

MODUL PRAKTIKUM

# ALGORITMA PEMROGRAMAN

S1 REKAYASA PERANGKAT LUNAK



*Published by school of computing*

Our official instagram



@informaticslab\_telu

## LEMBAR PENGESAHAN

Saya yang bertanda tangan di bawah ini:

Nama : Aaz Muhammad Hafidz Aziz, S.T., M.T.

NIP : 23990046

Koordinator Mata Kuliah : Algoritma Pemrograman

Prodi : S1 Rekayasa Perangkat Lunak

Menerangkan dengan sesungguhnya bahwa modul ini digunakan untuk pelaksanaan praktikum di Semester Ganjil Tahun Ajaran 2024/2025 di Laboratorium Informatika, Fakultas Informatika, Universitas Telkom.



Bandung, 20 Agustus 2024  
**Fakultas Informatika**  
**School of Computing**  
**Telkom University**



informatics lab

Mengesahkan,

Koordinator Mata Kuliah

Algoritma Pemrograman

Aaz Muhammad Hafidz Aziz, S.T., M.T.

NIP. 23990046

Mengetahui,

Kaprodi S1 Rekayasa Perangkat Lunak

Dr. Mira Kania Sabariah, S.T., M.T.

NIP. 14770011

## DAFTAR ISI

<b>LEMBAR PENGESAHAN.....</b>	<b>i</b>
<b>DAFTAR ISI.....</b>	<b>ii</b>
<b>DAFTAR GAMBAR .....</b>	<b>v</b>
<b>DAFTAR TABEL .....</b>	<b>vi</b>
<b>PRAKATA .....</b>	<b>vii</b>
<b>Modul 1 Running Modul.....</b>	<b>1</b>
1. Peraturan Praktikum Laboratorium Informatika .....	1
1. Penyampaian Keluhan Praktikum IFLAB Melalui iGracias .....	3
2. Penyampaian Keluhan Praktikum IFLAB Melalui Situs Web.....	3
3. Persiapan Praktikum Algoritma Pemrograman .....	4
a. Download Java.....	4
b. Instalasi Java .....	5
c. Cara Setting Path Java di Windows .....	7
<b>Modul 2 Pengenalan Pemrograman Java .....</b>	<b>11</b>
1. Tujuan .....	11
2. Perkenalan Bahasa Java .....	11
3. Cara Kerja Java .....	12
4. Lingkup Pengembangan Java .....	12
5. Hello World .....	13
<b>Modul 3 Variabel dan Operator .....</b>	<b>14</b>
1. Tujuan .....	14
2. Variabel dalam bahasa Java .....	14
a. Variabel dalam bahasa Java.....	14
b. Deklarasi dan Inisialisasi Variabel.....	15
c. Pernyataan Masukan dan Keluaran.....	16
3. Operator.....	17
4. Contoh Program.....	18
5. Latihan Soal.....	20
<b>Modul 4 Runtunan/Sekuensi .....</b>	<b>21</b>
1. Tujuan .....	21
2. Runtunan/ Sekuensi.....	21
3. Contoh Program.....	21
4. Latihan Soal.....	22
<b>Modul 5 Percabangan If dan switch .....</b>	<b>23</b>
1. Tujuan .....	23
2. Struktur kontrol If .....	23
a. If-then .....	23
b. If-else .....	23
c. Else-if .....	23
3. Struktur kontrol Switch.....	24
4. Contoh Program.....	24
5. Latihan Soal.....	27

<b>Modul 6 Perulangan while do - do while .....</b>	<b>28</b>
1. Tujuan .....	28
2. While do .....	28
3. Do While .....	28
4. Perbandingan while dan do-while .....	28
5. Contoh Program.....	29
6. Latihan Soal.....	30
<b>Modul 7 Perulangan for .....</b>	<b>31</b>
1. Tujuan .....	31
2. Perulangan For.....	31
3. Contoh Program.....	31
4. Latihan Soal.....	32
<b>Modul 8 Assessment 1 .....</b>	<b>33</b>
<b>Modul 9 Method tanpa Nilai Kembalian .....</b>	<b>34</b>
1. Tujuan .....	34
2. Method .....	34
a. Karakteristik Method Tanpa Nilai Kembalian .....	34
3. Contoh Program.....	35
a. Method untuk Mencetak Pesan .....	35
b. Method untuk Mengubah Nilai Variabel .....	35
5. Latihan Soal.....	36
<b>Modul 10 Method dengan Kembalian.....</b>	<b>37</b>
1. Tujuan .....	37
2. Method .....	37
a. Karakteristik Method dengan Nilai Kembalian.....	37
3. Contoh Program.....	38
a. Method untuk Mengembalikan Bilangan Bulat (int).....	38
b. Method untuk Mengembalikan String .....	38
c. Method untuk Mengembalikan Boolean .....	39
<b>Modul 11 Assesment 2 .....</b>	<b>41</b>
<b>Modul 12 Array / Larik I (dasar konsep) .....</b>	<b>42</b>
1. Tujuan .....	42
2. Array.....	42
3. Karakteristik Array .....	42
4. Deklarasi.....	42
5. Inisialisasi .....	42
6. Mengakses Elemen Array.....	43
7. Looping Melalui Array.....	43
a. Menggunakan Loop for:.....	43
b. Menggunakan Loop for-each: .....	43
8. Contoh Program.....	43
a. ArrayEx.....	43
b. Menghitung Rata-Rata.....	44
9. Latihan Soal.....	45
<b>Modul 13 Pencarian Nilai Ekstrim .....</b>	<b>46</b>

1. Tujuan .....	46
2. Find Min Max .....	46
a. Algoritma untuk Menemukan Nilai Minimum dan Maksimum .....	46
b. Langkah-Langkah Implementasi .....	46
3. Contoh Program.....	46
4. Latihan Soal.....	47
<b>Modul 14 Review Materi.....</b>	<b>48</b>
<b>Modul 15 Assesment 3 .....</b>	<b>49</b>
<b>Modul 16 Praktikum Susulan .....</b>	<b>50</b>
<b>DAFTAR PUSTAKA .....</b>	<b>51</b>



## DAFTAR GAMBAR

Gambar 1.1 Download Java .....	5
Gambar 1.2 Membuka file jdk .....	5
Gambar 1.3 Direktori install JDK .....	6
Gambar 1.4 Proses Instalasi JDK .....	6
Gambar 1.5 Instalasi JDK Selesai.....	7
Gambar 1.6 Direktori JDK yang telah di install.....	7
Gambar 1.7 Menu Edit System Environment Variable pada Search Windows .....	8
Gambar 1.8 System Properties Environment Variable .....	8
Gambar 1.9 Daftar Environment variables default.....	9
Gambar 1.10 Menambahkan Environment Variables untuk JDK .....	9
Gambar 1.11 Menyimpan Environment Variables yang sudah ditambahkan .....	10
Gambar 1.12 Cek setting Path .....	10
Gambar 2.1 Cara Kerja Java .....	12
Gambar 2.2 Menulis Program Java Pertama .....	13
Gambar 2.3 Menulis Program Java Pertama .....	13
Gambar 3.1 Tipe data dalam Bahasa Java.....	14
Gambar 3.2 Deklarasi dan Inisialisasi Variabel dalam bahasa Java .....	15
Gambar 3.3 Keluaran dalam bahasa Java .....	16
Gambar 3.4 Latihan Soal Variabel dan Operator .....	20
Gambar 4.1 Latihan Soal Runtutan/Sekuensi 1 .....	22
Gambar 4.2 Latihan Soal Runtutan/Sekuensi 2 .....	22
Gambar 5.1 Latihan Soal Percabangan 1 .....	27
Gambar 5.2 Latihan Soal Percabangan 2 .....	27
Gambar 6.1 Latihan Soal Perulangan While 1.....	30
Gambar 6.2 Latihan Soal Perulangan While 2.....	30
Gambar 6.3 Latihan Soal Perulangan While 3.....	30
Gambar 7.1 Latihan Soal Perulangan for .....	32
Gambar 9.1 Latihan Soal Method tanpa Nilai Kembalian .....	36
Gambar 12.1 Latihan Soal Array 1 .....	45
Gambar 12.2 Latihan Soal Array 2 .....	45
Gambar 14.1 Latihan Soal Pencarian Nilai Ekstrem .....	47

## DAFTAR TABEL

Tabel 2.1 Cara Kerja Java .....	12
Tabel 3.1 Nilai Default pada Java .....	16
Tabel 3.2 Operator pada Java .....	18
Tabel 6.1 Perbandingan While do dan do While .....	29



## PRAKATA

Buku modul praktikum Algoritma Pemrograman ini merupakan panduan mahasiswa dalam pelaksanaan praktikum dalam mata kuliah Algoritma Pemrograman di Program Studi S1 Rekayasa Perangkat Lunak, Fakultas Informatika, Universitas Telkom. Buku modul ini berisi materi, contoh soal, dan kumpulan soal-soal latihan yang bisa digunakan mahasiswa untuk belajar sebelum pelaksanaan praktikum dimulai. Bahasa pemrograman yang digunakan adalah Java. Materi dalam modul praktikum ini akan membahas mengenai konsep fundamental pemrograman menggunakan Java, seperti Variabel dan Operator, Percabangan, Perulangan, Method, serta Array/Larik, akan dijelaskan secara lebih detail dalam modul ini untuk memberikan pemahaman yang lebih mendalam tentang Algoritma Pemrograman.

Buku modul praktikum ini terdiri dari 16 modul yang akan dibahas secara berurutan sepanjang periode praktikum. Setiap pekan, mahasiswa akan melaksanakan kegiatan praktikum sesuai dengan urutan modul yang terdapat pada buku ini. Meskipun buku modul ini mengajarkan algoritma dengan pendekatan praktikum pemrograman menggunakan Java, buku ini tidak dapat dijadikan sebagai referensi utama untuk pembelajaran pemrograman secara lebih luas. Hal ini disebabkan oleh fokus materi dan tingkat kesulitan soal yang disesuaikan dengan course learning outcome (CLO) yang telah ditetapkan dalam rencana pembelajaran semester pada mata kuliah Algoritma Pemrograman di Program Studi S1 Rekayasa Perangkat Lunak, Fakultas Informatika, Universitas Telkom.

Buku modul ini diharapkan bisa memberikan manfaat kepada mahasiswa atau praktikan dalam menguasai materi yang diberikan pada mata kuliah Algoritma Pemrograman.

Bandung, 21 Agustus 2024

Tim Penulis



## Modul 1 Running Modul

### 1. Peraturan Praktikum Laboratorium Informatika

- 2) Praktikum diampu oleh dosen kelas dan dibantu oleh asisten laboratorium dan asisten praktikum.
- 3) Praktikum dilaksanakan di Gedung Telkom University Lanmark Tower (TULT) Lantai 6 dan Lantai 7 sesuai jadwal yang ditentukan.
- 4) Praktikan wajib membawa modul praktikum, kartu praktikum, dan alat tulis.
- 5) Praktikan wajib mengecek kehadiran di igracias dan sheet yang dibagikan asisten.
- 6) Durasi kegiatan praktikum S-1 = 2 jam (100 menit).
- 7) Jumlah pertemuan praktikum:
  - 16 kali pertemuan
- 8) Praktikan wajib hadir minimal 75% dari seluruh pertemuan praktikum di lab.
- 9) Praktikan yang datang terlambat :
  - a.  $\leq 5$  menit : diperbolehkan mengikuti praktikum tanpa tambahan praktikum.
  - b.  $\leq 30$  menit : tidak diperbolehkan mengikuti praktikum.
- 10) Saat praktikum berlangsung, asisten praktikum dan praktikan:
  - a. Wajib menggunakan seragam sesuai aturan institusi.
  - b. Wajib mematikan/ mengkondisikan semua alat komunikasi.
  - c. Dilarang membuka aplikasi yang tidak berhubungan dengan praktikum yang berlangsung.
  - d. Dilarang mengubah pengaturan *software* maupun *hardware* komputer tanpa ijin.
  - e. Dilarang membawa makanan maupun minuman di ruang praktikum.
  - f. Dilarang memberikan jawaban ke praktikan lain.
  - g. Dilarang menyebarkan soal praktikum.
  - h. Dilarang membuang sampah di ruangan praktikum.
  - i. Wajib meletakkan alas kaki dengan rapi pada tempat yang telah disediakan.

11) Setiap praktikan dapat mengikuti praktikum susulan maksimal dua modul untuk satu mata kuliah praktikum.

- a. Praktikan yang dapat mengikuti praktikum susulan hanyalah praktikan yang memenuhi syarat sesuai ketentuan institusi, yaitu: sakit (dibuktikan dengan surat keterangan medis), tugas dari institusi (dibuktikan dengan surat dinas atau dispensasi dari institusi), atau mendapat musibah atau keduakaan (menunjukkan surat keterangan dari orangtua/wali mahasiswa.)
- b. Persyaratan untuk praktikum susulan diserahkan sesegera mungkin kepada asisten laboratorium untuk keperluan administrasi.
- c. Praktikan yang diijinkan menjadi peserta praktikum susulan ditetapkan oleh Asman Lab dan Bengkel Informatika dan tidak dapat diganggu gugat.

12) a. Ketidakhadiran pada kelas praktikum

- Nilai modul = 0

b. Meminta, mendapatkan, dan menyebarluaskan soal dan atau kunci jawaban praktikum:

- Penyebar soal dan kunci jawaban: Pengajuan sanksi kepada Komisi Disiplin Fakultas
- Penerima soal dan kunci jawaban: Nilai '0' pada (seluruh assesment) praktikum

c. Lupa menghapus file praktikum

- Pengurangan nilai modul 20%

d. Memicu kegaduhan, sehingga membuat situasi tidak kondusif (jalan-jalan, mengganggu teman, mengobrol, dll), asisten praktikum diwajibkan menegur sebanyak 3x

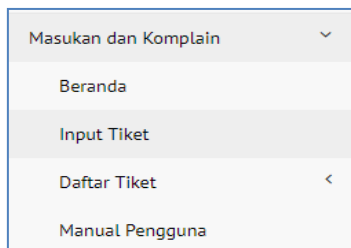
- Pengurangan nilai modul 50%

e. MENYALAHGUNAKAN FITUR LMS

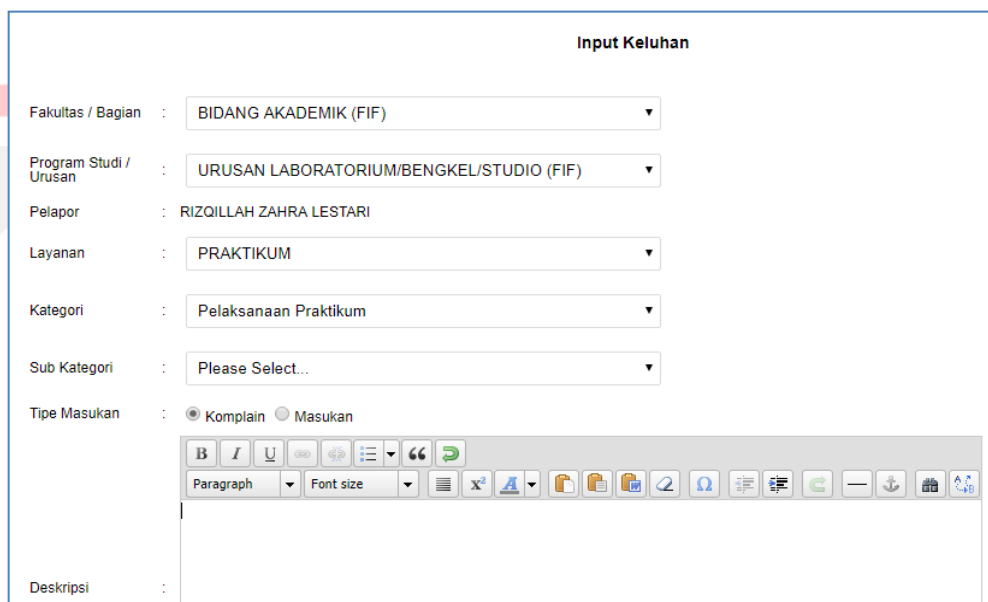
- Siap menerima sanksi dari lab

## 1. Penyampaian Keluhan Praktikum IFLAB Melalui iGracias

- 1) Login **iGracias**
- 2) Pilih menu **Masukan dan Komplain**, pilih **Input Tiket**



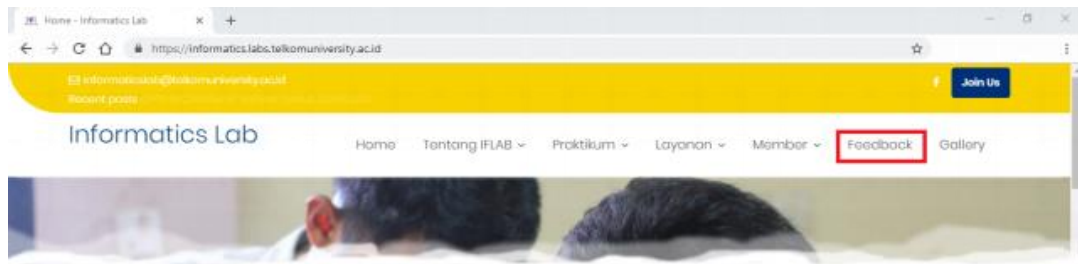
- 3) Pilih Fakultas/Bagian: **Bidang Akademik (FIF)**
- 4) Pilih Program Studi/Urusan: **Urusan Laboratorium/Bengkel/Studio (FIF)**
- 5) Pilih Layanan: **Praktikum**
- 6) Pilih Kategori: **Pelaksanaan Praktikum**, lalu pilih **Sub Kategori**.
- 7) Isi **Deskripsi** sesuai komplain yang ingin disampaikan.

A screenshot of a web form titled 'Input Keluhan'. The form contains several dropdown menus and a text area. The fields are: 'Fakultas / Bagian' (set to 'BIDANG AKADEMIK (FIF)'), 'Program Studi / Urusan' (set to 'URUSAN LABORATORIUM/BENGKEL/STUDIO (FIF)'), 'Pelapor' (set to 'RIZQILLAH ZAHRA LESTARI'), 'Layanan' (set to 'PRAKTIKUM'), 'Kategori' (set to 'Pelaksanaan Praktikum'), 'Sub Kategori' (set to 'Please Select...'), 'Tipe Masukan' (with radio buttons for 'Komplain' and 'Masukan', where 'Komplain' is selected), and 'Deskripsi' (a large text area with a rich text editor toolbar above it). The toolbar includes icons for bold, italic, underline, text color, background color, bulleted list, numbered list, link, unlink, and other formatting options.

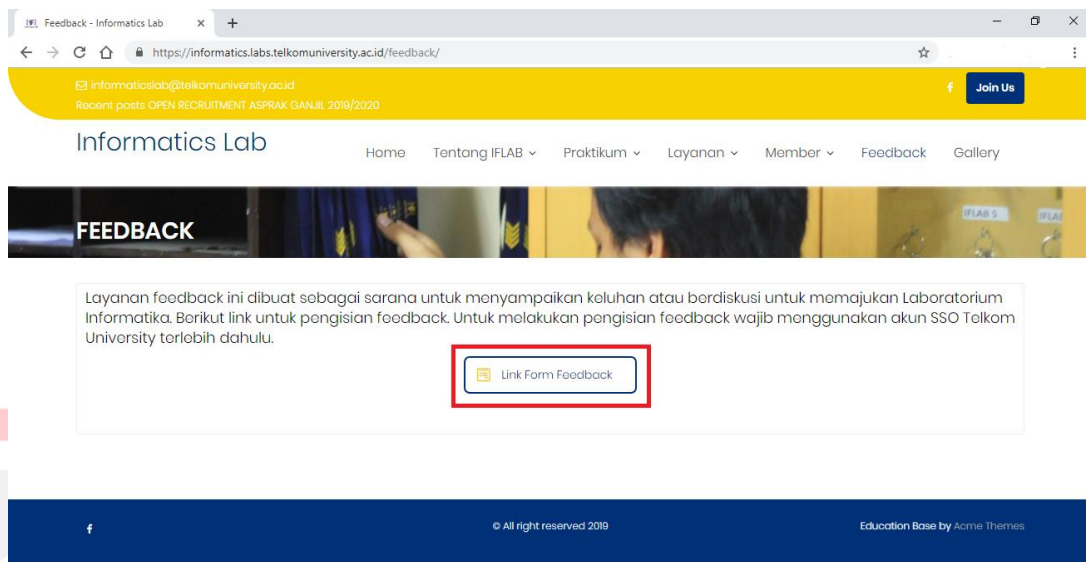
- 8) Lampirkan file jika perlu. Lalu klik Kirim.

## 2. Penyampaian Keluhan Praktikum IFLAB Melalui Situs Web

- 1) Gunakan browser untuk membuka situs <https://informatics.labs.telkomuniversity.ac.id/>
- 2) Pilih menu **Feedback** pada navigation bar website



3) Pilih tombol Link Form Feedback.



4) Lakukan login menggunakan akun SSO Telkom University untuk mengakses form feedback.

5) Isi form sesuai dengan feedback yang ingin diberikan.

### 3. Persiapan Praktikum Algoritma Pemrograman

Sebelum belajar pemrograman Java, kita perlu menginstall compiler dan runtime untuk Java nya terlebih dahulu. Dimana terdapat dua jenis aplikasi Java

- JDK (Java Development Kit), yaitu aplikasi yang digunakan untuk melakukan kompilasi kode program Java menjadi file binary Java
- JRE (Java Runtime Environment), yaitu aplikasi yang digunakan untuk menjalankan file binary Java yang telah di compile oleh JDK

Namun saat kita menginstall JDK, kita tidak perlu lagi menginstall JRE, karena di dalam JDK sudah terdapat juga JRE nya. Oleh karena itu untuk menginstall Java, kita cukup download aplikasi JDK

a. Download Java

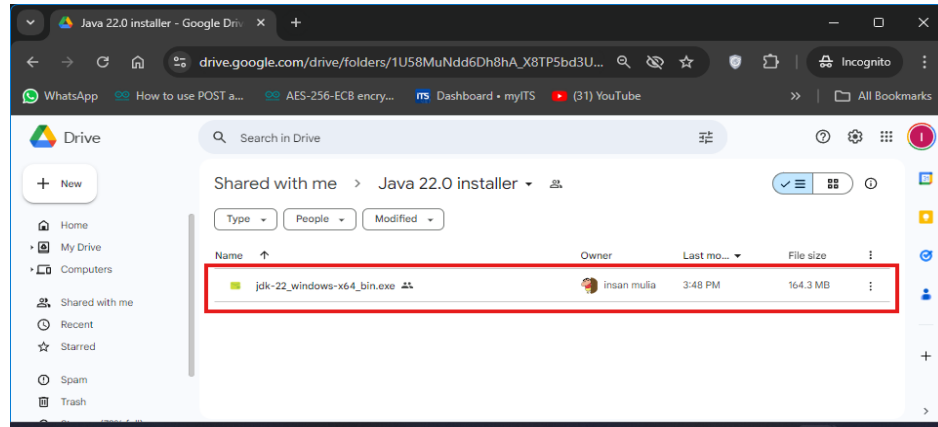
i. Kunjungi link yang telah disediakan berikut.

<https://rb.gy/h8yazq>

atau

<https://www.oracle.com/id/java/technologies/downloads/>

- ii. Download JDK yang telah disediakan.



Gambar 1.1 Download Java

## b. Instalasi Java

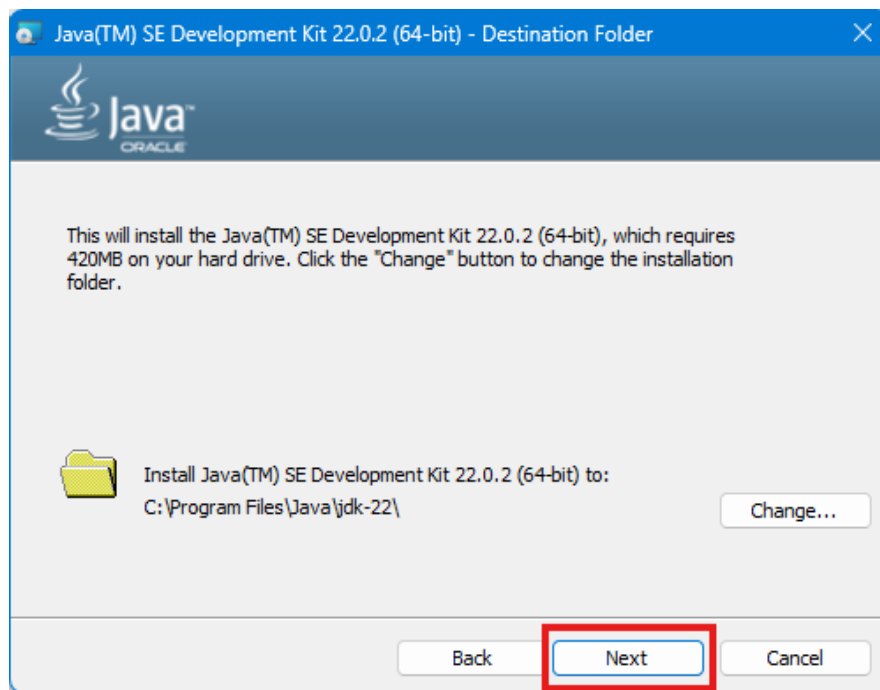
Untuk melakukan proses instalasi Java, pastikan anda sudah mendownload JDK terlebih dahulu. Anda dapat mengikuti langkah-langkah berikut untuk melakukan proses instalasi Java.

- i. Klik open pada JDK yang telah anda install, kemudian klik Next



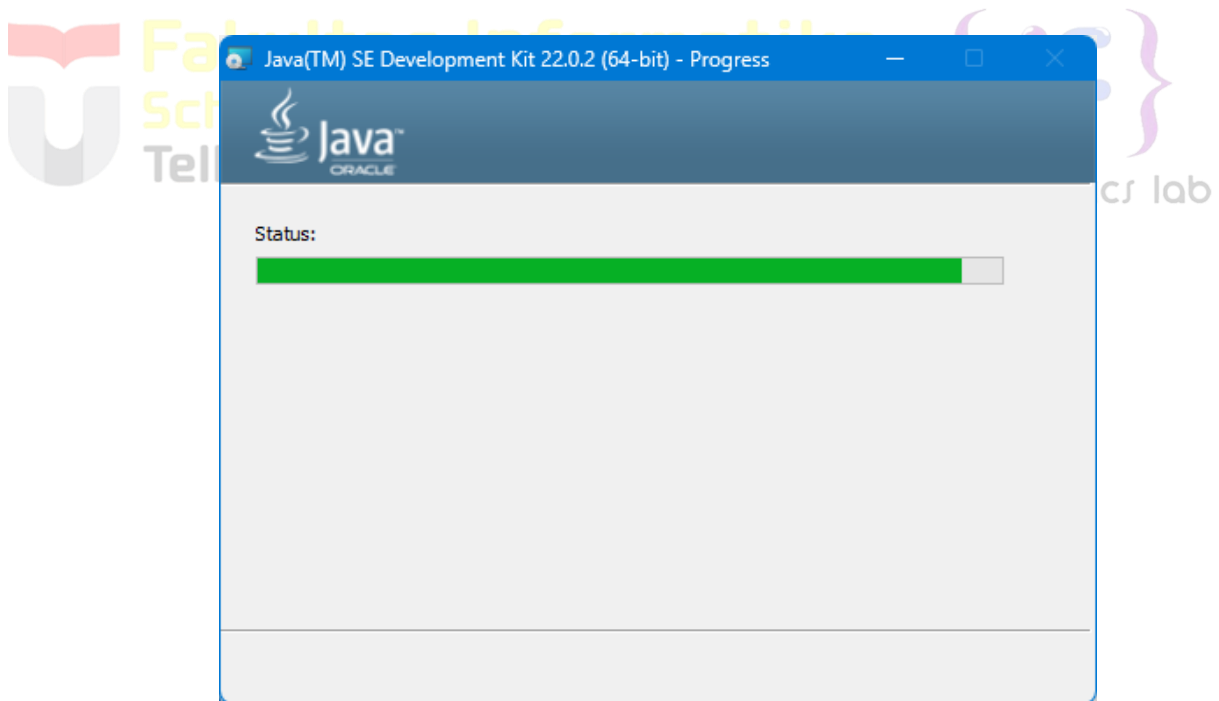
Gambar 1.2 Membuka file jdk

- ii. Jika anda ingin mengubah direktori instalasi, anda dapat mengubahnya pada tombol 'Change', kemudian klik Next. Jika tidak, anda dapat langsung klik Next.



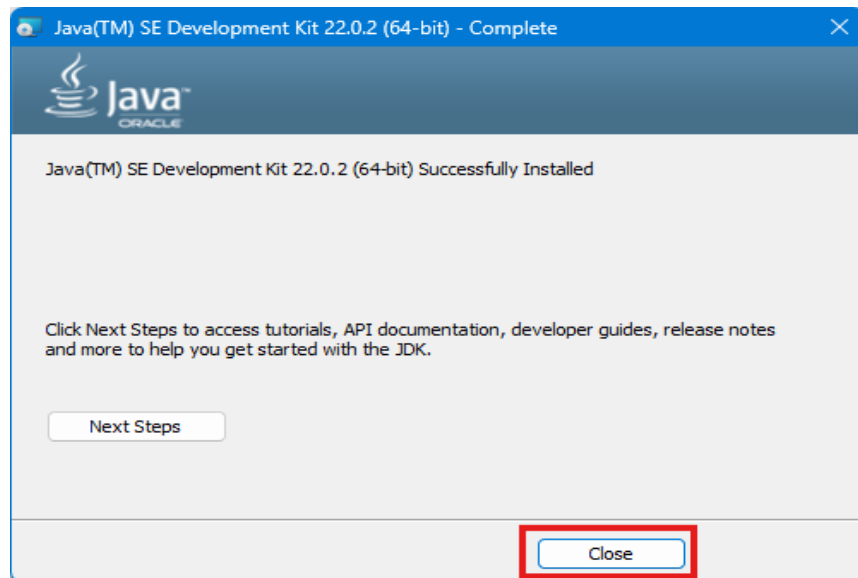
Gambar 1.3 Direktori install JDK

iii. Tunggu hingga proses instalasi selesai.



Gambar 1.4 Proses Instalasi JDK

iv. Klik 'Close' untuk menutup proses instalasi

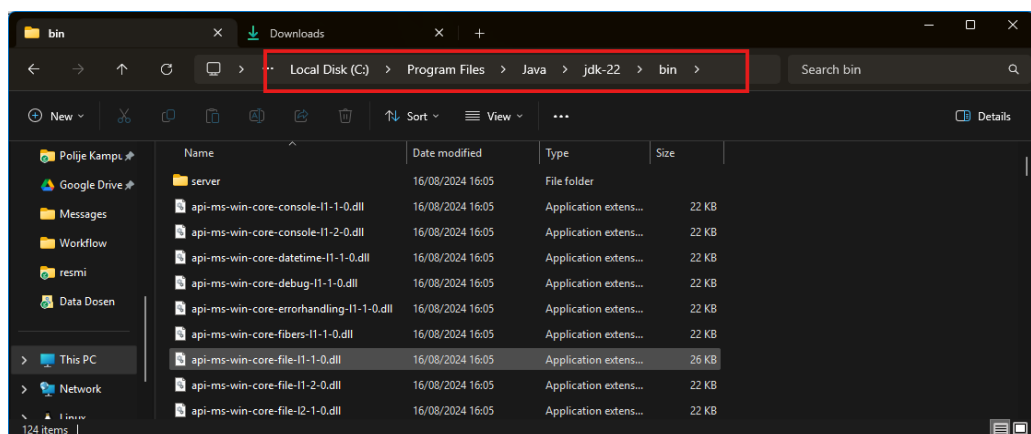


Gambar 1.5 Instalasi JDK Selesai

### c. Cara Setting Path Java di Windows

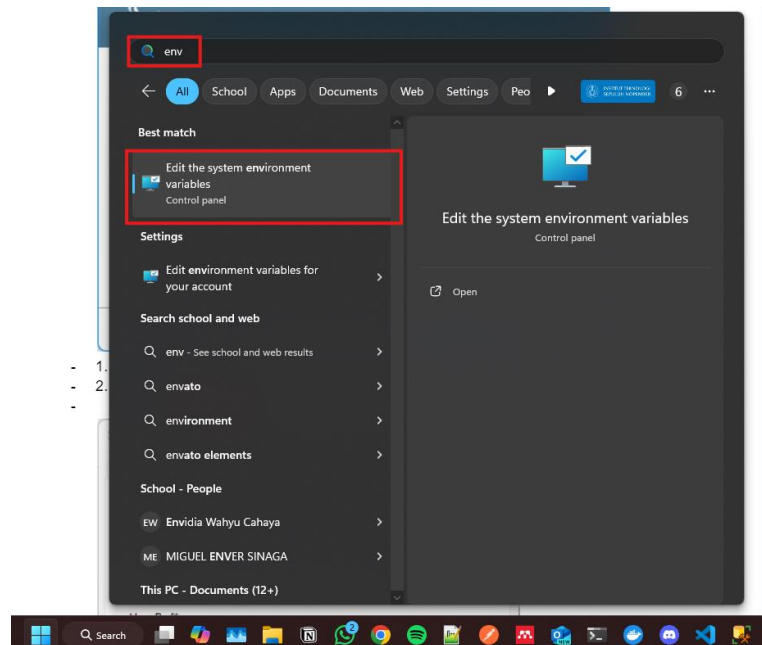
Langkah selanjutnya setelah anda berhasil melakukan proses instalasi Java adalah melakukan Setting Path pada komputer anda. Berikut adalah langkah-langkah yang dapat anda ikuti untuk melakukan setting path Java di Windows

- i. Buka direktori dari java yang telah anda install. Jika anda tidak mengubah direktori pada saat install Java, maka direktori tersebut adalah 'C:\Program Files\Java\jdk-22\bin'



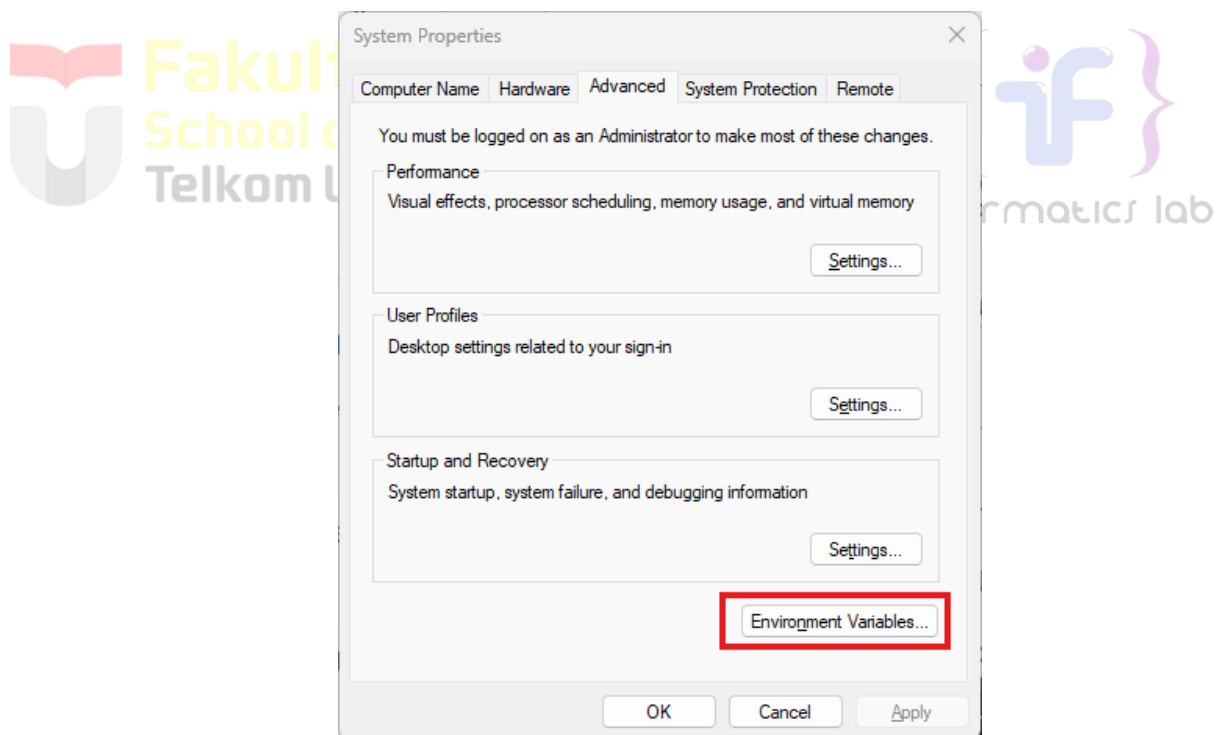
Gambar 1.6 Direktori JDK yang telah di install

- ii. Cari 'Environment' pada menu search di Windows, dan pilih menu seperti di bawah



Gambar 1.7 Menu Edit System Environment Variable pada Search Windows

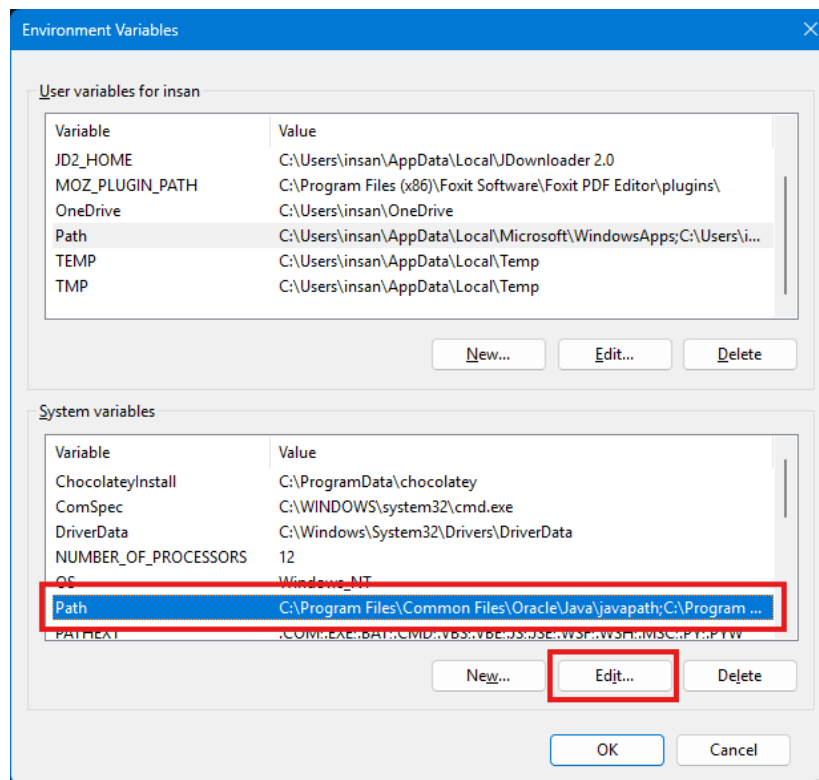
iii. Klik 'Environment Variables'



Gambar 1.8 System Properties Environment Variable

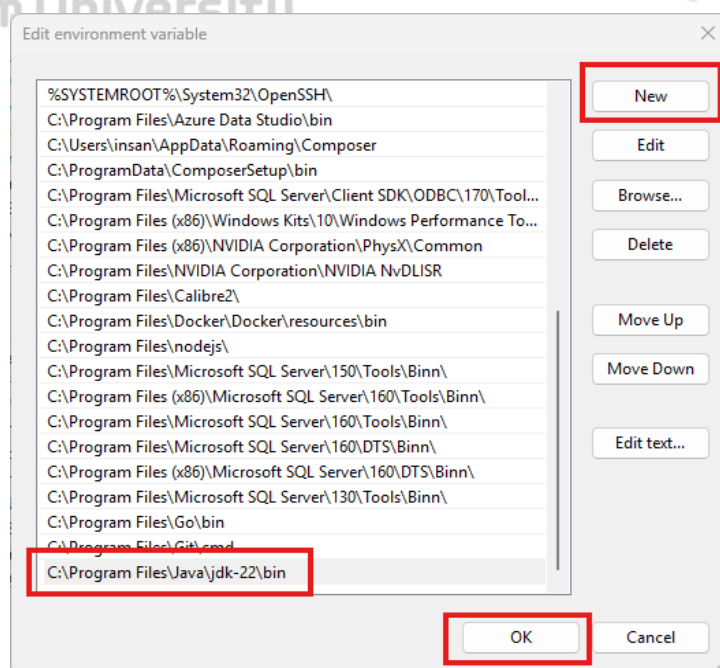
iv. Klik 'Edit' pada bagian System Variables





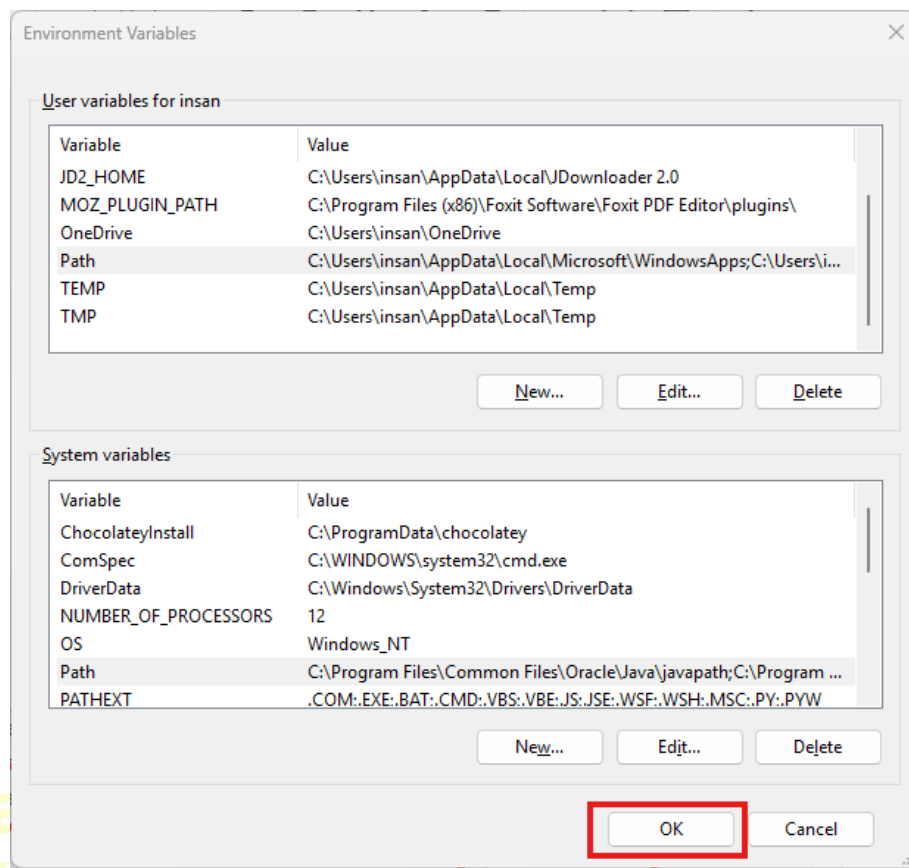
Gambar 1.9 Daftar Environment variables default

- v. Kemudian klik New, dan paste path yang telah anda copy di tahap 1, kemudian klik 'OK'



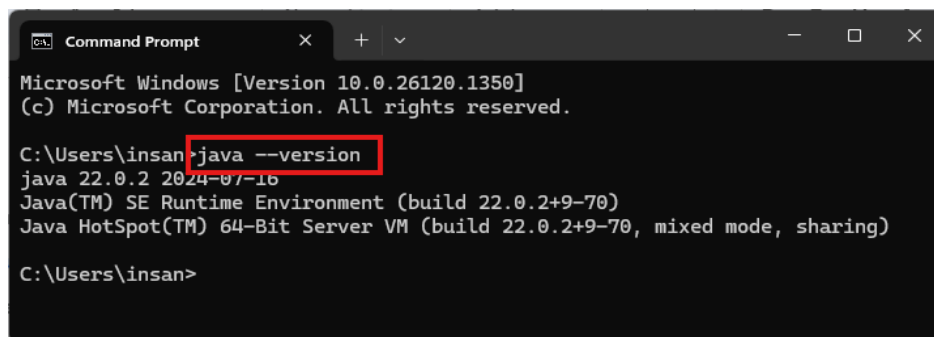
Gambar 1.10 Menambahkan Environment Variables untuk JDK

vi. Klik 'OK' pada tahap ini



Gambar 1.11 Menyimpan Environment Variables yang sudah ditambahkan

vii. Tulis 'java --version' di CMD, akan tampak seperti berikut jika proses setting path nya benar



Gambar 1.12 Cek setting Path

## Modul 2 Pengenalan Pemrograman Java

### 1. Tujuan

Dalam modul ini, kita akan mempelajari apa itu Java dan mengembangkan program java sederhana. Struktur pembahasan pada modul ini yaitu :

1. Perkenalan Bahasa Java
2. Cara Kerja Java
3. Lingkup Pengembangan
4. Getting Started

### 2. Perkenalan Bahasa Java

Java awalnya dikenal sebagai nama sebuah pulau di Indonesia dan juga nama kopi yang berasal dari wilayah tersebut. Namun, yang akan kita pelajari kali ini bukanlah tentang pulau atau kopi tersebut, melainkan tentang Java sebagai bahasa pemrograman.

Java adalah bahasa pemrograman berorientasi objek yang dikembangkan dan dirilis oleh Sun Microsystems di Amerika Serikat pada tahun 1995.

Ada dua teori mengenai asal-usul nama Java. Teori pertama menyebutkan bahwa nama tersebut diambil dari kombinasi huruf pertama nama para pengembangnya: James Gosling, Arthur Van Hoff, dan Andy Bechtolsheim. Teori kedua menyatakan bahwa nama Java dipilih karena para pengembang sering minum kopi jenis Java saat bekerja.

Sejarah Java sebagai bahasa pemrograman sebenarnya cukup singkat. Bagaimana bahasa ini diciptakan?

Pada tahun 1990, Sun Microsystems memulai sebuah proyek yang disebut Green Project, dengan tujuan mengembangkan kotak kontrol untuk televisi interaktif. Seiring berkembangnya proyek tersebut, menjadi jelas bahwa mereka membutuhkan sebuah program yang dapat berjalan di berbagai perangkat. Untuk itu, tim pengembangan Green Project menciptakan bahasa baru yang diberi nama Oak. James Gosling, pemimpin proyek ini, kemudian mengembangkan bahasa tersebut menjadi Java. Sun Microsystems akhirnya mengembangkan Java untuk digunakan di World Wide Web (WWW), yang kemudian menjadi sangat populer di seluruh dunia pada awal tahun 1990-an.

Java mewarisi fitur-fitur dari Oak, namun dikemas dalam bentuk yang lebih kecil, sederhana, dan efisien. Bahasa ini relatif mudah dipelajari karena menghilangkan sintaksis kompleks seperti pointer dan pernyataan go to (), yang sering menjadi sumber kesalahan di bahasa lain. Selain itu, Java juga menyediakan penanganan exception yang memastikan program tidak akan macet saat terjadi kesalahan yang tak terduga, sehingga meningkatkan keandalan program bagi pengguna.

### 3. Cara Kerja Java



Gambar 2.1 Cara Kerja Java

Step 1.	Step 2.	Step 3.	Step 4.
Membuat source code (nama file.java)	Compile source code Java	Java File eksekusi (nama file.class)	Eksekusi program Java
Buatlah source code Java sesuai dengan seperangkat aturan (grammar)	Kode Java yang dibuat akan diproses oleh compiler (dikompilasi). Compiler akan memeriksa kesalahan, dan jika tidak ada error, maka akan menghasilkan file yang dapat dieksekusi	File yang dihasilkan dari proses kompilasi. Pada perangkat yang mendukung Java, file ini akan ditafsirkan dan dieksekusi.	Anda dapat menjalankan eksekusi Java (*.Class) pada mesin virtual (JVM) yang dapat menjalankan java

Tabel 2.1 Cara Kerja Java

Istilah-istilah :

- Source Code: Kode program yang ditulis dalam bahasa pemrograman tertentu, yang dapat dibaca dan dipahami oleh manusia.
- Native Code: Kode yang telah dikompilasi dan dapat langsung dipahami serta dijalankan oleh komputer tanpa perlu proses interpretasi lebih lanjut.
- Compile: Proses mengonversi source code menjadi kode mesin atau native code yang dapat dijalankan oleh komputer.

Run: Proses menjalankan kode mesin atau native code pada komputer sehingga program tersebut dapat berfungsi sesuai dengan instruksi yang dituliskan.

### 4. Lingkup Pengembangan Java

JDK(Java Development Kit) adalah alat pengembang Java yang harus diinstall oleh pengembang yang akan mengembangkan menggunakan Bahasa Java.

### JVM (Java Virtual Machine)

JVM adalah mesin virtual yang menginterpretasikan dan mengeksekusi file bytecode Java, memungkinkan program Java untuk berjalan diberbagai platform.

### JRE (Java Runtime Environment)

JRE adalah lingkungan yang memungkinkan program yang ditulis dalam bahasa Java untuk dijalankan. JRE terdiri dari JVM dan API dasar yang dibutuhkan untuk menginterpretasikan dan mengeksekusi file bytecode Java. JRE harus diinstal untuk menjalankan program yang ditulis dalam bahasa Java.

### JDK (Java Development Kit)

JDK adalah paket pengembangan yang digunakan untuk membuat program menggunakan bahasa Java. JDK terdiri dari JRE, kompiler, dan berbagai utilitas lainnya. JDK dapat diunduh dari situs Oracle (<http://www.oracle.com>). Jika Anda ingin mengembangkan program dalam bahasa Java, Anda harus menginstal JDK.

## 5. Hello World



Gambar 2.2 Menulis Program Java Pertama



Gambar 2.3 Menulis Program Java Pertama

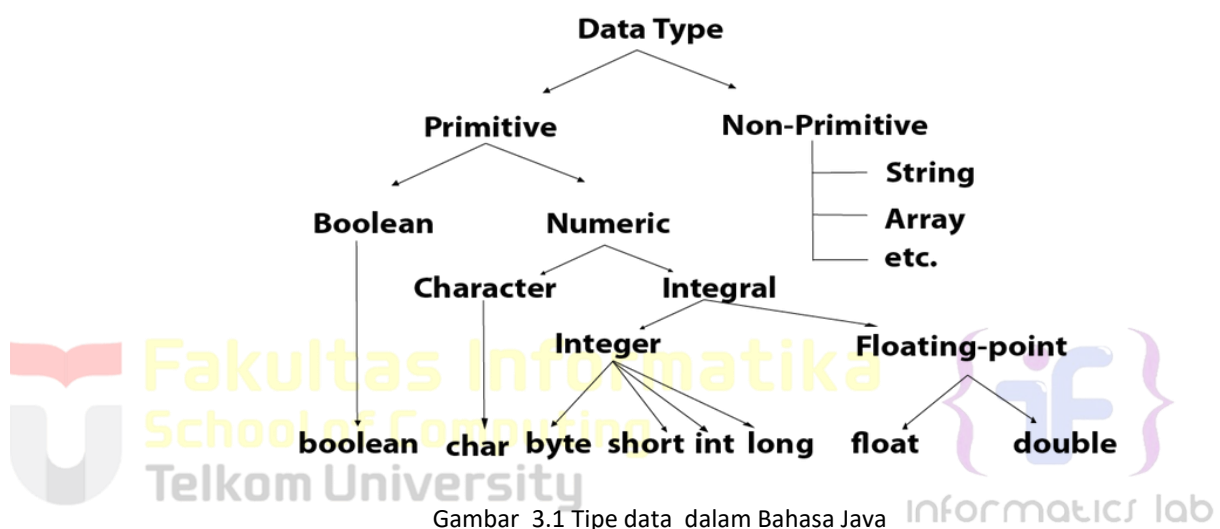
## Modul 3 Variabel dan Operator

### 1. Tujuan

Pada modul ini, Anda akan belajar tentang "variabel," yang bertanggung jawab untuk menyimpan data ketika mengimplementasikan program, dan melihat bagaimana mahasiswa dapat mendeklarasikan dan menggunakan variabel untuk tipe data yang berbeda.

### 2. Variabel dalam bahasa Java

#### a. Variabel dalam bahasa Java



Variabel adalah tempat penyimpanan data yang digunakan oleh program. Setiap variabel memiliki tipe data tertentu yang menentukan jenis data apa yang dapat disimpan. Ada beberapa jenis variabel dalam Java, antara lain:

**Tipe data primitif:** Tipe data primitif meliputi boolean, char, byte, short, int, long, float, dan double.

**Tipe data non-primitif:** Tipe data non-primitif meliputi String, Kelas (Classes), Antarmuka (Interfaces), dan Array.

#### i. Tipe Data Primitif

Tipe data primitif adalah tipe data dasar yang sudah disediakan oleh bahasa Java. Tipe data ini antara lain:

- **int:** Tipe data bilangan bulat. Contoh: `int a = 10;`
- **float:** Tipe data bilangan desimal dengan presisi tunggal. Contoh: `float b = 5.5f;`

- **double**: Tipe data bilangan desimal dengan presisi ganda. Contoh: `double c = 10.5;`
- **char**: Tipe data karakter. Contoh: `char d = 'A';`
- **boolean**: Tipe data yang hanya memiliki dua nilai, yaitu `true` atau `false`. Contoh: `boolean e = true;`
- **byte**: Tipe data bilangan bulat dengan ukuran 8 bit. Contoh: `byte f = 100;`
- **short**: Tipe data bilangan bulat dengan ukuran 16 bit. Contoh: `short g = 1000;`
- **long**: Tipe data bilangan bulat dengan ukuran 64 bit. Contoh: `long h = 100000L.`

## ii. Tipe Data non Primitif / reference

Tipe data referensi adalah tipe data yang merujuk ke objek. Tipe data ini termasuk kelas, antarmuka, array, dan lain-lain. Contoh penggunaan variabel dengan tipe data referensi:

```
String str = "Hello, World!";
int[] numbers = {1, 2, 3, 4, 5};
```

## b. Deklarasi dan Inisialisasi Variabel

[Grammar] **Data\_Type Nama variabel = Data(Nilai);**

[Contoh]     int     bornDay   = 20 ;  
                  int     year     = 1960 ;  
                  int     javaScore = 97 ;

Gambar 3.2 Deklarasi dan Inisialisasi Variabel dalam bahasa Java

## i. Nilai Default

Tidak selalu perlu untuk menentukan suatu nilai ketika sebuah field dideklarasikan. Field yang dideklarasikan tetapi tidak diinisialisasi akan di set ke default yang ada oleh kompiler. Secara umum, default ini akan bernilai null atau zero tergantung pada tipe datanya. Merangkum nilai default untuk tipe data di atas.

Tipe data	Nilai Default (untuk field)
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d
char	'\u0000'
String (atau obyek)	Null
boolean	False

Tabel 3.1 Nilai Default pada Java

### c. Pernyataan Masukan dan Keluaran

Dalam bahasa pemrograman Java, kita akan menggunakan pernyataan masukan dan keluaran. Pernyataan masukan adalah pernyataan untuk mendapatkan masukan dari keyboard. Sedangkan pernyataan keluaran adalah pernyataan untuk menampilkan suatu nilai ke layar.

#### i. Komponen Keluaran

Untuk menampilkan ke layar secara tekstual, java mempunyai fasilitas output, yaitu dengan perintah

[Grammar] **System.out.println( Nama Variabel );**

[Contoh]   int price = 5000 ;  
              System.out.println( price );

Gambar 3.3 Keluaran dalam bahasa Java

#### ii. Komponen Masukan

Untuk memasukkan sebuah nilai ke variabel yang sudah didefinisikan digunakan kelas Scanner.

##### ■ Input data bertipe Integer

Untuk menginputkan data dengan tipe integer digunakan method nextInt di dalam kelas Scanner.

##### ■ Input data bertipe String

Untuk menginputkan data dengan tipe integer digunakan method nextInt di dalam kelas Scanner.



■ Input tipe yang lain

Untuk input data dari keyboard dengan tipe yang lain dan disediakan oleh kelas Scanner adalah sebagai berikut

nextBoolean : input tipe boolean (true atau false saja)  
 nextShort : input tipe short integer  
 nextLong : input tipe long integer  
 nextFloat : input tipe float  
 nextDouble : input tipe double

### 3. Operator

Operator adalah simbol khusus yang menyajikan operasi khusus pada satu, dua, atau tiga operand dan kemudian mengembalikan hasilnya. Operator-operator tersebut digunakan untuk membentuk ekspresi. Sebuah konstanta atau entitas dengan sendirinya merupakan ekspresi, demikian juga kombinasi dari konstanta dan variabel dengan operator. Suatu ekspresi yang diikuti dengan sebuah titik koma adalah sebuah pernyataan. Operator mempunyai sifat- sifat unary, binary dan ternary.

Operator Type	Category	Precedence
Unary	postfix	expr++ expr--
	prefix	++expr --expr +expr -expr ~ !
Arithmetic	multiplicative	* / %
	additive	+ -
Relational	comparison	< > <= >= instanceof
	equality	== !=
Bitwise	bitwise AND	&
	bitwise exclusive OR	^
	bitwise inclusive OR	
Logical	logical AND	&&
	logical OR	
Ternary	ternary	? :
Assignment	assignment	= += -= *= /= %= &= ^=  = <<= >>= >>>=
Operator Type	Category	Precedence
Unary	postfix	expr++ expr--
	prefix	++expr --expr +expr -expr ~ !

Arithmetic	multiplicative	* / %
	additive	+ -
Relational	comparison	< > <= >= instanceof
	equality	== !=
Bitwise	bitwise AND	&
	bitwise exclusive OR	^
	bitwise inclusive OR	
Logical	logical AND	&&
	logical OR	
Ternary	ternary	? :
Assignment	assignment	= += -= *= /= %= &= ^=  = <<= >>= >>>=

Tabel 3.2 Operator pada Java

#### 4. Contoh Program

```
// Buat variabel untuk menyimpan nilai ulang tahun ibu saya,
// 20 Juni 1960, dan periksa nilai yang disimpan //.
public class VariableEx {
    public static void main(String[] args) {
        int bornYear; // Variabel menyimpan tahun kelahiran
        bornYear = 1960; // simpan nilai 1960 untuk tahun lahir
        int bornMonth; // Variabel untuk bulan lahir
        bornMonth = 6; // simpan nilai 6 untuk bulan lahir
        int bornDay = 20; // simpan nilai 20 untuk menyimpan
        variable tanggal lahir

        System.out.println(bornYear);
        System.out.println(bornMonth);
        System.out.println(bornDay);
    }
}
```

**output :**

```
PS D:\Dev\java\Modul> javac VariableEx.java
PS D:\Dev\java\Modul> java VariableEx.java
1960
6
20
```

```
public class DivisionEx {
    public static void main(String[] args) {
        int number1 = 10;
        int number2 = 3;

        int divisionResult = number1 / number2;
        int remainderResult = number1 % number2;

        System.out.println(number1 + " /" + number2);
        System.out.println("Hasil Pembagian :" + divisionResult);
        System.out.println("Hasil Modular :" + remainderResult);
    }
}
```



**output :**

```
PS D:\Dev\java\Modul> javac DivisionEx.java
PS D:\Dev\java\Modul> java DivisionEx.java
10 /3
Hasil Pembagian :3
Hasil Modular :1
```

## 5. Latihan Soal

- **Nama Package :** com.operator
- **Nama Class :** CalculatorTest
- **Isi Implementasi**  
Tulislah sebuah program kalkulator sederhana.  
Nyatakan angka pertama 20 dan angka kedua 7, dan lakukan operasi aritmatika dasar yaitu penjumlahan, pengurangan, perkalian, dan pembagian. Terakhir, lakukan operasi lainnya.
- **Hasil Run**  
Hasil penjumlahan: 27  
Hasil Pengurangan: 13  
Hasil perkalian: 140  
Hasil Pembagian: 2  
Sisa: 6

Gambar 3.4 Latihan Soal Variabel dan Operator



## Modul 4 Runtunan/Sekuensi

### 1. Tujuan

Pada modul ini mahasiswa akan mempelajari cara menyelesaikan suatu masalah dengan mengimplementasikan algoritma beruntun ke dalam sebuah program. Fokus utama adalah bagaimana menyusun langkah-langkah yang logis dan sistematis untuk mencapai solusi dari sebuah permasalahan.

### 2. Runtunan/ Sekuensi

Sekuensi adalah pemrograman sederhana yang hanya dapat memecahkan masalah-masalah yang sederhana. Masalah yang dapat diselesaikan dengan sekuensi saja biasanya hanya satu masalah kecil yang berdiri sendiri. Pembahasan pada modul sebelumnya, sejauh ini menggunakan prinsip sekuensi.

### 3. Contoh Program

Buatkan program untuk menghitung konversi dari meter ke centimeter dan dari inci ke centimeter seperti berikut ini :

1 meter = 100  
1 inci = 2.54 cm

```
import java.util.Scanner;

public class Konversi {
    public static void main(String args[]) {
        Scanner masuk = new Scanner(System.in);
        float m, cm, inci;
        System.out.print("Masukan ukuran dalam Meter:");
        m = masuk.nextFloat();
        cm = m * 100;
        inci = m * 100 / 2.54f;
        System.out.println("Ukuran dalam CM = " + cm);
        System.out.println("Ukuran dalam Inci = " + inci);
    }
}
```

#### output :

```
PS D:\Dev\java\Modul> javac Konversi.java
PS D:\Dev\java\Modul> java Konversi.java
Masukan ukuran dalam Meter:50
Ukuran dalam CM = 5000.0
Ukuran dalam Inci = 1968.5039
```

#### 4. Latihan Soal

- **Nama Package** : com.runtunan
- **Nama Class** : TestScore
- **Rincian implementasi**  
Tulislah sebuah program untuk mencetak nilai SW Boot Camp dari Rian, seorang Karyawan A  
Rian mendapatkan nilai 100 dalam Pemrograman Java dan 80 dalam SQL, Servlet & JSP dengan skor 85.  
Menerapkan cara untuk menyimpan nilai setiap subjek dalam sebuah variabel integer dan mencetak nilai apa adanya.
- **Hasil Run**  
100  
80  
85

Gambar 4.1 Latihan Soal Runtutan/Sekuensi 1

- **Nama Package** : com.runtunan
- **Nama Class** : GradeTest
- **Implementasi isi**  
Saya mengikuti tes dan mendapat nilai 75. Bagaimana jika Anda ingin tahu berapa nilai 75?  
100 hingga 90 poin: Nilai A  
89 hingga 80 poin: Nilai B  
79 hingga 70 poin: C  
69 hingga 60 poin: D
- **Hasil run**  
A : salah  
B: salah  
C: benar  
D: salah

Gambar 4.2 Latihan Soal Runtutan/Sekuensi 2

## Modul 5 Percabangan If dan switch

### 1. Tujuan

Pada modul ini, Anda akan belajar tentang struktur kontrol percabangan dengan menggunakan If dan switch untuk menyelesaikan suatu masalah dengan membuat keputusan berdasarkan kondisi tertentu.

### 2. Struktur kontrol If

Struktur kontrol percabangan If adalah pernyataan kondisional yang digunakan untuk menjalankan baris program tertentu jika kondisi yang dibuat bernilai benar ('true'). Struktur percabangan If mempunyai 3 bentuk utama yang sering digunakan:

#### a. If-then

If-then adalah bentuk paling sederhana dari pernyataan if dimana program akan menjalankan aksi yang terdapat di dalam bentuk If jika kondisi yang dibuat bernilai benar ('true').

```
if (kondisi) {  
    // Blok kode yang dieksekusi jika kondisi benar  
}
```

#### b. If-else

Pernyataan **if else** digunakan ketika kita ingin mengeksekusi blok kode yang berbeda berdasarkan kondisi yang diberikan. Jika kondisi pada **if** benar, blok kode di dalam **if** akan dieksekusi; jika salah, maka blok kode di dalam **else** akan dieksekusi.

```
if (kondisi) {  
    // Blok kode yang dieksekusi jika kondisi benar  
} else {  
    // Blok kode yang dieksekusi jika kondisi salah  
}
```

#### c. Else-if

Pernyataan **else if** digunakan untuk mengecek beberapa kondisi yang berbeda secara berurutan. Jika kondisi pertama salah, maka akan dicek kondisi berikutnya, dan seterusnya. Jika salah satu kondisi benar, maka blok kode yang terkait dengan kondisi

tersebut akan dieksekusi. Jika semua kondisi salah, maka blok **else** akan dieksekusi (jika ada).

```
if (kondisi1) {  
    // Blok kode yang dieksekusi jika kondisi1 benar  
} else if (kondisi2) {  
    // Blok kode yang dieksekusi jika kondisi2 benar  
} else if (kondisi3) {  
    // Blok kode yang dieksekusi jika kondisi3 benar  
} else {  
    // Blok kode yang dieksekusi jika semua kondisi salah  
}
```

### 3. Struktur kontrol Switch

Struktur kontrol **switch** digunakan untuk memilih salah satu dari banyak blok kode yang akan dieksekusi. Ini sangat berguna ketika Anda ingin membandingkan nilai variabel dengan sejumlah nilai potensial dan mengeksekusi kode yang sesuai dengan nilai tersebut. **switch** bisa menjadi alternatif yang lebih mudah dibaca dan efisien daripada menggunakan banyak pernyataan **if else if** secara berurutan.

Sintaks **switch**

```
switch (ekspresi) {  
    case nilai1:  
        // Blok kode yang dijalankan jika ekspresi == nilai1  
        break;  
  
    case nilai2:  
        // Blok kode yang dijalankan jika ekspresi == nilai2  
        break;  
  
    // Anda dapat menambahkan lebih banyak case sesuai  
    // kebutuhan  
  
    default:  
        // Blok kode yang dijalankan jika tidak ada case yang  
        // cocok  
}
```

### 4. Contoh Program

```
public class ContohIfElse {  
    public static void main(String[] args) {  
        int x = 8;  
  
        // Contoh penggunaan if  
        if (x > 10) {  
            System.out.println("x lebih besar dari 10");  
        }  
    }  
}
```



```

    }

    // Contoh penggunaan if else
    if (x > 5) {
        System.out.println("x lebih besar dari 5");
    } else {
        System.out.println("x tidak lebih besar dari 5");
    }

    // Contoh penggunaan else if
    if (x > 10) {
        System.out.println("x lebih besar dari 10");
    } else if (x > 5) {
        System.out.println("x lebih besar dari 5, tapi tidak
lebih besar dari 10");
    } else if (x > 0) {
        System.out.println("x lebih besar dari 0, tapi tidak
lebih besar dari 5");
    } else {
        System.out.println("x kurang dari atau sama dengan 0");
    }
}

```

 **Fakultas Informatika** 

**output :**

```

PS D:\Dev\java\Modul> javac ContohIfElse.java
PS D:\Dev\java\Modul> java ContohIfElse.java
x lebih besar dari 5
x lebih besar dari 5, tapi tidak lebih besar dari 10

```

```

public class ContohSwitch {

    public static void main(String[] args) {
        int hari = 3;
        String namaHari;

        // Menggunakan switch untuk menentukan hari
        switch (hari) {
            case 1:
                namaHari = "Senin";
                break;
            case 2:
                namaHari = "Selasa";
                break;
            case 3:
                namaHari = "Rabu";
                break;
            case 4:

```

```

        namaHari = "Kamis";
        break;
    case 5:
        namaHari = "Jumat";
        break;
    case 6:
        namaHari = "Sabtu";
        break;
    case 7:
        namaHari = "Minggu";
        break;
    default:
        namaHari = "Hari tidak valid";
        break;
    }

    System.out.println("Hari ke-" + hari + " adalah " +
namaHari);
}
}

```

**Output :**

```

PS D:\Dev\java\Modul> javac ContohSwitch.java
PS D:\Dev\java\Modul> java ContohSwitch.java
Hari ke-3 adalah Rabu
PS D:\Dev\java\Modul>

```

## 5. Latihan Soal

```
> Package Name : com.percabangan
> Class Name : GradelfTest

> Isi implementasi
Mari kita tingkatkan sistem manajemen kelas yang telah kita buat sebelumnya.
100 atau kurang hingga 90 atau lebih: Nilai A.
Di bawah 90 hingga di atas 80: Nilai B.
Di bawah 80 hingga di atas 70: Nilai C.
Kurang dari 70 poin hingga lebih dari 60 poin: Nilai D.
Kurang dari 60 poin: Nilai F.

Hong Gil-Dong : 92 poin

> Hasil Run
Nilai A.
```

Gambar 5.1 Latihan Soal Percabangan 1

```
> Package Name : com.switch
> Class Name : TestMenu

> Isi implementasi
Mari kita membuat layar menu untuk meluncurkan berbagai fungsi.

> Hasil Run
=====
1. Mendaftar
2. lupa nama pengguna
3. temukan kata sandi Anda
4. End
=====
Masukkan menu yang Anda inginkan : 3
** Pengambilan kata sandi berhasil!!**
```

Gambar 5.2 Latihan Soal Percabangan 2

## Modul 6 Perulangan while do - do while

### 1. Tujuan

Pada modul ini, mahasiswa akan mempelajari dan menerapkan konsep perulangan do..while dan while dalam pemrograman. Mahasiswa diharapkan mampu memahami perbedaan dan karakteristik dari kedua jenis perulangan tersebut, termasuk kapan situasi yang tepat untuk menggunakannya. Melalui contoh-contoh kasus dan tugas-tugas yang diberikan, mahasiswa akan dilatih untuk menyelesaikan berbagai permasalahan yang memerlukan pengulangan suatu blok kode hingga kondisi tertentu terpenuhi.

### 2. While do

**while** adalah loop yang digunakan untuk mengeksekusi blok kode berulang kali selama kondisi yang diberikan bernilai benar (**true**). Loop ini akan terus berjalan hingga kondisi tersebut menjadi salah (**false**).

```
while (kondisi) {  
    // Blok kode yang akan dieksekusi selama kondisi benar  
}
```

### 3. Do While

**do-while** adalah loop yang mirip dengan **while**, tetapi dengan perbedaan utama bahwa kondisi dievaluasi setelah blok kode dieksekusi. Dengan kata lain, **do-while** akan selalu mengeksekusi blok kode setidaknya sekali, meskipun kondisi awalnya salah (**false**).

```
do {  
    // Blok kode yang akan dieksekusi  
} while (kondisi);
```

### 4. Perbandingan **while** dan **do-while**

Aspek	while	do-while
Evaluasi Kondisi	Dilakukan sebelum memasuki blok kode.	Dilakukan setelah blok kode dieksekusi.
Eksekusi Awal	Tidak dijamin akan dijalankan jika kondisi awalnya salah.	Blok kode dijalankan setidaknya sekali.

Penggunaan	Umum digunakan ketika kita ingin mengecek kondisi sebelum melakukan eksekusi.	Digunakan ketika kita ingin memastikan blok kode dieksekusi setidaknya sekali.
------------	---	--

Tabel 6.1 Perbandingan While do dan do While

## 5. Contoh Program

```
public class ContohDoWhile {

    public static void main(String[] args) {
        int i = 1;

        do {
            System.out.println("Perulangan ke-" + i);
            i++;
        } while (i <= 5);
    }
}
```

### output :

```
PS D:\Dev\java\Modul> javac ContohDoWhile.java
PS D:\Dev\java\Modul> java ContohDoWhile.java
Perulangan ke-1
Perulangan ke-2
Perulangan ke-3
Perulangan ke-4
Perulangan ke-5
```

## 6. Latihan Soal

```
> Package Name : com.perulangan
> Class Name : GuGuDanTest

> Isi implementasi
Mari kita cetak seluruh frasa seperti yang ditunjukkan di bawah ini.

> Hasil run
2 * 1 = 2, 2 * 2 = 4, 2 * 3 = 6, 2 * 4 = 8, 2 * 5 = 10, ..., 2 * 9 = 18
3 * 1 = 3, 3 * 2 = 6, 3 * 3 = 9, 3 * 4 = 12, 3 * 5 = 15, ..., 3 * 9 = 27
4 * 1 = 4, 4 * 2 = 8, 4 * 3 = 12, 4 * 4 = 16, 4 * 5 = 20, ..., 4 * 9 = 36
...
8 * 1 = 8, 8 * 2 = 16, 8 * 3 = 24, 8 * 4 = 32, 8 * 5 = 40, ..., 8 * 9 = 72
9 * 1 = 9, 9 * 2 = 18, 9 * 3 = 27, 9 * 4 = 36, 9 * 5 = 45, ..., 9 * 9 = 81
```

Gambar 6.1 Latihan Soal Perulangan While 1

```
> Package Name : com.while
> Class Name : MultipleFiveSumTest

> Isi implementasi
Mintalah siswa menjumlahkan setiap kelipatan 5 secara bergantian. Hentikan
penjumlahan kelipatan 5 jika jumlah totalnya sudah mencapai 100 atau lebih.

> Hasil Run
1. ( + 5 ) 5
2. ( + 10 ) 15
3. ( + 15 ) 30
4. ( + 20 ) 50
5. ( + 25 ) 75
6. ( + 30 ) 105
```

Gambar 6.2 Latihan Soal Perulangan While 2

```
> Package Name : com.while
> Class Name : WhilePrintTest

> Isi implementasi
Dengan menggunakan perulangan while, cetak string "Ini adalah latihan while."
10 kali.

> Hasil run
1. Ini adalah latihan while
2. Ini adalah latihan while
3. Ini adalah latihan while
...
10. Ini adalah latihan while
```

Gambar 6.3 Latihan Soal Perulangan While 3

## Modul 7 Perulangan for

### 1. Tujuan


Pada modul ini mahasiswa dapat menyelesaikan permasalahan dengan menggunakan perulangan for.

### 2. Perulangan For

**for** loop adalah struktur kontrol yang digunakan untuk menjalankan blok kode berulang kali dengan jumlah iterasi yang telah ditentukan. Loop ini biasanya digunakan ketika kita tahu berapa kali loop perlu dijalankan.

```
for (inisialisasi; kondisi; iterasi) {  
    // Blok kode yang akan dieksekusi berulang kali  
}
```

### 3. Contoh Program



```
public class ContohForLoop {  
  
    public static void main(String[] args) {  
        for (int i = 1; i <= 5; i++) {  
            System.out.println("Perulangan ke-" + i);  
        }  
    }  
}
```

#### output :

```
PS D:\Dev\java\Modul> javac ContohForLoop.java  
PS D:\Dev\java\Modul> java ContohForLoop.java  
Perulangan ke-1  
Perulangan ke-2  
Perulangan ke-3  
Perulangan ke-4  
Perulangan ke-5
```

#### 4. Latihan Soal

- **Package Name** : com.for
- **Class Name** : PrintStarTest
- **Isi implementasi**  
Mari kita keluarkan bintang (\*) seperti yang ditunjukkan dalam eksekusi di bawah ini.
- **Hasil Run**  
\*  
\*\*  
\*\*\*  
\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

Gambar 7.1 Latihan Soal Perulangan for



## Modul 8 Assessment 1

Pada modul ini, mahasiswa akan diuji pemahamannya terkait dasar-dasar pemrograman menggunakan bahasa Java. Materi yang diujikan mencakup pengenalan bahasa Java, konsep variabel dan operator, serta alur kontrol program melalui runtutan/sekuensi. Selain itu, mahasiswa akan diuji dalam menerapkan struktur percabangan seperti *if* dan *switch*, serta penggunaan berbagai jenis perulangan seperti *while*, *do-while*, dan *for*. Melalui ujian ini, diharapkan mahasiswa dapat menunjukkan pemahaman dan keterampilan mereka dalam mengimplementasikan konsep-konsep tersebut secara praktis dalam pemrograman Java.



## Modul 9 Method tanpa Nilai Kembalian

### 1. Tujuan

Pada modul ini mahasiswa diharapkan memahami konsep subprogram tanpa pengembalian nilai (void), termasuk pengertian dan penerapannya dalam pemrograman. Selain itu, mahasiswa juga diharapkan mampu membuat subprogram sederhana yang dapat digunakan untuk menjalankan tugas-tugas tertentu secara modular dalam suatu program. Melalui pemahaman dan praktik ini, mahasiswa dapat lebih menguasai cara memecah permasalahan menjadi bagian-bagian yang lebih kecil dan terstruktur.

### 2. Method

Dalam Java, **method tanpa nilai kembalian** adalah method yang tidak mengembalikan nilai apapun setelah eksekusi. Method seperti ini hanya melakukan tugas tertentu, seperti mencetak sesuatu ke layar atau mengubah nilai variabel, tanpa memberikan hasil kembali kepada pemanggilnya.

#### a. Karakteristik Method Tanpa Nilai Kembalian

- **Tipe Return void:** Method ini menggunakan kata kunci **void** sebagai tipe return-nya, yang menandakan bahwa method ini tidak akan mengembalikan nilai apapun.
- **Tidak Ada return Statement:** Tidak perlu ada **return** statement dalam method ini, kecuali untuk keluar dari method sebelum waktunya.

Berikut adalah sintaks umum untuk mendefinisikan method tanpa nilai kembalian:

```
returnType namaMethod() {  
    // Blok kode yang akan dijalankan ketika method dipanggil  
}
```

#### Penjelasan:

1. **returnType:** Untuk method tanpa nilai kembalian, gunakan **void**.
2. **namaMethod:** Nama method yang akan digunakan untuk memanggilnya.
3. **()**: Tanda kurung bisa diisi dengan parameter jika diperlukan, atau dibiarkan kosong jika tidak ada parameter.
4. **Blok Kode:** Berisi instruksi atau logika yang akan dijalankan saat method dipanggil.

### 3. Contoh Program

#### a. Method untuk Mencetak Pesan

```
public class ContohMethodVoid {  
  
    // Method tanpa nilai kembalian  
    void cetakPesan() {  
        System.out.println("Ini adalah method tanpa nilai  
kembalian.");  
    }  
  
    public static void main(String[] args) {  
        // Membuat objek dari kelas  
        ContohMethodVoid obj = new ContohMethodVoid();  
  
        // Memanggil method tanpa nilai kembalian  
        obj.cetakPesan();  
    }  
}
```

output :

```
PS D:\Dev\java\Modul> javac ContohMethodVoid.java  
PS D:\Dev\java\Modul> java ContohMethodVoid.java  
Ini adalah method tanpa nilai kembalian.
```

#### b. Method untuk Mengubah Nilai Variabel

```
public class ContohVoidParameter {  
  
    // Method tanpa nilai kembalian, dengan parameter  
    void tambahNilai(int x) {  
        x += 10;  
        System.out.println("Nilai setelah ditambah: " + x);  
    }  
  
    public static void main(String[] args) {  
        int nilai = 5;  
  
        // Membuat objek dari kelas  
        ContohVoidParameter obj = new ContohVoidParameter();  
  
        // Memanggil method tambahNilai  
        obj.tambahNilai(nilai);  
  
        // Mencetak nilai asli untuk menunjukkan tidak ada  
perubahan  
        System.out.println("Nilai asli: " + nilai);  
    }  
}
```

**output :**

```
PS D:\Dev\java\Modul> javac ContohVoidParameter.java
PS D:\Dev\java\Modul> java ContohVoidParameter.java
Nilai setelah ditambah: 15
Nilai asli: 5
```

## 5. Latihan Soal

➤ **Package Name** : com.method

➤ **Class Name** : Student

➤ **Isi implementasi**

Anda ingin membuat sistem untuk mengelola nilai siswa.

Satu kelas yang benar-benar kita butuhkan adalah kelas "Student", jadi mari kita buat kelas bernama Student.

Informasi yang dibutuhkan oleh sistem adalah nama siswa, nomor mata kuliah, nilai bahasa, nilai bahasa Inggris, dan nilai matematika.

Sistem sering kali menghitung rata-rata nilai dan mengambil informasi siswa sekaligus. Hal ini juga terjadi ketika siswa mengubah nomor mata kuliah mereka.

Gambar 9.1 Latihan Soal Method tanpa Nilai Kembalian

## Modul 10 Method dengan Kembalian

### 1. Tujuan

Mahasiswa diharapkan mampu memahami konsep subprogram dengan pengembalian nilai (return), termasuk pengertian dan peranannya dalam pengembangan program yang modular dan terstruktur. Pada bagian ini, mahasiswa akan mempelajari bagaimana sebuah subprogram atau fungsi dapat mengembalikan hasil perhitungan atau nilai tertentu setelah proses eksekusi selesai.

### 2. Method

Dalam Java, **method dengan nilai kembalian** adalah method yang setelah menyelesaikan tugasnya akan mengembalikan suatu nilai kepada pemanggilnya. Nilai yang dikembalikan ini bisa berupa tipe data primitif seperti `int`, `double`, `char`, atau tipe data objek seperti `String`, `ArrayList`, atau bahkan objek dari kelas buatan sendiri.

#### a. Karakteristik Method dengan Nilai Kembalian

- **Tipe Return:** Method ini harus mendefinisikan tipe data dari nilai yang dikembalikannya. Jika method mengembalikan nilai tipe `int`, maka tipe return-nya adalah `int`. Jika mengembalikan nilai `String`, maka tipe return-nya adalah `String`.
- **return Statement:** Method dengan nilai kembalian harus menggunakan pernyataan `return` di dalamnya untuk mengembalikan nilai tersebut ke pemanggil.

Berikut adalah sintaks umum untuk mendefinisikan method dengan nilai kembalian:

```
returnType namaMethod() {  
    // Blok kode yang akan dijalankan ketika method dipanggil  
  
    return nilai; // Mengembalikan nilai  
}
```

#### Penjelasan :

- returnType:** Tipe data dari nilai yang akan dikembalikan oleh method.
- namaMethod:** Nama method yang digunakan untuk memanggilnya.
- return nilai;** Pernyataan `return` digunakan untuk mengembalikan nilai dari method.

### 3. Contoh Program

#### a. Method untuk Mengembalikan Bilangan Bulat (int)

```
public class ContohMethodInt {  
  
    // Method dengan nilai kembalian int  
    int hitungPersegi(int sisi) {  
        int luas = sisi * sisi;  
        return luas;  
    }  
  
    public static void main(String[] args) {  
        // Membuat objek dari kelas  
        ContohMethodInt obj = new ContohMethodInt();  
  
        // Memanggil method dan menyimpan nilai kembalian  
        int hasil = obj.hitungPersegi(4);  
  
        // Menampilkan hasil  
        System.out.println("Luas persegi: " + hasil);  
    }  
}
```

#### Penjelasan:

- Method `hitungPersegi`:** Method ini menerima parameter `sisi` dan mengembalikan hasil perhitungan luas persegi (`sisi * sisi`).
- Pernyataan `return`:** Nilai `luas` dikembalikan kepada pemanggilnya.
- Panggilan Method:** Method dipanggil dengan argumen `4`, dan hasil perhitungannya disimpan dalam variabel `hasil`.

#### output :

```
PS D:\Dev\java\Modul> javac ContohMethodInt.java  
PS D:\Dev\java\Modul> java ContohMethodInt.java  
Luas persegi: 16
```

#### b. Method untuk Mengembalikan String

```
public class ContohMethodString {  
  
    // Method dengan nilai kembalian String  
    String sambutPengguna(String nama) {  
        return "Selamat datang, " + nama + "!";  
    }  
  
    public static void main(String[] args) {  
        // Membuat objek dari kelas  
        ContohMethodString obj = new ContohMethodString();  
    }  
}
```

```

// Memanggil method dan menyimpan nilai kembalian
String pesan = obj.sambutPengguna("Andi");

// Menampilkan pesan
System.out.println(pesan);
}
}

```

#### Penjelasan:

- **Method `sambutPengguna`:** Method ini menerima parameter `nama` dan mengembalikan pesan sambutan yang berupa string.
- **Pernyataan `return`:** Mengembalikan string yang telah dibangun.
- **Panggilan Method:** Method dipanggil dengan argumen `Andi`, dan hasilnya disimpan dalam variabel `pesan`.

#### output :

```

PS D:\Dev\java\Modul> javac ContohMethodString.java
PS D:\Dev\java\Modul> java ContohMethodString.java
Selamat datang, Andi!

```

#### c. Method untuk Mengembalikan Boolean

```

public class ContohMethodBoolean {

    // Method dengan nilai kembalian boolean
    boolean cekGenap(int angka) {
        return angka % 2 == 0;
    }

    public static void main(String[] args) {
        // Membuat objek dari kelas
        ContohMethodBoolean obj = new ContohMethodBoolean();

        // Memanggil method dan menyimpan nilai kembalian
        boolean hasil = obj.cekGenap(8);

        // Menampilkan hasil
        if (hasil) {
            System.out.println("Angka tersebut adalah genap.");
        } else {
            System.out.println("Angka tersebut adalah ganjil.");
        }
    }
}

```

**Penjelasan:**

- **Method `cekGenap`:** Method ini menerima parameter `angka` dan mengembalikan nilai boolean (`true` jika angka genap, `false` jika ganjil).
- **Pernyataan `return`:** Mengembalikan hasil dari ekspresi `angka % 2 == 0`.
- **Panggilan Method:** Method dipanggil dengan argumen `8`, dan hasilnya disimpan dalam variabel `hasil`.

**output :**

```
PS D:\Dev\java\Modul> javac ContohMethodBoolean.java
PS D:\Dev\java\Modul> java ContohMethodBoolean.java
Angka tersebut adalah genap.
```



## Modul 11 Assesment 2

Pada modul ini, mahasiswa akan diuji pemahamannya mengenai konsep *method* atau subprogram dalam pemrograman, baik yang tanpa nilai kembalian (*void*) maupun yang memiliki nilai kembalian. Melalui assignment ini, mahasiswa diharapkan mampu menerapkan konsep tersebut dalam menyelesaikan berbagai permasalahan yang diberikan. Mahasiswa akan belajar bagaimana merancang dan mengimplementasikan *method* sederhana yang berfungsi sebagai bagian dari program yang lebih besar, serta memahami kapan situasi yang tepat untuk menggunakan *method* dengan atau tanpa nilai kembalian. Dengan menguasai materi ini, mahasiswa akan dapat menulis program yang lebih modular, terstruktur, dan efisien.



## Modul 12 Array / Larik I (dasar konsep)

### 1. Tujuan

Pada modul ini mahasiswa diharapkan mampu memahami konsep dasar array, termasuk pengertian, fungsi, dan penerapannya dalam pemrograman. Pada bagian ini, mahasiswa akan mempelajari bagaimana array dapat digunakan untuk menyimpan dan mengelola sejumlah besar data yang memiliki tipe yang sama secara efisien. Pemahaman ini mencakup bagaimana mendeklarasikan, menginisialisasi, dan mengakses elemen-elemen array.

### 2. Array

**Array** adalah struktur data yang digunakan untuk menyimpan sekumpulan elemen dengan tipe data yang sama. Array menyimpan elemen-elemen tersebut dalam memori yang bersebelahan, dan setiap elemen diidentifikasi dengan indeks yang dimulai dari 0.

### 3. Karakteristik Array

1. **Ukuran Tetap:** Ukuran array ditentukan saat array dibuat dan tidak bisa diubah setelah itu.
2. **Tipe Data yang Sama:** Semua elemen dalam array harus memiliki tipe data yang sama.
3. **Akses Elemen:** Setiap elemen dalam array dapat diakses menggunakan indeksnya.

### 4. Deklarasi

Untuk mendeklarasikan array di Java, Anda perlu menentukan tipe data dari elemen yang akan disimpan dalam array, diikuti dengan tanda kurung siku `[]`.

```
int[] angka; // Deklarasi array untuk menyimpan bilangan bulat
String[] kata; // Deklarasi array untuk menyimpan string
```

### 5. Inisialisasi

Setelah deklarasi, Anda bisa menginisialisasi array dengan menentukan ukuran atau langsung mengisinya dengan elemen-elemen tertentu.

Inisialisasi dengan Ukuran:

```
// Array dengan 5 elemen, semua diinisialisasi ke 0
int[] angka = new int[5];
```

Inisialisasi dengan Nilai:

```
// Array dengan elemen 1, 2, 3, 4, 5
int[] angka = {1, 2, 3, 4, 5};
```

## 6. Mengakses Elemen Array

Anda dapat mengakses elemen array dengan menggunakan indeks. Indeks array di Java dimulai dari 0.

```
int[] angka = {1, 2, 3, 4, 5};

// Mengakses elemen pertama
int pertama = angka[0]; // Nilai pertama adalah 1

// Mengubah elemen ke-3
angka[2] = 10; // Sekarang array menjadi {1, 2, 10, 4, 5}
```

## 7. Looping Melalui Array

Array sering diakses menggunakan loop untuk memproses setiap elemen. Loop **for** dan **for-each** sering digunakan untuk ini.

### a. Menggunakan Loop **for**:

```
int[] angka = {1, 2, 3, 4, 5};

for (int i = 0; i < angka.length; i++) {
    System.out.println("Elemen ke-" + i + ": " + angka[i]);
}
```

### b. Menggunakan Loop **for-each**:

```
int[] angka = {1, 2, 3, 4, 5};

for (int num : angka) {
    System.out.println("Elemen: " + num);
}
```

## 8. Contoh Program

### a. ArrayEx

```
// Buat array untuk menyimpan angka lotto dan array untuk
// menyimpan nama 10 murid.
public class ArrayCreationEx {
    public static void main(String[] args) {

        int[] lottoNumbers;
        String[] names;

        lottoNumbers = new int[6];
        names = new String[10];
    }
}
```

```

        System.out.println(lottoNumbers[0]);
        System.out.println(lottoNumbers[1]);
        System.out.println(names[5]);
        System.out.println(names[9]);
    }
}

```

**Output :**

```

PS D:\Dev\java\Modul> javac ArrayCreationEx.java
PS D:\Dev\java\Modul> java ArrayCreationEx.java
0
0
null
null

```

**b. Menghitung Rata-Rata**

```

public class RataRataArray {

    public static void main(String[] args) {
        int[] nilai = { 70, 85, 90, 75, 80 };
        int total = 0;

        for (int i = 0; i < nilai.length; i++) {
            total += nilai[i];
        }

        double rataRata = (double) total / nilai.length;
        System.out.println("Rata-rata nilai: " + rataRata);
    }
}

```

**Penjelasan:**

- **Menghitung Total:** Setiap elemen dalam array **nilai** dijumlahkan ke dalam variabel **total**.
- **Menghitung Rata-Rata:** Total dibagi dengan jumlah elemen dalam array untuk mendapatkan rata-rata.

**Output :**

```

PS D:\Dev\java\Modul> javac RataRataArray.java
PS D:\Dev\java\Modul> java RataRataArray.java
Rata-rata nilai: 80.0

```

## 9. Latihan Soal

- **Nama Package** : com.Array
- **Nama Class** : ScoreArrayTest
- **Rincian Implementasi**  
Pengelolaan nilai Bahasa Korea, Bahasa Inggris, dan Matematika  
Sebelumnya, setiap nilai menggunakan variabel integer yang dikelola masing - masing seperti int korean, int english, int math  
Sekarang, kita akan menggunakan array dan menggunakan 1 variabel saja
- **Hasil Run**  
Bahasa Korea : 0  
Bahasa Inggris : 0  
Matematika : 0

Gambar 12.1 Latihan Soal Array 1

- **Nama Package** : com.Array
- **Nama Class** : ScoreArrayTest
- **Rincian Implementasi**  
Masukkan nilai Bahasa Korea, Bahasa Inggris, dan Matematika yang sesungguhnya ke dalam program array yang sudah di buat, lalu tampilkan hasilnya  
Nilai yang didapat adalah nilai Bahasa Korea 93, nilai Bahasa Inggris 87, nilai Matematika adalah 90  
Simpan nilai masing - masing yang sudah dibuat di array, lalu cari tau juga nilai rata - ratanya
- **Hasil Run**  
Bahasa Korea : 93  
Bahasa Inggris : 87  
Matematika : 90  
Rata - rata : 90.0

Gambar 12.2 Latihan Soal Array 2

## Modul 13 Pencarian Nilai Ekstrim

### 1. Tujuan

Mahasiswa diharapkan mampu memahami konsep pencarian nilai ekstrem dalam suatu kumpulan data, baik nilai minimum maupun maksimum. Pada bagian ini, mahasiswa akan mempelajari bagaimana menerapkan algoritma pencarian nilai ekstrem secara efisien dalam program, termasuk mengidentifikasi dan mengambil nilai terkecil atau terbesar dari sekumpulan data yang diolah

### 2. Find Min Max

**Nilai Minimum:** Elemen terkecil dalam array.

**Nilai Maksimum:** Elemen terbesar dalam array.

Mencari nilai minimum dan maksimum dari array sering digunakan dalam berbagai aplikasi, seperti statistik, pengolahan data, atau analisis dataset.

#### a. Algoritma untuk Menemukan Nilai Minimum dan Maksimum

Untuk menemukan nilai minimum dan maksimum dalam array:

1. Anggap bahwa elemen pertama sebagai nilai minimum dan maksimum sementara.
2. Iterasi setiap elemen dalam array:
  - o Jika elemen saat ini lebih kecil dari nilai minimum sementara, perbarui nilai minimum.
  - o Jika elemen saat ini lebih besar dari nilai maksimum sementara, perbarui nilai maksimum.

#### b. Langkah-Langkah Implementasi

- Inisialisasi nilai minimum dan maksimum sementara dengan elemen pertama dari array.
- Loop melalui array mulai dari elemen kedua hingga elemen terakhir.
- Bandingkan setiap elemen dengan nilai minimum dan maksimum sementara:
- Jika lebih kecil, perbarui nilai minimum.
- Jika lebih besar, perbarui nilai maksimum.

### 3. Contoh Program

```
public class FindMinMax {  
  
    public static void main(String[] args) {  
        int[] angka = { 12, 3, 5, 7, 19, 1, 25, 17 };  
  
        // Inisialisasi nilai minimum dan maksimum sementara  
        int min = angka[0];
```

```

int max = angka[0];

// Loop melalui array untuk mencari nilai min dan max
for (int i = 1; i < angka.length; i++) {
    if (angka[i] < min) {
        min = angka[i]; // Perbarui nilai minimum
    }
    if (angka[i] > max) {
        max = angka[i]; // Perbarui nilai maksimum
    }
}

// Menampilkan nilai minimum dan maksimum
System.out.println("Nilai Minimum: " + min);
System.out.println("Nilai Maksimum: " + max);
}
}

```

#### Penjelasan:

- **Inisialisasi Awal:** Nilai **min** dan **max** diinisialisasi dengan elemen pertama dari array.
- **Looping:** Array diiterasi dari indeks kedua (**i = 1**) hingga akhir.
- **Kondisi:** Jika elemen saat ini lebih kecil dari **min**, maka **min** diperbarui. Jika lebih besar dari **max**, maka **max** diperbarui.

#### Output :

```

PS D:\Dev\java\Modul> javac FindMinMax.java
PS D:\Dev\java\Modul> java FindMinMax.java
Nilai Minimum: 1
Nilai Maksimum: 25

```

## 4. Latihan Soal

```

> Nama Package : com.lgcns.chapter08
> Nama Class : MaxMinTest

> Rincian Implementasi
Setelah memasukkan data ke dalam array, carilah nilai maksimum dan minimum
di antara nilai yang disimpan dalam array.

> Hasil Run
Data array : 12, 53, 24, 10, 22, 38, 64, 39
Nilai maksimum : 64
Nilai minimum : 10

```

Gambar 13.1 Latihan Soal Pencarian Nilai Ekstrim

## Modul 14 Review Materi

Pada modul ini, mahasiswa akan melakukan tinjauan ulang terhadap materi-materi penting yang telah dipelajari selama perkuliahan. Topik yang akan direview meliputi pengenalan bahasa Java, penggunaan variabel dan operator, serta konsep runtutan/sekuensi dalam pemrograman. Mahasiswa juga akan mengkaji kembali penerapan struktur percabangan seperti *if* dan *switch*, serta perulangan seperti *while*, *do-while*, dan *for*. Selain itu, mahasiswa akan merefleksikan pemahaman mereka terkait penggunaan *method* tanpa nilai kembalian (*void*) dan *method* dengan nilai kembalian. Melalui review ini, mahasiswa diharapkan mampu memperkuat penguasaan konsep-konsep dasar pemrograman Java yang telah dipelajari, memantapkan logika pemrograman, serta siap menghadapi tugas atau ujian yang membutuhkan pemahaman komprehensif terhadap materi-materi tersebut.





## Modul 15 Assesment 3

Pada modul ini, mahasiswa akan diuji mengenai pemahaman dan penerapan konsep array serta teknik pencarian nilai ekstrem. Ujian ini mencakup tugas-tugas yang memerlukan mahasiswa untuk mendemonstrasikan keterampilan dalam mendefinisikan, menginisialisasi, dan memanipulasi array, serta menerapkan algoritma untuk menemukan nilai minimum dan maksimum dalam kumpulan data. Mahasiswa diharapkan dapat menyelesaikan berbagai jenis soal yang menguji kemampuan mereka dalam menggunakan array secara efisien dan mengembangkan solusi yang efektif untuk pencarian nilai ekstrem. Ujian ini bertujuan untuk mengevaluasi kemampuan mahasiswa dalam mengimplementasikan konsep-konsep tersebut dalam program, serta memastikan mereka dapat menyusun kode yang terstruktur dan berfungsi dengan baik untuk menyelesaikan permasalahan berbasis data.



## Modul 16 Praktikum Susulan

Pada modul ini, mahasiswa akan mengikuti praktikum susulan yang dirancang khusus untuk menggantikan sesi praktikum yang terlewat sebelumnya. Praktikum ini disediakan bagi mahasiswa yang berhalangan hadir dengan alasan yang jelas dan valid. Melalui praktikum susulan ini, mahasiswa akan memiliki kesempatan untuk menyelesaikan tugas-tugas yang sebelumnya terlewat, memastikan bahwa mereka tetap dapat mengejar ketertinggalan materi dan keterampilan praktis yang diperlukan. Modul ini bertujuan untuk memastikan bahwa setiap mahasiswa memiliki kesempatan yang adil untuk memenuhi persyaratan praktikum dan mencapai kompetensi yang diharapkan meskipun ada kendala kehadiran sebelumnya.



## DAFTAR PUSTAKA

- Darmawan, E. (2014). Pemrograman Dasar C-Java-C#. Informatika.
- Goodrich, M. T. (2002). Data Structures and Algorithms in Java. Science.
- LG CNS. (2024). LG CNS Java C# Oracle Fusion Institute. In LG CNS. LG CNS.
- Qhesta, S. (2022). Introduction to Java Programming and Data Structure.
- Sianipar, R. (2014). Soal, Kasus dan Penyelesaian Pemrograman Java. Informatika.
- Y. Daniel Liang. (2011). Introduction to Java Programming and Data Structures - Comprehensive Version. In Journal of Physics A: Mathematical and Theoretical (Vol. 44, Issue 8).
- Yuniansyah. (2020). Algoritma Dan Pemrograman Menggunakan Bahasa Pemograman Jawa (Teori Dan Aplikasinya). In Penerbit Lindan Bestari.
- (Darmawan, 2014; Goodrich, 2002; LG CNS, 2024; Qhesta, 2022; Sianipar, 2014; Y. Daniel Liang, 2011; Yuniansyah, 2020)

