan “Anda belum cukup umur punya SIM”.

```

1 package PEKAN4;
2
3 import java.util.Scanner;
4
5 public class multiIf {
6     public static void main(String[] args) {
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
30

```

d) NamaBulan.java

- Import java.util.Scanner.
- Deklarasikan variabel bulan bertipe int.
- Buat objek Scanner.
- Tampilkan pesan “Masukkan angka bulan (1-12):”.
- Terima input angka 1-12 dengan nextInt().
- Gunakan struktur switch-case:
 - Jika input 1 □ tampilkan “Januari”. ○ Jika input 2 □ tampilkan “Februari”.
 - ... sampai angka 12 □ “Desember”.
- Tambahkan break; pada setiap case agar tidak lanjut ke case berikutnya.

```

1 package PEKAN4;
2
3 import java.util.Scanner;
4
5 public class NamaBulan {
6
7     public static void main(String[] args) {
8         Scanner scanner = new Scanner(System.in);
9         System.out.print("Masukkan angka bulan (1 - 12): ");
10        int bulan = scanner.nextInt();
11        switch (bulan) {
12            case 1:
13                System.out.println("Januari");
14                break;
15            case 2:
16                System.out.println("Februari");
17                break;
18            case 3:
19                System.out.println("Maret");
20                break;
21            case 4:
22                System.out.println("April");
23                break;
24            case 5:
25                System.out.println("Mei");
26                break;
27            case 6:
28                System.out.println("Juni");
29                break;
30            case 7:
31                System.out.println("Juli");

```

e) Nilai.java

- Import java.util.Scanner.
- Deklarasikan variabel nilai bertipe int.
- Buat objek Scanner.
- Tampilkan pesan "Input nilai Anda:".
- Terima input angka depan nextInt().
- Tutup objek Scanner.
- Gunakan if-else-if untuk menentukan grade:
 - Jika nilai >= 81 □ tampilkan 'A'.
 - Jika nilai >= 70 □ tampilkan 'B'.
 - Jika nilai >= 60 □ tampilkan 'C'.
 - Jika nilai >= 50 □ tampilkan 'D'.
 - Jika kurang dari 50 □ tampilkan 'E'.

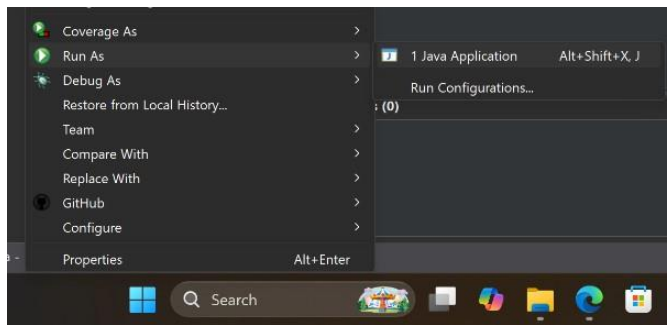
```

1 package PEKAN4;
2
3 import java.util.Scanner;
4
5 public class Nilai {
6     public static void main(String[] args) {
7         int nilai;
8         Scanner input = new Scanner(System.in);
9         nilai = input.nextInt();
10        input.close();
11
12        if (nilai >= 81) {
13            System.out.println("A");
14        } else if (nilai >= 70) {
15            System.out.println("B");
16        } else if (nilai >= 60) {
17            System.out.println("C");
18        } else if (nilai >= 50) {
19            System.out.println("D");
20        } else {
21            System.out.println("E");
22        }
23    }
24 }
25
26

```

5. Kompilasi dan Menjalankan Program

- Klik kanan file program, lalu klik **Run As**, lalu klik **Java Application**.



- b) Amati hasil output yang ditampilkan di tab **Console**.

2.2 Analisis Hasil dan Pembahasan

1. Program Ifelse1.java

- a) Hasil: Program meminta input IPK dari pengguna, lalu melakukan pengecekan dengan struktur if-else. Jika nilai $IPK \geq 2.75$, maka ditampilkan output “Anda Lulus Sangat Memuaskan dengan IPK ...”. Jika kurang dari 2.75, output menjadi “Anda Tidak Lulus”.
- b) Analisis: Struktur if-else digunakan untuk pengambilan keputusan berdasarkan kondisi logis. Variabel IPK bertipe double sehingga dapat menyimpan angka desimal. Logika program ini sederhana: hanya ada dua kemungkinan hasil, yaitu lulus atau tidak lulus. Program ini bisa diperluas misalnya dengan menambahkan kategori predikat (memuaskan, cumlaude, dsb).
- c) Teori Pendukung: Menurut dokumentasi resmi Java, pernyataan if-else digunakan untuk percabangan (branching) dalam alur program. Kondisi dievaluasi sebagai boolean (true atau false). Jika bernilai true, blok if dijalankan, jika tidak maka blok else dijalankan.

2. Program latIf1.java

- a) Hasil: Program meminta input IPK, kemudian hanya memeriksa kondisi if ($IPK \geq 2.75$). Jika benar, ditampilkan “Anda Lulus Sangat Memuaskan dengan IPK ...”. Namun, jika salah, tidak ada output tambahan (karena tidak ada else).
- b) Analisis: Program ini lebih sederhana dari IfElse1.java. Struktur if tunggal digunakan, sehingga hanya memberikan respon saat kondisi terpenuhi. Hal ini menunjukkan perbedaan penggunaan if dengan if-else.
- c) Teori Pendukung: Pernyataan if di Java memungkinkan eksekusi bersyarat berdasarkan ekspresi boolean. Jika kondisi bernilai false, program akan melanjutkan eksekusi ke baris berikutnya tanpa melakukan aksi apapun.

3. multiIf.java

- a) Hasil: Program meminta input umur (integer) dan kepemilikan SIM (karakter 'y'/'t'). Kemudian dilakukan pengecekan dengan beberapa kondisi if untuk menentukan apakah seseorang boleh membawa motor atau tidak. Hasilnya bisa bervariasi, misalnya:
- $\text{Umur} \geq 17$ dan punya SIM \rightarrow **“Anda sudah dewasa dan boleh bawa motor”**.
 - $\text{Umur} \geq 17$ tetapi tidak punya SIM \rightarrow **“Anda sudah dewasa tetapi tidak boleh bawa motor”**.
 - $\text{Umur} < 17 \rightarrow$ output berbeda sesuai kondisi SIM.
- b) Analisis: memungkinkan lebih banyak kombinasi hasil. Struktur seperti ini sering digunakan saat terdapat lebih dari dua kondisi yang harus diuji secara independen.
- c) Teori Pendukung: Dalam Java, kita dapat menggunakan banyak if untuk memeriksa beberapa syarat sekaligus. Operator logika seperti $\&\&$ (AND) digunakan untuk menggabungkan lebih dari satu kondisi sebelum keputusan dibuat.

4. NamaBulan.java

- a) Hasil: Program meminta input angka bulan (1–12), lalu menggunakan struktur switch-case untuk menampilkan nama bulan yang sesuai. Misalnya input 10 menghasilkan output **“Oktober”**.
- b) Analisis: Struktur switch-case digunakan agar lebih rapi dibanding if-else berantai ketika menghadapi banyak pilihan (multi-branching). Setiap case mewakili satu kemungkinan nilai, dan break digunakan untuk menghentikan eksekusi setelah menemukan kecocokan.
- c) Teori Pendukung:
Menurut dokumentasi Java, switch-case efektif digunakan untuk kondisi diskrit dengan banyak pilihan, karena lebih mudah dibaca dan dieksekusi lebih efisien dibanding if-else panjang.

5. Nilai.java

- a) Hasil: Program meminta input nilai angka, kemudian menentukan predikat berdasarkan interval nilai menggunakan if-else if. Misalnya:
- $\geq 81 \rightarrow \mathbf{A}$
 - $\geq 70 \rightarrow \mathbf{B}$
 - $\geq 60 \rightarrow \mathbf{C}$
 - $\geq 50 \rightarrow \mathbf{D}$
 - $< 50 \rightarrow \mathbf{E}$

- b) Analisis: Program ini menunjukkan penggunaan if-else if untuk seleksi bertingkat, di mana setiap kondisi diperiksa berurutan hingga menemukan yang sesuai. Struktur ini cocok untuk klasifikasi atau pemberian kategori.
- c) Teori Pendukung: if-else if di Java memungkinkan program memilih satu dari beberapa blok kode berdasarkan kondisi tertentu. Urutan kondisi sangat penting: kondisi paling spesifik sebaiknya diletakkan di awal untuk menghindari kesalahan logika.

BAB III PENUTUP

3.1 Kesimpulan

Berdasarkan rangkaian praktikum yang telah dilakukan, dapat disimpulkan bahwa penggunaan struktur kontrol **if**, **if-else**, **multi-if**, **switch-case**, dan **if-else if** dalam Java sangat penting untuk memahami dasar pengambilan keputusan dalam pemrograman.

1. **Ifelse1.java** memperlihatkan perbedaan antara kondisi lulus dan tidak lulus dengan dua kemungkinan hasil menggunakan struktur *if-else*.
2. **latIf1.java** menekankan perbedaan *if tunggal*, yaitu hanya mengeksekusi perintah jika kondisi benar tanpa memberikan alternatif jika salah.
3. **multif.java** menunjukkan bagaimana *if* dapat digunakan berulang untuk menangani lebih banyak kondisi sekaligus, termasuk penggunaan operator logika untuk menggabungkan syarat.
4. **NamaBulan.java** membuktikan bahwa *switch-case* lebih efisien dan rapi dibanding *if-else berantai* ketika harus menangani banyak pilihan yang bersifat diskrit.
5. **Nilai.java** memberikan contoh nyata penerapan *if-else if* dalam klasifikasi nilai, di mana kondisi diuji secara bertahap hingga sesuai dengan kategori yang ditentukan.

Secara keseluruhan, praktikum ini menegaskan bahwa **pemahaman struktur kontrol percabangan** merupakan dasar penting dalam pemrograman Java. Setiap bentuk percabangan memiliki kegunaan spesifik:

- *if* → untuk kondisi sederhana,
- *if-else* → untuk dua kemungkinan hasil,
- *multi-if* → untuk kombinasi kondisi,
- *switch-case* → untuk banyak pilihan yang jelas,
- *if-else if* → untuk klasifikasi atau kategori bertingkat.

Dengan menguasai berbagai bentuk percabangan ini, mahasiswa dapat menyusun program yang lebih terstruktur, efisien, dan sesuai dengan kebutuhan logika pemecahan masalah dalam dunia nyata.

DAFTAR PUSTAKA

- [1] Wahyudi, *Statement Condisional di Java*, presentasi kuliah, Program Studi Informatika, Universitas Andalas, 2025.
- [2] Oracle, *The Java™ Tutorials – The if-then and if-then-else Statement*. Oracle. [Online]. Available:
<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/if.html>. [Accessed: 02-Oct-2025].
- [3] Oracle, *The Java™ Tutorials – switch Statements*. Oracle. [Online]. Available:
<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/switch.html>. [Accessed: 02-Oct2025].
- [4] Oracle, *Java SE Documentation*. Oracle. [Online]. Available:
<https://docs.oracle.com/en/java/javase/>. [Accessed: 02-Oct-2025].
- [5] H.M. Deitel and P.J. Deitel, *Java How to Program, 11th Edition*. Pearson, 2018.