**LAPORAN PRAKTIKUM ALGORITMA DAN PEMROGAMAN STRUKTUR CONTROL PERCABANGAN**

DI SUSUN OLEH :

DIGO YUANDRA

NIM 2511533017

DOSEN PENGAMPU : Dr.WAHYUDI, S.T, M.T

ASISTEN LABORATORIUM: JOVANTRI IMMANUEL GULO



**DEPARTEMEN INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS ANDALAS**

**PADANG, 2025**

# **KATA PENGANTAR**

Sebelumnya saya Panjatkan Puji syukur atas kehadiran Tuhan Yang Maha Esa, berkat rahmat dan izinnya juga laporan praktikum “struktur control percabangan ini dapat diselesaikan dengan baik.Saya ucapkan terima kasih yang sebesar besar nya kepada Dr.Wahyudi, S.T, M.T selaku dosen pengampu yang telah membimbing Mata kuliah Algoritma dan Pemrograman. Dan tidak lupa saya ucapkan terima kasih kepada Uda Aufan Tafiqurrahman selaku asisten labor yang telah membimbing praktikum struktur control percabangan.

Laporan ini disusun sebagai salah satu tugas dalam mata kuliah Pemrograman Dasar Java dengan topik " Struktur Control Percabangan ".

Semoga laporan ini dapat bermanfaat dan memenuhi tujuan pembelajaran. Penulis menyadari bahwa laporan ini masih jauh dari sempurna, sehingga kritik dan saran yang membangun sangat diharapkan.

Padang , 30 september 2025

Penulis

i

# **DAFTAR**

[**KATA PENGANTAR** **i**](#_Toc9857)

[**DAFTAR ISI** **ii**](#_Toc9858)

[**BAB I PENDAHULUAN** **3**](#_Toc9859)

[1.1 Latar Belakang **3**](#_Toc9860)

[1.2 Tujuan **3**](#_Toc9861)

[1.3 Manfaat **3**](#_Toc9862)

[**BAB II PEMBAHASAN** **4**](#_Toc9863)

[2.1 Struktur kontrol percabangan **4**](#_Toc9864)

[2.2 Langkah Langkah pengerjaan **4**](#_Toc9865)

[**BAB III KESIMPULAN** **10**](#_Toc9866)

[3.1 Kesimpulan **10**](#_Toc9867)

[3.2 Saran **10**](#_Toc9868)

[**DAFTAR PUSTAKA** **11**](#_Toc9869)

ii

# **BAB I PENDAHULUAN**

# 1.1 Latar Belakang

Praktikum struktur control percabangan dalam java merupakan kegiatan pembelajaran yang bertujuan untuk memahami dan menerapkan berbagai cara untuk mengendalikan alur eksekusi program yang tersedia dalam bahasa pemrograman Java. Struktur control adalah pernyataan khusus yang menentukan urutan pernyataan mana yang akan di eksekusi. Percabangan, khususnya, seperti pernyataan *if, if-else* dan *switch,* memungkinkan program untuk membuat keputusan dan memilih blok kode mana yang akan dijalankan berdasarkan kondisi atau nilai tertentu.

# 1.2 Tujuan

1. Memahami dan menerapkan pernyataan percabangan (*if, if-else,* dan *switch).*

1. Menganalisis pengaruh ekspresi *Boolean* terhadap alur eksekusi program.

1. Mengembangkan logika program menggunakan struktur control untuk pengambilan keputusan.

1. Melatih *debugging* dan verifikasi hasil percabangan pada program bercabang.

# 1.3 Manfaat

1. Menguasai penggunaan percabangan (*if, switch*) untuk alur program.

1. Meningkatkan kemampuan *decision-making* dalam kode.

1. Memperkuat logika pemrograman untuk kondisi *Boolean.*

1. Melatih analisis dan *debugging .*

# **BAB II PEMBAHASAN**

# 2.1 Struktur Control Percabangan

Struktur Control Percabangan dalam Java adalah mekanisme dalam pemrograman yang memungkinkan program untuk membuat Keputusan dan memilih jalur eksekusi yang berbeda berdasarkan kondisi atau syarat tertentu.

1. Struktur *if-else*

Ini adalah struktur paling sederhana, digunakan untuk mengeksekusi blok perintah tertentu jika suatu kondisi terpengaruhi (true)

* + *if* : Jika kondisi benar, blok perintah di dalam *If* akan dieksekusi  *else*: Jika kondisi *if* tidak terpenuhi (false), blok perintah di dalam *else* akan dieksekusi
  + *if*-*else if* : Memperluas struktur *if-else* untuk menangani banyak kondisi secara bertingkat, memungkinkan eksekusi perintah khusus jika kondisi tertentu terpenuhi

1. Struktur *switch-case*

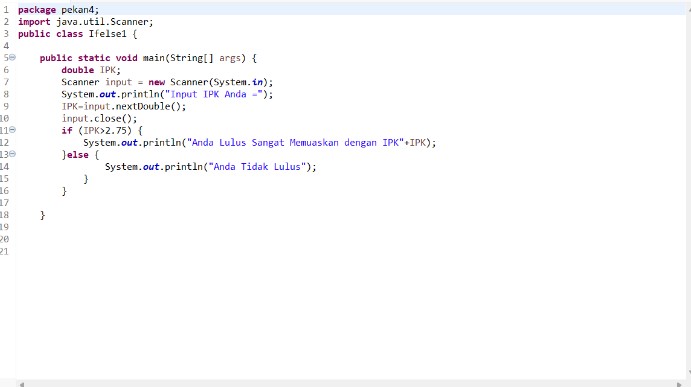
Struktur ini berguna untuk menangani banyak pilihan atau kondisi yang berbeda dengan cara yang efisien, terutama ketika ada beberapa nilai untuk sebuah variable

* + *switch* : Memilih satu dari beberapa kasus berdasarkan nilai suatu variabel
  + *case* : Menentukan kondisi spesifik yang akan dieksekusi jika nilai variabel cocok dengan nilai yang ditentukan dalam kasus tersebut
  + *break* : Menghentikan eksekusi setelah menemukan kasus yang sesuai dan mencegah program eksekusi kasus selanjutnya
  + *default* : Pilihan untuk menjalankan blok perintah jika ada nilai yang cocok dalam setiap kasus yang diberikan

# 2.2 Langkah Langkah pengerjaan

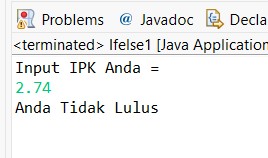
1. *If else*
   * Buat file *Java* bernama Ifelse1 di dalam package pekan4
   * Import library *java.util.Scanner* untuk memungkinkan program menerima input dari pengguna
   * Deklarasi kelas public ifelse1 yang berisi metode utama *public static void(String[] args)*
   * Di dalam metode *main,* deklarasikan variabel IPK bertipe *double* untuk menyimpan nilai indeks prestasi kumulatif
   * Buat objek *Scanner* Bernama *input* untuk membaca data dari keyboard
   * Tampilkan pesan “*Input IPK Anda=”* ke layar agar pengguna tahu harus memasukkan nilai IPK
   * Baca nilai yang dimasukkan pengguna menggunakan *input.nextDouble()* dan simpan ke variabel IPK
   * Tutup objek *Scanner* dengan perintah *input.close()* untuk membebaskan sumber daya system
   * Terapkan struktur percabangan *if-else:* 
     + Jika IPK > 2.75, tampilkan pesan *“Anda Lulus Sangat Memuaskan dengan IPK [nilai IPK]*
     + Jika tidak memengaruhi syarat (IPK ≤ 2.75), tampilkan pesan “*anda Tidak Lulus”*

 Jalankan program, masukkan nilai IPK saat diminta , lalu amati output yang sesuai dengan kondisi percabangan



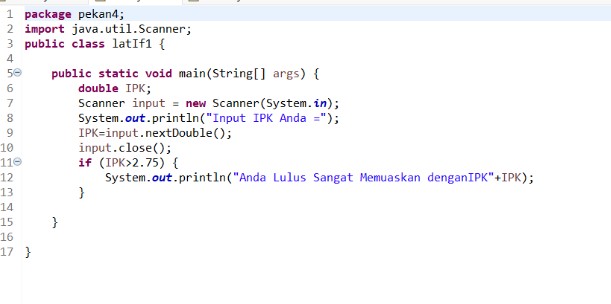
Gambar 2.1

Output yang dihasilkan



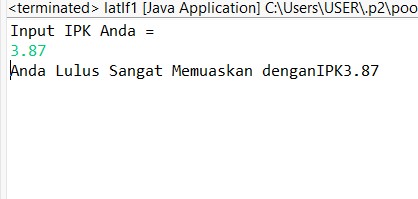
Gambar 2.2

1. *If*
   * + Buat file Intiíl java di package pekat.
     + Impor java.util.Scanner
     + Buat kelas latifl dengan metode main
     + Deklarasikan variabel double IPK
     + Buat objek Scanner untuk input pengguna.
     + Tampilkan pesan "Input IPK Anda dan baca nilainya ke variabel IPK.
     + Tutup Scanner.
     + Gunakan percabangan if jika IPK 2.75, tampilkan pesan kelulusan.
     + Jalankan program dan uji dengan nilai IPK.



Gambar 2.3

Output yang dihasilkan



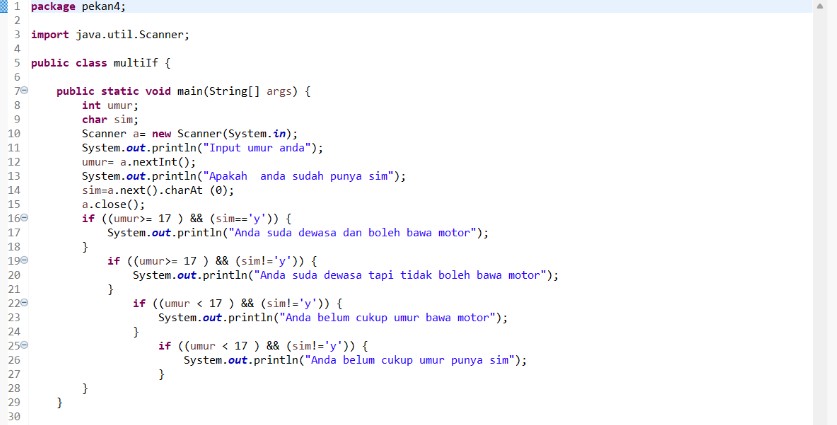
Gambar 2.4

1. *Multi if*

* + - Buat file Java bernama multiIf di package pekan4
    - Impor library java.util.Scanner untuk membaca input pengguna
    - Deklarasikan kelas publik mitilf dengan metode main
    - Deklarasikan variabel int umur dan char sim untuk menyimpan data input.
    - Buat objek Scanner bernama a untuk membaca input dari keyboard.
    - Tampilkan pesan Input umur anda lalu baca nilai integer ke variabel umur  Tampilkan pesan "Apakah anda sudah punya sim lalu baca karakter pertama (charAt(0)) ke variabel sim
    - Tutup objek Scanner dengan a.close()
    - Gunakan empat kondisi if terpisah untuk mengecek kombinasi umur dan SIM:

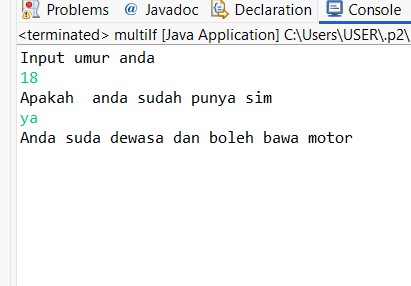


* + - Jika umur >=17 && sim == 'y' tampilkan "Anda sudah dewasa dan boleh bawa motor"
    - Jika umur >= 17 && sim != 'y' tampilkan "Anda sudah dewasa tapi tidak boleh bawa motor".
    - Jika umur < 17 && sim !='y' tampilkan "Anda belum cukup umur bawa motor".
    - Jika umur <17 && sim == 'y' tampilkan "anda belum cukup umur punya sim".



Gambar 2.5

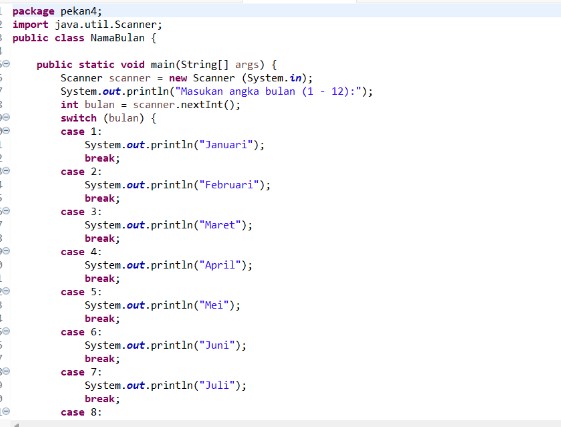
Output yang dihasilkan



Gambar 2.6

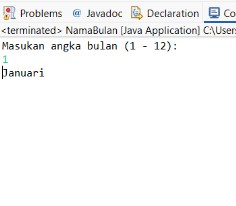
1. *Switch-case*

* Buat file Java bernama NamaBulan di package pekan4
* Impor library java.util.Scanner untuk membaca input pengguna
* Deklarasikan kelas publik NamaBulan dengan metode main
* Deklarasikan variabel int bulan untuk menyimpan angka input bulan
* Buat objek Scanner bernama scanner untuk membaca input dari keyboard
* Tampilkan pesan "Masukan angka bulan (1 - 12):" lalu baca nilai integer ke variabel bulan
* Tutup objek Scanner dengan scanner.close()
* Gunakan struktur switch-case untuk mengecek nilai variabel bulan:
* Jika bulan == 1, tampilkan "Januari"
* Jika bulan == 2, tampilkan "Februari"
* Jika bulan == 3, tampilkan "Maret"
* Jika bulan == 4, tampilkan "April"
* Jika bulan == 5, tampilkan "Mei"
* Jika bulan == 6, tampilkan "Juni"
* Jika bulan == 7, tampilkan "Juli"
* Jika bulan == 8, tampilkan "Agustus"
* Jika bulan == 9, tampilkan "September"
* Jika bulan == 10, tampilkan "Oktober"
* Jika bulan == 11, tampilkan "November"
* Jika bulan == 12, tampilkan "Desember"
* Jika nilai bulan bukan antara 1–12, tampilkan "Angka tidak valid" (bagian default)



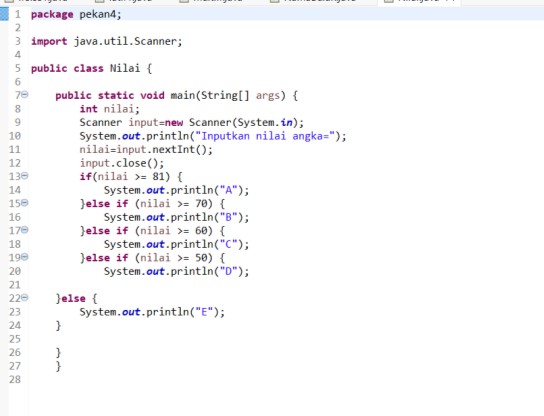
Gambar 2.7

Output yang dihasilkan

 Gambar 2.8

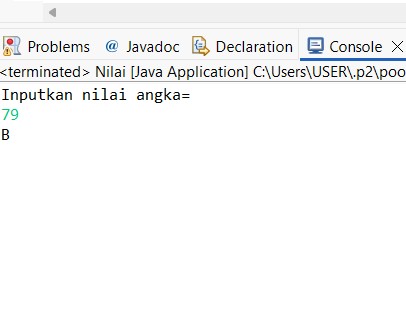
*5. Else if*

* - Buat file Java bernama Nilai di package pekan4
* - Impor library java.util.Scanner untuk membaca input pengguna
* - Deklarasikan kelas publik Nilai dengan metode main
* - Deklarasikan variabel int nilai untuk menyimpan angka input nilai
* - Buat objek Scanner bernama input untuk membaca input dari keyboard
* - Tampilkan pesan "Inputkan nilai angka=" lalu baca nilai integer ke variabel nilai
* - Tutup objek Scanner dengan input.close()
* - Gunakan struktur if-else if-else bertingkat untuk mengecek rentang nilai:  - Jika nilai >= 81, tampilkan "A"
* - Jika nilai >= 70 (dan kurang dari 81), tampilkan "B"
* - Jika nilai >= 60 (dan kurang dari 70), tampilkan "C"
* - Jika nilai >= 50 (dan kurang dari 60), tampilkan "D"  - Jika nilai < 50, tampilkan "E"



Gambar 2.8

Output yang dihasilkan



Gambar 2.9

# **BAB III** **KESIMPULAN**

# 3.1 KESIMPULAN

Berdasarkan praktikum yang telah dilakukan, dapat disimpulkan bahwa struktur kontrol percabangan berfungsi untuk menentukan alur program sesuai kondisi yang diberikan. Melalui penggunaan if, if-else, maupun switch, program dapat mengambil keputusan dengan benar sehingga menghasilkan output yang sesuai

# 3.2 SARAN

1. Akan lebih baik bila dosen menyelenggarakan sesi pra-praktikum di kelas agar mahasiswa dapat memperoleh pemahaman awal yang lebih memadai, sehingga dapat menghindari kepanikan atau kesalahan saat praktikum

1. Sebaiknya dosen membagikan materi praktikum terlebih dahulu melalui iLearn agar mahasiswa bisa mempersiapkan diri sebelum pelaksanaan praktikum

# **DAFTAR PUSTAKA**

[5] Nugroho, ADI . *Algoritma dan Struktur Data dalam Bahasa Java*.

Yogyakarta: Penerbit Andi, 20O8.

[2] Hartono, Budi dan tim, *Pemrograman Java untuk Pemula*.

Semarang: Penerbit Yayasan Prima Agus Teknik; 2022..