

Fadhel Akhmad Hizham, S.Kom, M.Kom

Modul Praktikum

ALGORITMA PEMROGRAMAN

MODUL PRAKTIKUM
ALGORITMA PEMROGRAMAN

© 2024, Fadhel Akhmad Hizham

Cetakan Pertama, November 2024

vi + 184 hlm; 15,5 x 23 cm

Penulis: Fadhel Akhmad Hizham

Desain Sampul: Azyan Mitra Media

Tata Letak Isi: Moh. Mursyid

Diterbitkan Oleh:



WIDYA GAMA PRESS

ANGGOTA ASOSIASI PENERBIT PERGURUAN TINGGI
INDONESIA (APPTI)

Office:

Institut Teknologi dan Bisnis (ITB) Widya Gama Lumajang

Jl. Gatot Subroto No. 4, Karangsari, Kec. Sukodono,

Kabupaten Lumajang, Jawa Timur.

Telp. (0334) 881924

PRAKATA

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa atas terselesaikannya buku *Modul Praktikum Algoritma Pemrograman*. Buku ini disusun sebagai panduan praktikum bagi mahasiswa atau pemula yang ingin mempelajari algoritma dan pemrograman secara sistematis dan mendalam, mulai dari konsep dasar hingga implementasi praktis. Kami berharap, modul ini dapat menjadi referensi yang berguna dalam membantu pembaca menguasai dasar-dasar pemrograman melalui latihan-latihan yang terstruktur dan aplikatif.

Dalam modul ini, kami menyusun materi secara berurutan agar mudah dipahami dan diterapkan. Bab pertama dan kedua membahas pendahuluan serta persiapan praktikum, yang dirancang untuk memperkenalkan pembaca pada konsep dan alat-alat dasar yang diperlukan. Selanjutnya, materi-materi pokok mengenai tipe data, variabel, input dan output, operator, percabangan, dan perulangan dijelaskan secara terperinci. Topik ini diuraikan dengan contoh-contoh yang relevan dan latihan praktikum, sehingga pembaca dapat memahami tidak hanya konsep, tetapi juga penerapannya dalam pemrograman.

Bagian akhir buku ini mencakup materi lanjutan seperti array satu dimensi dan multi dimensi, yang akan membantu pembaca memahami pengelolaan data secara lebih kompleks. Kami juga menyertakan *Proyek Akhir* untuk menguji pemahaman pembaca secara keseluruhan melalui aplikasi yang lebih mendalam dan menyeluruh dari materi yang dipelajari.

Kami menyadari bahwa penyusunan buku ini masih jauh dari sempurna. Oleh karena itu, kami sangat mengharapkan saran dan masukan yang konstruktif dari para pembaca untuk perbaikan di masa yang akan datang. Semoga buku ini bermanfaat dan dapat memotivasi para pembaca untuk terus belajar dan berkarya di bidang pemrograman.

Selamat belajar dan selamat berpraktikum!

Penulis

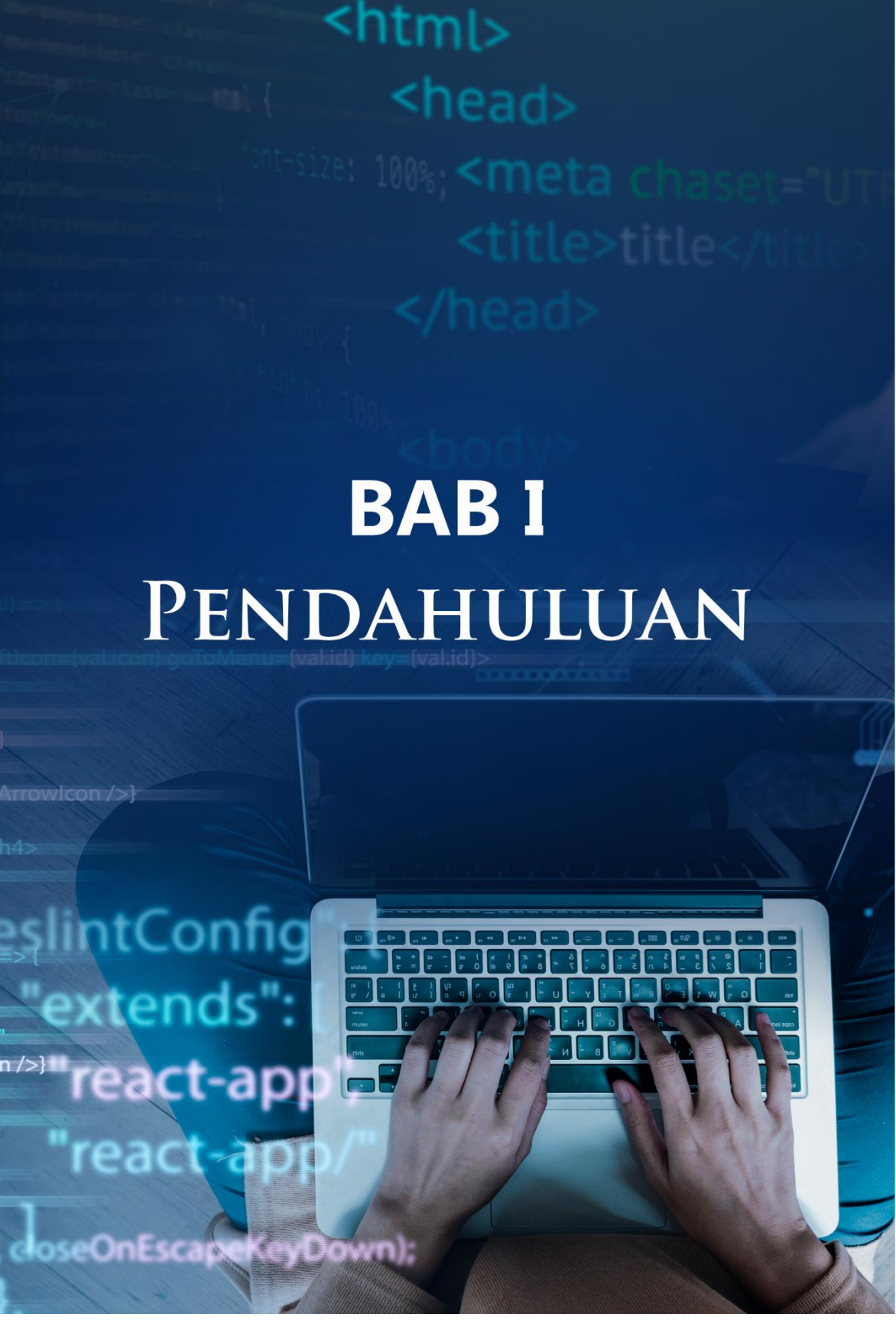
DAFTAR ISI

PRAKATA	iii
DAFTA ISI.....	v
BAB 1 PENDAHULUAN.....	1
BAB 2 PERSIAPAN PRAKTIKUM	12
BAB 3 TIPE DATA DAN VARIABEL	23
BAB 4 DEKLARASI INPUT DAN OUTPUT	44
BAB 5 FUNGSI OPERATOR.....	58
BAB 6 FUNGSI PERCABANGAN (Bagian 1).....	79
BAB 7 FUNGSI PERCABANGAN (Bagian 2).....	85
BAB 8 FUNGSI PERULANGAN (Bagian 1)	121
BAB 9 FUNGSI PERULANGAN (Bagian 2)	125
BAB 10 ARRAY (Bagian 1) 1 DIMENSI.....	145
BAB 11 ARRAY (Bagian 2) MULTI DIMENSI.....	152
BAB 12 PROYEK AKHIR	177
REFERENSI.....	183



BAB I

PENDAHULUAN



1.1. Pengenalan Algoritma

Sebelum mempelajari mengenai pemrograman, adakalanya perlu pemahaman mengenai algoritma. Algoritma dapat didefinisikan sebagai **solusi** yang berupa susunan yang diurutkan berdasarkan sistematika tertentu untuk memecahkan masalah. Ketika berbicara tentang algoritma di bidang pemrograman, maka yang dimaksud adalah solusi untuk memecahkan suatu masalah yang harus diselesaikan dengan menggunakan bahasa pemrograman.

Ciri-ciri dari suatu algoritma diantaranya:

- ***Finitless* (keterbatasan)**

Algoritma harus berhenti setelah menjalankan sejumlah langkah terbatas apabila tujuan akhir tercapai. Jika tidak berhenti, maka terindikasi ada algoritma yang salah.

- ***Definitless* (kepastian)**

Setiap langkah pengerjaan harus didefinisikan secara tepat dan tidak bermakna ganda atau ambigu, sehingga tidak terjadi kesalahan dalam menghasilkan keluaran.

- ***Input* (masukan)**

Suatu permasalahan yang dihadapi dan akan dicarikan solusinya. Algoritma ini mempunyai nol atau lebih input, yaitu besaran yang diberikan pada algoritma untuk diproses.

- ***Output* (keluaran)**

Algoritma memiliki nol atau lebih output dan harus berisi penyelesaian atau solusi dari suatu masalah.

- ***Effectiveness* (keefektifan)**

Setiap tahapan algoritma harus sesederhana mungkin, sehingga dapat dikerjakan dalam waktu yang masuk akal.

Perbedaan antara algoritma dengan program:

- **Desain**

Algoritma merupakan suatu desain yang dibuat sebagai langkah awal sebelum membangun mesin dalam rekayasa perangkat lunak. Sementara program merupakan implementasi kode dari desain yang telah dibuat.

- **Penulisan**

Untuk menulis algoritma diperlukan seorang dengan pengetahuan analisis pemecahan masalah untuk membuat desain. Sementara untuk menulis kode program diperlukan seorang *programmer*.

- **Hardware dan Sistem Operasi**

Untuk membuat algoritma tidak bergantung pada *hardware* dan sistem operasi, karena pada dasarnya hanya berupa rancangan pemecahan masalah saja. Sementara untuk membuat program sangat bergantung pada *hardware* dan sistem operasi.

- **Bahasa**

Algoritma dapat ditulis dengan bahasa sehari-hari, seperti Bahasa Indonesia, Bahasa Inggris, Bahasa Spanyol, dan sebagainya. Sementara program hanya dapat ditulis menggunakan bahasa khusus pemrograman, seperti C, C++, *Pascal*, *Java*, *Python*, dan sebagainya.

- **Memeriksa bug**

Algoritma menggunakan teknik analisis untuk memeriksa bug, Sementara program menggunakan teknik pengujian untuk memeriksa bug.

1.2. Memetakan Algoritma dalam Bahasa Pemrograman

Bahasa algoritmik (*pseudo-code*) merupakan bahasa penengah antara manusia dan komputer. Bahasa algoritmik dibuat untuk memudahkan algoritma dengan bahasa manusia diterjemahkan ke dalam bahasa pemrograman apapun yang dimengerti komputer. Setiap bahasa pemrograman memiliki sintaks atau cara menyatakan dalam bentuk bahasa komputer yang berbeda-beda, sesuai dengan bahasa pemrograman yang digunakan. Untuk menyelesaikan masalah ke dalam bahasa algoritmik, diperlukan 4 tahapan algoritma, yaitu:

Tabel 1.1. Tahapan Algoritma

No	Tahapan Algoritma	Keterangan
1	Deklarasi	Mendeklarasikan wadah kosong yang dipakai sebagai tempat untuk menyimpan sesuatu.
2	Inisialisasi	Menyiapkan proses yang akan dikerjakan untuk menyelesaikan masalah.
3	Proses	Tahapan menyelesaikan masalah untuk mencapai tujuan dari sebuah algoritma dibuat.
4	Finalisasi	Tahap akhir dari sebuah algoritma.

Sebagai contoh, untuk mengupas buah apel, kemudian menyajikan apel tersebut. Maka tahapan algoritma yang harus dikerjakan yaitu:

Tabel 1.2. Tahapan Algoritma Mengupas Buah Apel

No	Tahapan Algoritma	Keterangan
1	Deklarasi	Mendeklarasikan wadah kosong yang dipakai sebagai tempat untuk menyimpan apel dan pisau.
2	Inisialisasi	Mengisi wadah-wadah kosong yang akan dipergunakan

3	Proses	<ul style="list-style-type: none"> • Menuliskan ke layar mulai mengupas buah apel. • Memproses buah apel yang sedang dikupas dengan pisau.
4	Finalisasi	<ul style="list-style-type: none"> • Pisau selesai digunakan. • Menulis ke layar apel sudah dikupas dan siap untuk dihidangkan.

Agar dapat dimengerti oleh komputer, maka perlu menerjemahkan bahasa manusia ke dalam bahasa algoritmik. Penerjemahan ditunjukkan pada tabel berikut:

Tabel 1.3. Bahasa Algoritmik Mengupas Buah Apel

Tahapan	Bahasa Manusia	Bahasa Algoritmik
Deklarasi	Mendeklarasikan wadah kosong yang dipakai sebagai tempat untuk menyimpan apel dan pisau.	apel : integer pisau : integer
Inisialisasi	Mengisi wadah-wadah kosong yang akan dipergunakan	apel <- 1 pisau <- 1
Proses	Menuliskan ke layar mulai mengupas buah apel.	output ("mulai mengupas buah apel")
	Memproses buah apel yang sedang dikupas dengan pisau.	apel = apel + pisau
Finalisasi	Pisau selesai digunakan.	pisau <- 0
	Menulis ke layar apel sudah dikupas dan siap untuk dihidangkan.	output ("apel sudah dikupas dan siap untuk dihidangkan")








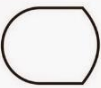








Contoh lain, yaitu algoritma untuk menghitung luas persegi, jika diketahui panjang sisinya 30 cm. Maka tahapan algoritmanya yaitu:

Tabel 1.4. Bahasa Algoritmik Menghitung Luas Persegi

Tahapan	Bahasa Manusia	Bahasa Algoritmik
Deklarasi	Mendeklarasikan wadah kosong yang dipakai sebagai tempat untuk menyimpan panjang sisi dan luas persegi.	sisi : double luas : double
Inisialisasi	Mengisi wadah-wadah kosong yang akan dipergunakan	sisi <- 30
Proses	Memproses perhitungan luas persegi.	luas = sisi * sisi
Finalisasi	Luas persegi berhasil ditemukan.	luas <- 900
	Menulis ke layar luas persegi panjang adalah (besar luasnya).	output ("Luas persegi panjang adalah" + luas)

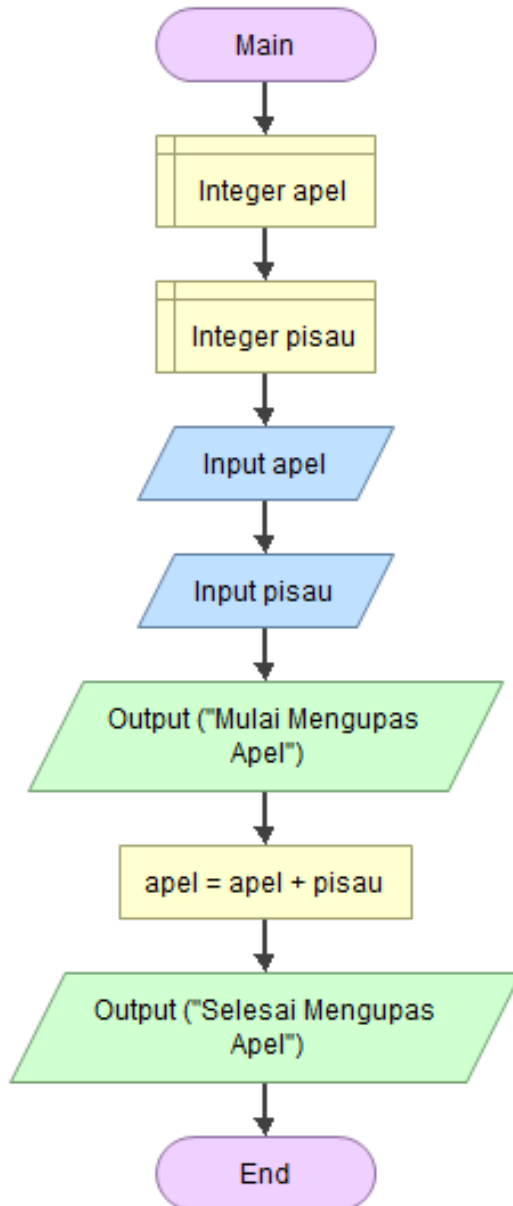
1.3. Flowchart

Flowchart berasal dari kata *flow* yang berarti aliran dan *chart* yang berarti bagan. Maka, flowchart dapat didefinisikan sebagai sebuah bagan aliran dari sesuatu, dan sesuatu itu dapat juga berupa aliran proses. Terdapat beberapa simbol dalam *flowchart*, diantaranya:

	Flow Direction symbol Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line.		Simbol Manual Input Simbol untuk pemasukan data secara manual on-line keyboard
	Terminator Symbol Yaitu simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan		Simbol Preparation Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage.
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses dalam lembar / halaman yang sama.		Simbol Predefine Proses Simbol untuk pelaksanaan suatu bagian (sub-program)/prosedure
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses pada lembar / halaman yang berbeda.		Simbol Display Simbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, printer dan sebagainya.
	Processing Symbol Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer		Simbol disk and On-line Storage Simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.
	Simbol Manual Operation Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh computer		Simbol magnetik tape Unit Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik.
	Simbol Decision Simbol pemilihan proses berdasarkan kondisi yang ada.		Simbol Punch Card Simbol yang menyatakan bahwa input berasal dari kartu atau output ditulis ke kartu
	Simbol Input-Output Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya		Simbol Dokumen Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

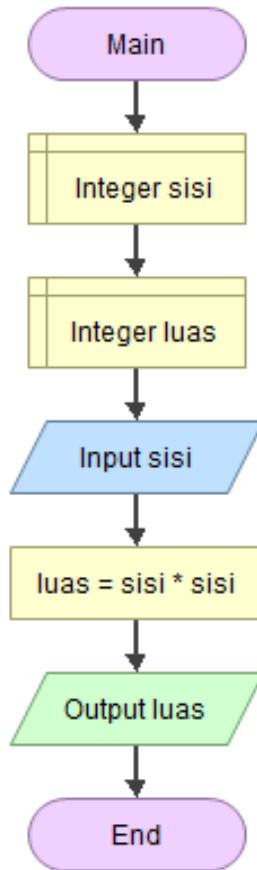
Gambar 1.1. Beberapa Simbol *Flowchart*

Sebagai contoh, untuk membuat *flowchart* yang berisi urutan mengupas buah apel. Maka, *flowchart*nya dapat digambarkan sebagai berikut.



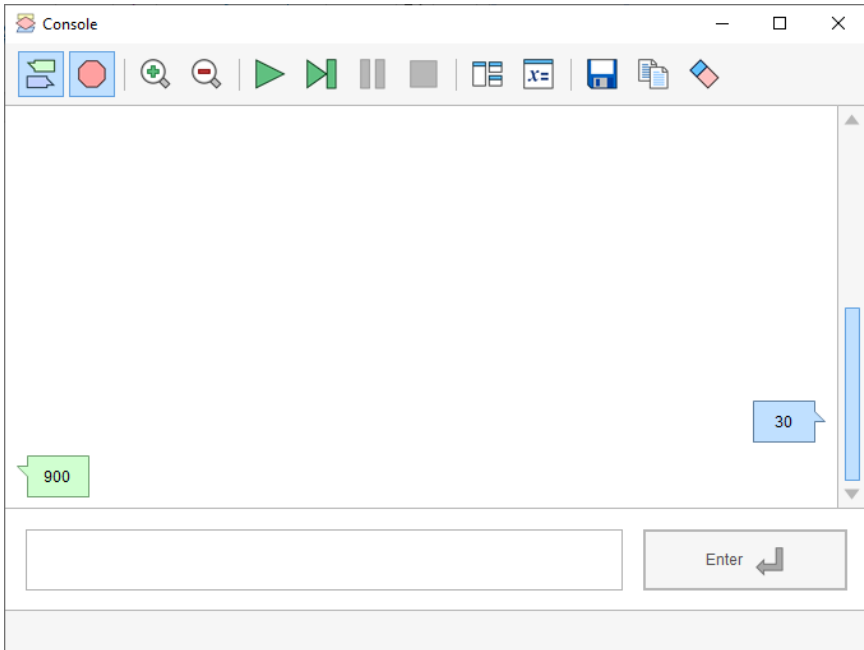
Gambar 1.2. *Flowchart* Mengupas Buah Apel

Contoh lainnya yaitu membuat *flowchart* yang berisi urutan menghitung luas persegi. Maka, *flowchart*nya dapat digambarkan sebagai berikut.



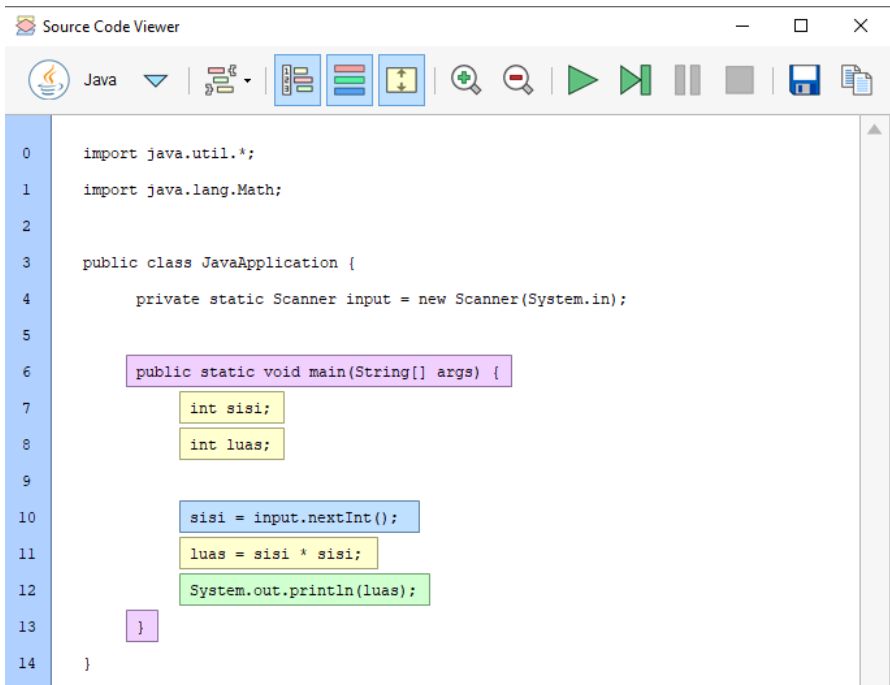
Gambar 1.3. *Flowchart* Menghitung Luas Persegi

Untuk membuat *flowchart* dari kedua contoh di atas, dapat menggunakan aplikasi *Flowgorithm*, yaitu aplikasi untuk membuat diagram alir berdasarkan permasalahan dari contoh tersebut. Di aplikasi tersebut, juga terdapat fitur *run*, untuk memasukkan nilai deklarasi dan menampilkan hasilnya sesuai dengan *flowchart* yang dibuat. Sebagai contoh, menjalankan *flowchart* menghitung luas persegi dapat ditunjukkan pada Gambar 1.4.



Gambar 1.4. Tampilan Menjalankan *Flowchart* Menghitung Luas Persegi

Dari gambar tersebut, memasukkan suatu nilai *inputan* dari variabel sisi, yaitu 30. Maka hasilnya akan muncul 900, yang berarti luas dari persegi yang sisinya 30 adalah 900, sesuai dengan rumus luas persegi, yaitu sisi * sisi. Untuk menampilkan kode programnya dalam Bahasa *Java*, dapat dilihat pada Gambar 1.5.



The image shows a 'Source Code Viewer' window with a toolbar at the top. The toolbar includes icons for Java, file operations, search, and execution. The code is as follows:

```
0 import java.util.*;
1 import java.lang.Math;
2
3 public class JavaApplication {
4     private static Scanner input = new Scanner(System.in);
5
6     public static void main(String[] args) {
7         int sisi;
8         int luas;
9
10        sisi = input.nextInt();
11        luas = sisi * sisi;
12        System.out.println(luas);
13    }
14 }
```

Gambar 1.5. Kode Program *Java* Menghitung Luas Persegi

The image features a person's hands typing on a laptop keyboard, which is the central focus. The background is a dark blue gradient with various code snippets in a lighter blue font, including HTML tags like <html>, <head>, <meta charset="UTF-8">, <title>title</title>, </head>, <body>, and JavaScript code like ArrowIcon />, extends: { "react-app", "react-app/" }, and closeOnEscapeKeyDown);. The overall aesthetic is modern and tech-oriented.

BAB 2

PERSIAPAN

PRAKTIKUM

2.1. Persiapan Praktikum

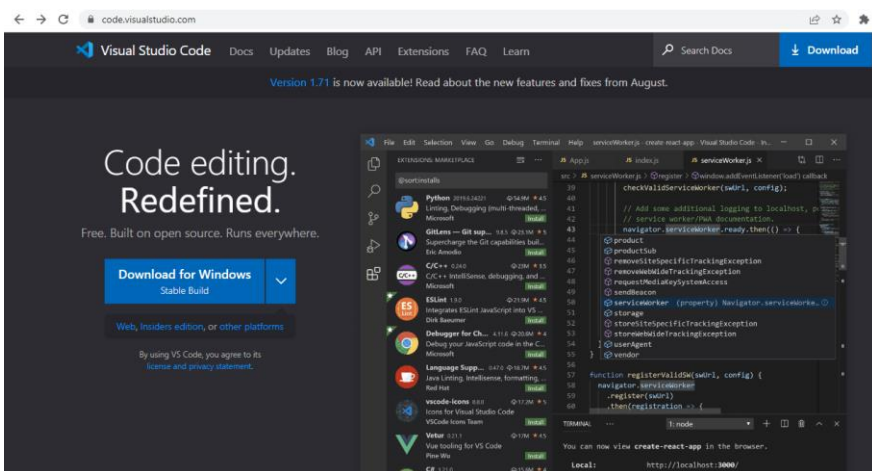
Untuk mengikuti serangkaian perkuliahan Praktikum Algoritma Pemrograman, diperlukan persiapan-persiapan, diantaranya:

- Perangkat Lunak (*Software*), berupa JDK (*Java Development Kit*) dan *Code Editor* (*Netbeans* atau *Visual Studio Code*)
- Perangkat Keras (*Hardware*), seperti laptop atau PC, dan proyektor.

Code editor yang lebih direkomendasikan yaitu menggunakan *Visual Studio Code*. Selain aplikasi tersebut lebih ringan dari *Netbeans*, aplikasi tersebut mendukung berbagai bahasa pemrograman apapun dan tersedia di *Extensions* untuk instalasi. Selain itu, *Visual Studio Code* juga tersedia di *multi-device*, baik di PC maupun *mobile*.

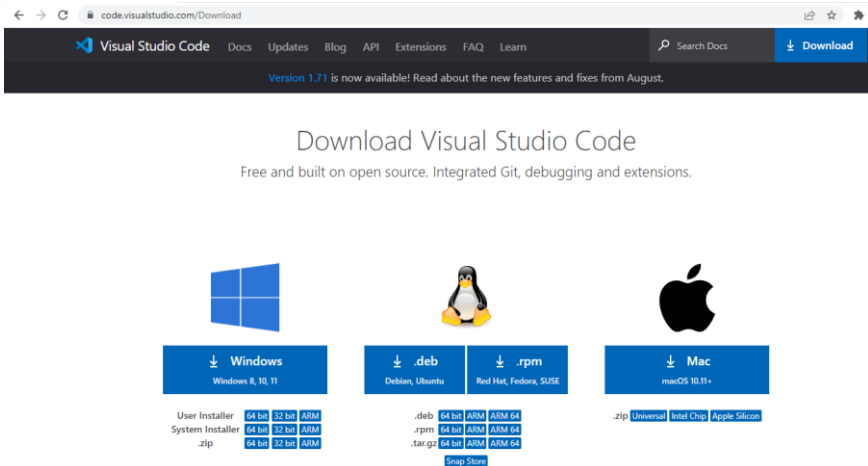
2.2. Instalasi Visual Studio Code

Langkah 1: Menuju tautan resmi dari *Visual Studio Code*, lalu klik *Download*.



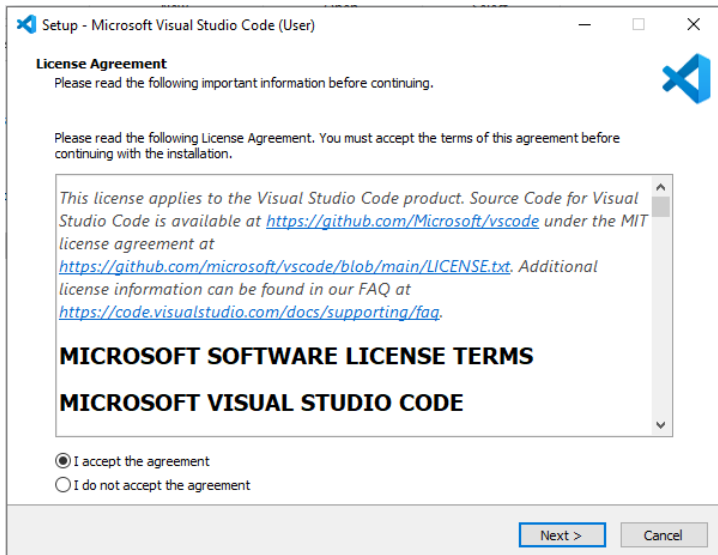
Gambar 2.1. Tautan Resmi *Visual Studio Code*

Langkah 2: Pilih sistem operasi yang digunakan lalu silakan tunggu hingga proses unduh selesai.

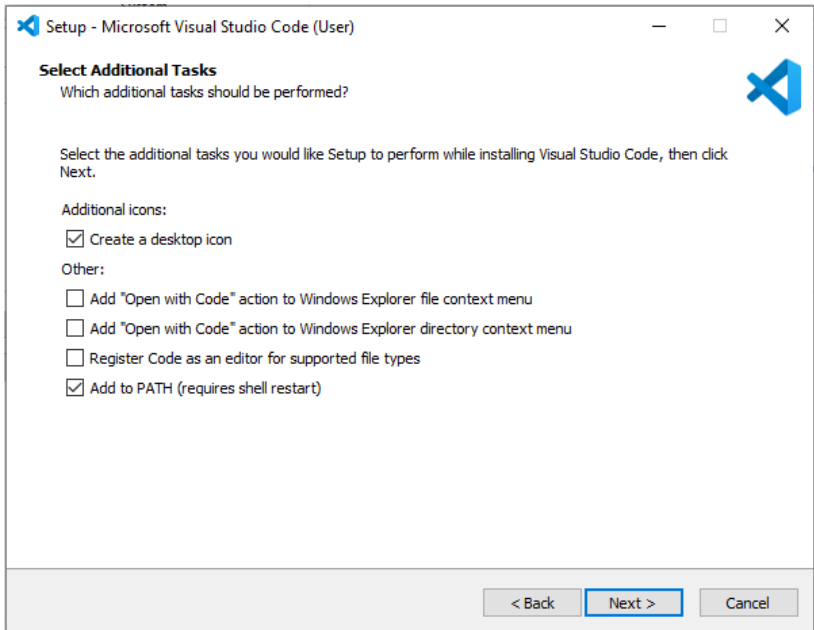


Gambar 2.2. Memilih Sistem Operasi untuk Mengunduh

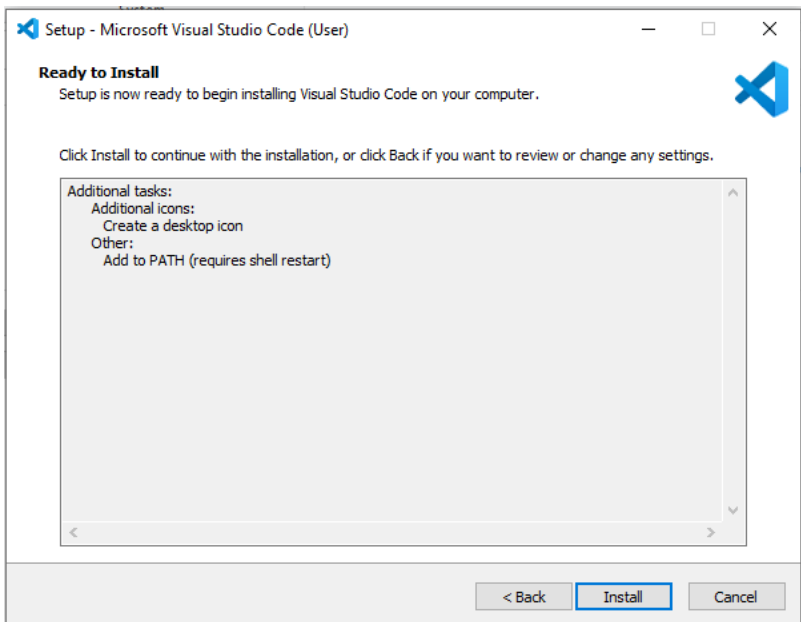
Langkah 3: Jalankan *setup*, lalu ikuti langkah-langkah instalasi hingga selesai.



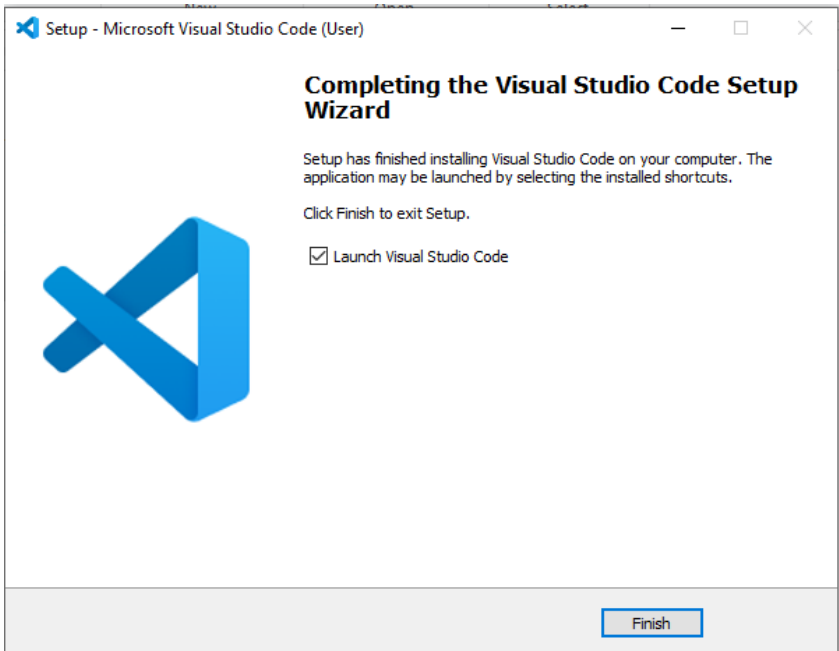
Gambar 2.3. Persetujuan Lisensi



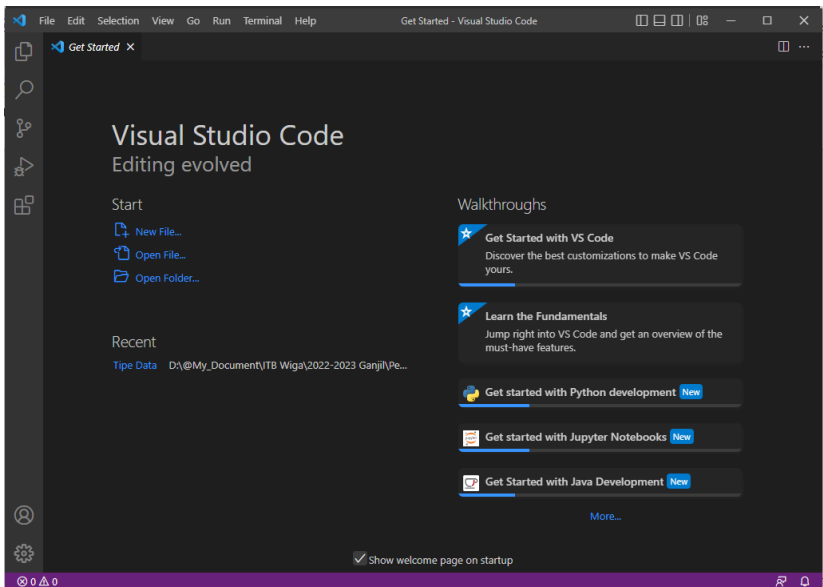
Gambar 2.4. Langkah Opsional Apabila Hendak dibuat Desktop



Gambar 2.5. Menuju Proses Instalasi



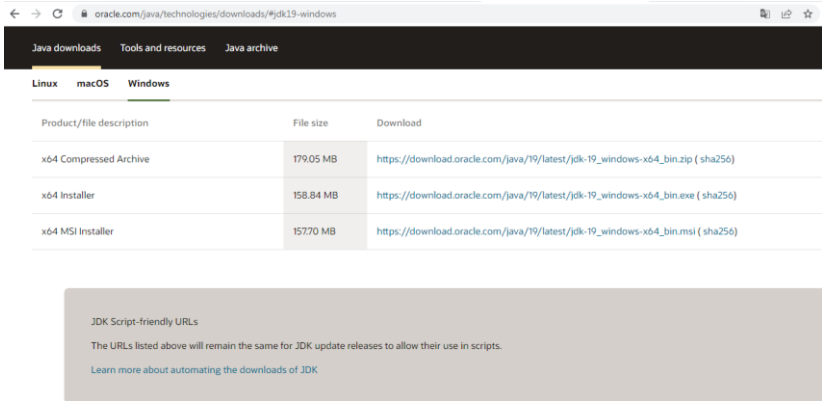
Gambar 2.6. Instalasi Selesai



Gambar 2.7. Tampilan Halaman Utama VSCode

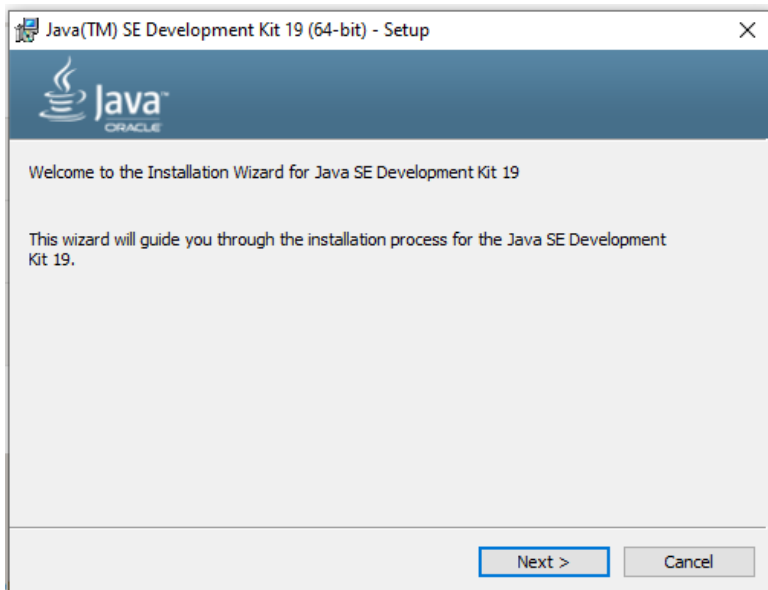
2.3. Instalasi JDK (*Java Development Kit*)

Langkah 1: Unduh JDK dari situs resmi



Gambar 2.8. Situs Resmi Penyedia JDK

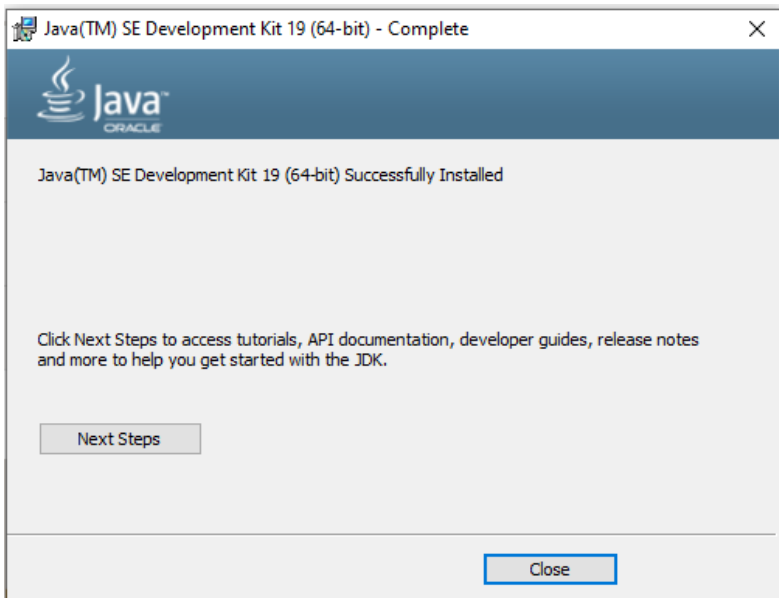
Langkah 2: Buka *setup*, lalu lakukan langkah-langkah instalasi hingga selesai



Gambar 2.9. Halaman Awal Instalasi JDK



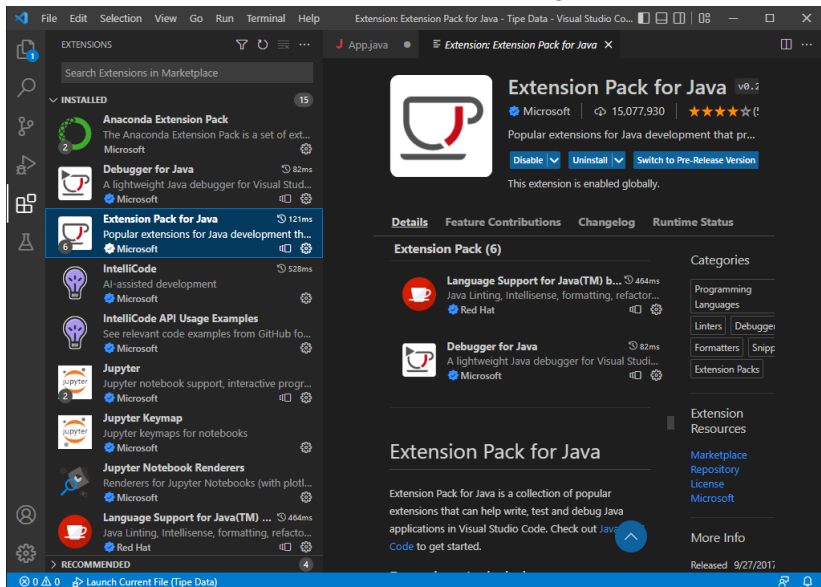
Gambar 2.10. Proses Instalasi JDK



Gambar 2.11. JDK Selesai Di-install

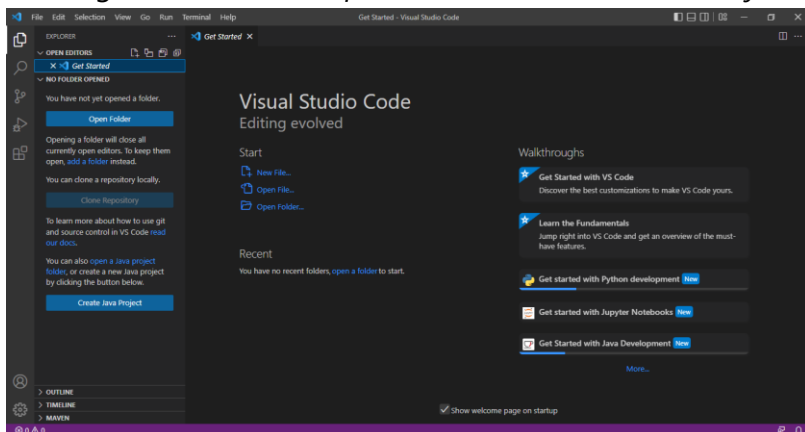
2.4. Membuat Proyek Java Baru di VSCode

Langkah 1: Buka aplikasi VSCode, lalu periksa pada tab *Extensions*, apakah "*Extensions Pack for Java*" telah terinstall atau belum. Jika belum lakukan instalasi terlebih dahulu. Jika telah ter-*install*, maka akan muncul tampilan sebagai berikut.



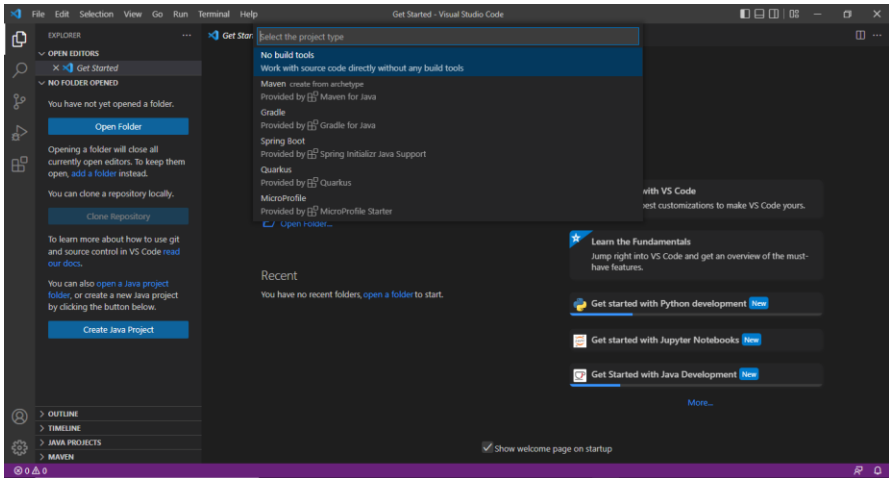
Gambar 2.12. Tampilan Ekstensi Java yang Telah Ter-*install*

Langkah 2: Buka tab *Explorer*, lalu klik "*Create Java Project*".



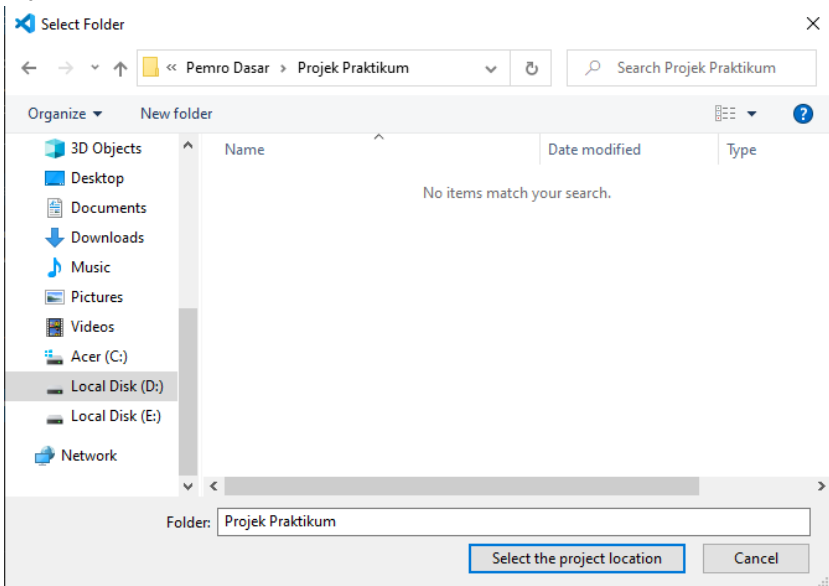
Gambar 2.13. Tampilan Tab *Explorer*

Langkah 3: Pilih menu "No build tools".

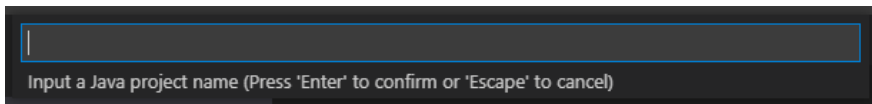


Gambar 2.14. Pilih Menu "No build tools"

Langkah 4: Pilih folder sebagai lokasi proyek, lalu beri nama proyek tersebut.



Gambar 2.15. Pilih Folder Lokasi Proyek



Gambar 2.16. Memberi Nama Proyek

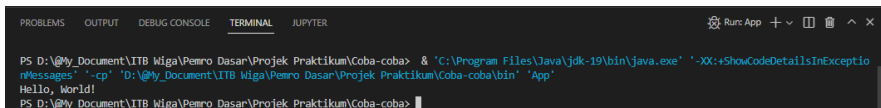
Langkah 5: Untuk membuka kode program Java, klik *src* lalu masuk ke file *App.java*, maka akan muncul tampilan kode program berikut.



Gambar 2.17. Tampilan Kode Program Java

Dari gambar diatas, berisi kode program yang ingin menampilkan kata "Hello, World!" ke layar. Untuk memodifikasi kode-kode program pada Java, dimulai dari tanda kurung kurawal terbuka setelah "public static void main" pada baris 2 sampai kurung kurawal tertutup pada baris 4.

Langkah 6: Untuk menampilkan hasil keluaran di layar, tekan *Ctrl + F5*, lalu hasil keluaran akan muncul pada tab Terminal.



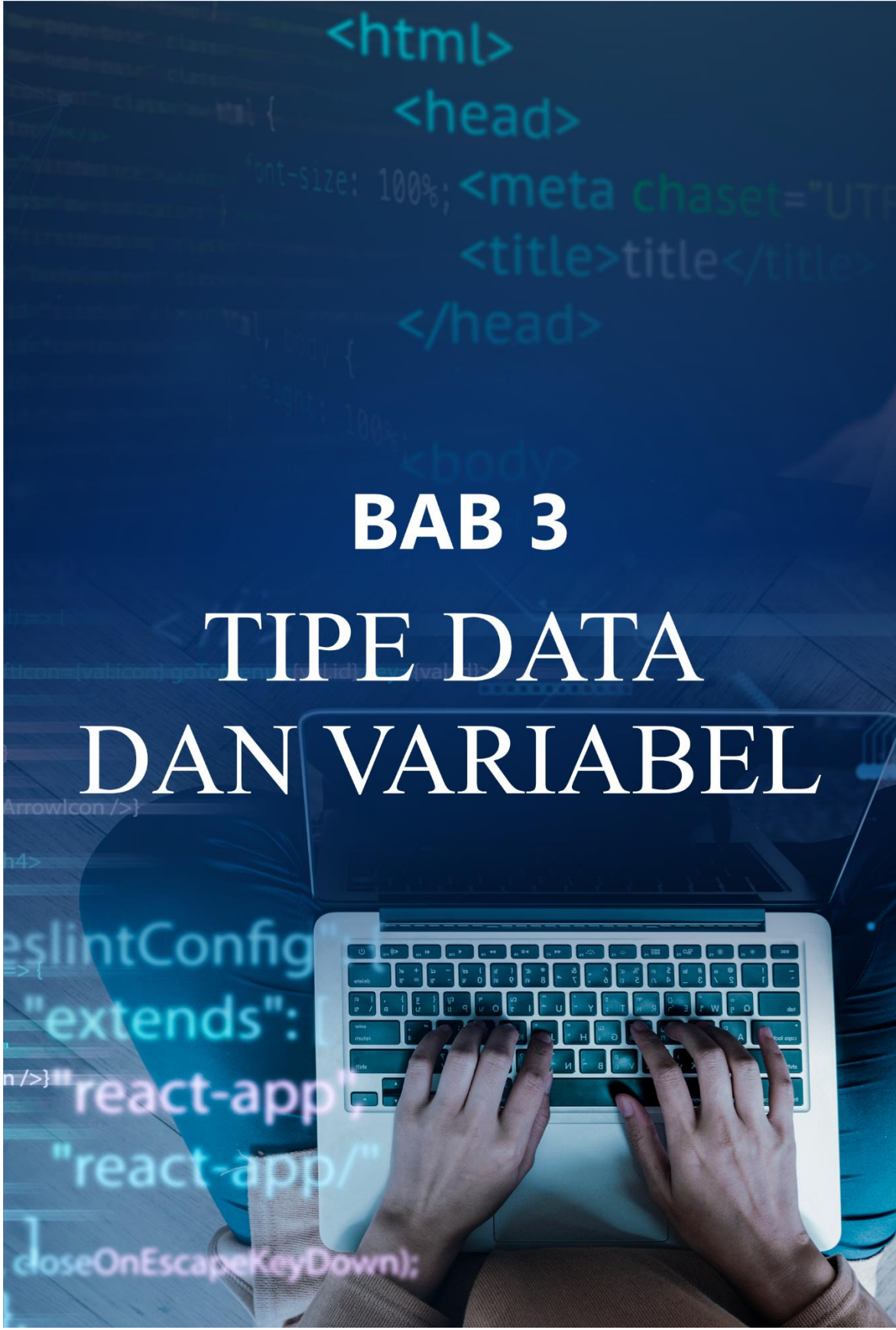
Gambar 2.18. Hasil Keluaran

Dari gambar diatas, kata "Hello, World!" muncul di layar pada tab Terminal. Perintah "System.out.println("Hello, World!)" pada baris 3 Gambar 2.17 bertujuan untuk menampilkan kata yang dimasukkan ke dalam layar. Apabila ingin menampilkan kata atau kalimat lain, cukup dengan mengubah isian dari perintah

"System.out.println" dengan kata-kata yang dikehendaki. Sebagai contoh, pengguna ingin menampilkan "Saya Belajar Pemrograman" di layar, maka perintahnya diubah menjadi *"System.out.println("Saya Belajar Pemrograman")"*.

BAB 3

TIPE DATA DAN VARIABEL



3.1. Tipe Data dan Variabel

Tipe data adalah **jenis data** berdasarkan isi dan sifatnya. Tipe data diibaratkan sebagai sesuatu yang dapat dimasukkan ke dalam kotak kosong yang hanya khusus dibuat untuk jenis benda dengan jenis tertentu. Sementara variabel adalah tempat untuk menyimpan data dengan tipe tertentu.

Beberapa ciri-ciri dari variabel diantaranya:

- Penamaan bersifat **UNIK**, artinya harus berbeda dengan variabel lainnya yang digunakan pada algoritma.
- Satu variabel hanya menyimpan satu jenis data saja.
- Tidak bergantung pada besarnya data.
- Nilai bisa berubah-ubah

CATATAN:

Variabel vs Konstanta

Variabel bersifat nilai dapat diubah-ubah. Jika suatu nilai dalam variabel diubah, maka nilai yang telah diisikan sebelumnya akan diganti dengan nilai yang baru.

Konstanta adalah variabel yang telah ditentukan standarnya (default) dari awal dan besarnya tidak dapat diubah-ubah (contoh: nilai phi, epsilon, dll).

Aturan penamaan variabel diantaranya:

- Karakter pertama harus berupa huruf, tidak berupa angka
- Tidak boleh ada spasi
- Tidak boleh ada simbol-simbol khusus (*, /, +, -, ^, &), kecuali tanda *underscore* (_)

CATATAN:

Underscore lazim digunakan untuk penamaan variabel apabila menamai variabel lebih dari satu kata dan sebagai alternatif untuk pengganti spasi.

Sebagai praktikum, perhatikan tabel kode program berikut.

Tabel 3.1. Kode Program Tipe Data dan Variabel

```
1 public class App {  
2     public static void main(String[] args) throws Exception {  
3         int bilangan1 = 67;  
4         System.out.println(bilangan1);  
5     }  
6 }
```

Kode diatas menginginkan *user* untuk menampilkan hasil "67" pada layar *output*, dengan cara variabel "bilangan1" dipanggil pada perintah "System.out.println", sehingga *user* hanya memodifikasi isian variabel "bilangan1" maka hasil *output* juga akan berubah. Baris ke-3 mendeklarasikan variabel "bilangan1" dengan tipe data int. Penulisan variabel dengan nama yang sama pada baris selanjutnya tetap diizinkan, dengan catatan tidak boleh menambahkan tipe data di depan variabel. Penulisan kode terdapat pada tabel di bawah ini.

Tabel 3.2. Kode Program yang Dimodifikasi

```
1 public class App {  
2     public static void main(String[] args) throws Exception {  
3         int bilangan1 = 67;  
4         bilangan1 = 120;  
5         System.out.println(bilangan1);  
6     }  
7 }
```

Apabila kode dituliskan seperti tabel di atas, maka hasilnya bukan lagi "67", melainkan "120". Sesuai kaidah pemrograman, apabila menuliskan variabel yang sama di baris selanjutnya, maka hasil yang dimunculkan yaitu yang terakhir dideklarasikan.

3.2. Tipe Data Bilangan Bulat

Tipe data bilangan bulat yaitu tipe data yang **hanya** dapat menyimpan bilangan bulat tanpa bilangan koma. Terdapat beberapa jenis tipe data berjenis bilangan bulat, diantaranya:

- byte, yaitu tipe data dengan jangkauan terkecil, antara -128 sampai 127
- short, yaitu tipe data dengan jangkauan antara -32768 sampai 32767
- int, yaitu tipe data dengan jangkauan antara -2147483648 sampai 2147483647
- long, yaitu tipe data dengan jangkauan terbesar, antara -9223372036854775808 sampai 9223372036854775807

CATATAN:

Dalam tipe data bilangan bulat, jangkauan masing-masing tipe data adalah 2 pangkat ukuran tipe data (dalam satuan bit). Namun, jangkauan tersebut dibagi menjadi 3 bilangan, yaitu bilangan bulat negatif, bilangan 0, dan bilangan bulat positif. Tipe data byte, berukuran 8 bit, yaitu dengan jangkauan sebanyak $2^8 = 256$ angka.

Tipe data short, berukuran 16 bit, yaitu dengan jangkauan sebanyak $2^{16} = 65.536$ angka.

Tipe data int, berukuran 32 bit, yaitu dengan jangkauan sebanyak $2^{32} = 4.294.967.296$ angka.

Tipe data long, berukuran 64 bit, yaitu dengan jangkauan sebanyak $2^{64} = 18.446.744.073.709.551.616$ angka.

Sementara itu, terdapat beberapa bentuk tipe data bilangan bulat, diantaranya:

- Bilangan bulat biasa, misal: `int angka = 10;`
- Bilangan oktal, misal: `int angka = 067;`
- Bilangan heksadesimal, misal: `int angka = 0x5d;`

CATATAN:

- Bilangan bulat biasa dilambangkan dengan deteran angka yang dimulai dengan bilangan bukan nol.
- Bilangan oktal adalah bilangan bulat berbasis 8, yang hanya terdiri atas angka 0 sampai 7. Ditulis seperti bilangan biasa dan dimulai dengan 0.
- Bilangan heksadesimal adalah bilangan bulat berbasis 16. terdiri dari angka 0 sampai 9 dan huruf a sampai f. Dimulai dengan 0x.

Pada outputnya, tetap akan menghasilkan bilangan bulat biasa

Contoh:

`int angka = 067` (bilangan oktal), maka yang muncul di output adalah 55 (dalam desimal).

`int angka = 0x5d` (bilangan heksadesimal), maka yang muncul di output adalah 93 (dalam desimal).

Sebagai praktikum, perhatikan tabel kode program berikut. Menurut Anda, bagaimana hasil outputnya?

Tabel 3.3. Kode Program Tipe Data Bilangan Bulat

1	public class App {
2	public static void main(String[] args) throws Exception {
3	byte bilangan1 = 67;
4	short bilangan2 = 548;
5	int bilangan3 = 26382;
6	long bilangan4 = 38274921;
7	System.out.println(bilangan1);
8	System.out.println(bilangan2);
9	System.out.println(bilangan3);
10	System.out.println(bilangan4);
11	}
12	}

Tabel 3.4. Kode Program Tipe Data Bilangan Bulat Oktal dan Heksadesimal

1	public class App {
2	public static void main(String[] args) throws Exception {
3	int bilangan1 = 0337;
4	int bilangan2 = 0x1a4;
5	System.out.println(bilangan1);
6	System.out.println(bilangan2);
7	}
8	}

3.3. Tipe Data Bilangan Riil

Tipe data bilangan riil yaitu tipe data menyatakan bilangan yang membutuhkan ketelitian dengan adanya nilai dibelakang koma. Tipe data ini dapat menuliskan bilangan bulat tanpa koma, namun pada hasil output akan menampilkan bilangan bulat dengan

koma. Misalkan, pada variabel tertentu memasukkan nilai "67", maka pada output akan muncul nilai "67.0". Terdapat beberapa jenis tipe data berjenis bilangan bulat, diantaranya:

- Float, yaitu tipe data berukuran 4 byte (32 bit) dengan jangkauan $1,4 \times 10^{-45}$ sampai $3,4 \times 10^{38}$, dengan presisi 6-7 digit
- Double, yaitu tipe data berukuran 8 byte (64 bit) dengan jangkauan $4,9 \times 10^{-324}$ sampai $1,8 \times 10^{308}$, dengan presisi 15 digit.

CATATAN:

Dalam bahasa Java, tipe data bilangan riil secara default mengandung tipe double, sehingga apabila ingin menggunakan tipe float, maka perlu menambahkan imbuhan akhir f.

Misal: float angka = 2.55f; *atau*
float angka = (float) 2.55;

Sebagai praktikum, perhatikan tabel kode program berikut. Menurut Anda, bagaimana hasil outputnya?

Tabel 3.5. Kode Program Tipe Data Bilangan Riil

1	public class App {
2	public static void main(String[] args) throws Exception {
3	double bilangan1 = 1038;
4	double bilangan2 = 1743.223;
5	System.out.println(bilangan1);
6	System.out.println(bilangan2);
7	}
8	}

3.4. Tipe Data Karakter

Suatu karakter dilambangkan dengan kode ASCII. Penulisan kodenya dimulai dengan \u00, kemudian diikuti dengan kode ASCII (dalam heksadesimal).

Contoh: char huruf = '\u0041'. Maka akan menampilkan huruf 'A'

AMERICAN STANDARD CODE for INFORMATION INTERCHANGE	
0	20
1	21
2	22
3	23
4	24
5	25
6	26
7	27
8	28
9	29
10	30
11	31
12	32
13	33
14	34
15	35
16	36
17	37
18	38
19	39
40	60
41	61
42	62
43	63
44	64
45	65
46	66
47	67
48	68
49	69
50	70
51	71
52	72
53	73
54	74
55	75
56	76
57	77
58	78
59	79
80	80
81	81
82	82
83	83
84	84
85	85
86	86
87	87
88	88
89	89
90	90
91	91
92	92
93	93
94	94
95	95
96	96
97	97
98	98
99	99
P	100
Q	101
R	102
S	103
T	104
U	105
V	106
W	107
X	108
Y	109
Z	110
[111
\	112
]	113
^	114
_	115
a	116
b	117
c	118
	119
d	120
e	121
f	122
g	123
h	124
i	125
j	126
k	127
l	128
m	129
n	130
o	131
p	132
q	133
r	134
s	135
t	136
u	137
v	138
w	139
x	140
y	141
z	142
<	143
>	144
~	145
Δ	146
Ç	147
ü	148
é	149
ê	150
â	151
ä	152
å	153
à	154
ç	155
ê	156
ë	157
è	158
é	159
î	160
ï	161
ä	162
å	163
æ	164
œ	165
ø	166
õ	167
ö	168
ù	169
û	170
ü	171
ý	172
ÿ	173
ÿ	174
ÿ	175
ÿ	176
ÿ	177
ÿ	178
ÿ	179
á	180
â	181
ã	182
ä	183
å	184
æ	185
ç	186
è	187
é	188
ê	189
ë	190
ì	191
í	192
î	193
ï	194
ê	195
ë	196
ì	197
í	198
î	199
ÿ	200
ÿ	201
ÿ	202
ÿ	203
ÿ	204
ÿ	205
ÿ	206
ÿ	207
ÿ	208
ÿ	209
ÿ	210
ÿ	211
ÿ	212
ÿ	213
ÿ	214
ÿ	215
ÿ	216
ÿ	217
ÿ	218
ÿ	219
ÿ	220
ÿ	221
ÿ	222
ÿ	223
ÿ	224
ÿ	225
ÿ	226
ÿ	227
ÿ	228
ÿ	229
ÿ	230
ÿ	231
ÿ	232
ÿ	233
ÿ	234
ÿ	235
ÿ	236
ÿ	237
ÿ	238
ÿ	239
ÿ	240
ÿ	241
ÿ	242
ÿ	243
ÿ	244
ÿ	245
ÿ	246
ÿ	247
ÿ	248
ÿ	249
ÿ	250
ÿ	251
ÿ	252
ÿ	253
ÿ	254
ÿ	255

CATATAN:

ASCII atau singkatan dari American Standard Code for Information Interchange adalah standar pengkodean karakter untuk alat komunikasi, yang mewakili teks dalam komputer, peralatan telekomunikasi, dan perangkat lainnya.

Tabel kode ASCII di atas menggunakan bilangan desimal. Untuk mengubahnya ke bilangan heksadesimal, maka cukup dengan membagi bilangan tersebut dengan 16.

Sebagai contoh: untuk karakter 'K', maka kode ASCII nya adalah 75.

Untuk mengubah ke bilangan heksadesimal, maka 75 dibagi 16, hasilnya adalah 4 sisa 11. (11 dalam bilangan heksadesimal adalah b)

Karena penulisan kodenya yaitu '\u00xx' dimana xx adalah dua digit bilangan heksadesimal, maka digit pertama adalah hasilnya, yaitu 4, sementara digit kedua adalah sisa pembagiannya, yaitu 11 (dalam heksadesimal = b).

Jadi, untuk menampilkan karakter 'K', maka kode yang ditulis adalah '\u004b'

Selain itu, terdapat karakter berjenis escape sequence, dinataranya yaitu sebagai berikut:

- \n = newline, enter atau baris baru
- \t = tab
- \b = backspace
- \\ = karakter \
- \' = karakter ' (petik satu)
- \" = karakter " (petik dua)

Sebagai praktikum, perhatikan tabel kode program berikut. Menurut Anda, bagaimana hasil outputnya?

Tabel 3.6. Kode Program Tipe Data Karakter

1	public class App {
2	public static void main(String[] args) throws Exception {
3	char karakter1 = 'A';
4	char karakter2 = '\\';
5	char karakter3 = '\u0041';
6	System.out.println(karakter1);
7	System.out.println(karakter2);
8	System.out.println(karakter3);
9	}
10	}

Tabel 3.7. Modifikasi Kode Program Tipe Data Karakter

1	public class App {
2	public static void main(String[] args) throws Exception {
3	char karakter1 = 'A';
4	int konversi = (int) karakter1;
5	System.out.println(konversi);
6	}
7	}

3.5. Tipe Data String

String merupakan untaian huruf dan angka yang tersusun menjadi satu kata atau kalimat. Tipe data **String** tidak disimpan dalam bentuk array. Panjang karakter untuk tipe string sebanyak 1 sampai 256 karakter.

Sebagai praktikum, perhatikan tabel kode program berikut. Menurut Anda, bagaimana hasil outputnya?

Tabel 3.8. Kode Program Tipe Data String

1	public class App {
2	public static void main(String[] args) throws Exception {
3	String nama = "Fadhel Akhmad Hizham";
4	String hallo = "Selamat datang \"Fadhel Akhmad
5	Hizham\"";
6	System.out.println(nama);
7	System.out.println(hallo);
8	System.out.println("Selamat Datang\nFadhel Akhmad
9	Hizham\nSemoga Harimu Cerah");
	}
	}

3.6. Tipe Data Boolean

Boolean merupakan sebuah tipe data untuk pernyataan benar atau salah. Tipe data ini **hanya** dapat diisi dengan dua buah nilai, yaitu true atau false.

ASESMEN

Jawablah pertanyaan di bawah ini dengan benar!

1. Berikut ini merupakan karakteristik dari variabel yaitu ...
(jawaban dapat lebih dari satu)
 - penamaan unik
 - dapat menyimpan berbagai jenis data
 - tidak bergantung pada besarnya data
 - nilai bisa berubah
2. Jenis data sebagai tempat penyimpanan khusus berdasarkan isi dan tempatnya disebut ...
3. Perhatikan penulisan variabel berikut ini. Berdasarkan informasi berikut, berilah pernyataan **Benar** atau **Salah**.

Pernyataan	Benar/Salah
satuVariabel	...
satu variabel	...
1Variabel	...
satu_variabel	...

4. Variabel yang telah ditentukan nilainya dan tidak dapat diubah-ubah dengan nilai apapun disebut ...
 - a. tipe data
 - b. konstanta
 - c. karakter
 - d. variabel

5. Perhatikan pernyataan berikut ini:

Salah satu aturan dalam penulisan variabel yaitu tidak boleh menambahkan spasi apabila variabel tersebut terdiri atas dua atau lebih kata. Selain itu, penulisan variabel tidak diperbolehkan untuk membubuhkan simbol apapun.

Maka, alternatif yang bisa digunakan apabila variabel berjumlah lebih dari satu kata dan ingin dipisahkan antar katanya, yaitu dengan membubuhkan simbol garis bawah di antara kata tersebut.

Setujukah Anda dengan pernyataan berikut? Berikan alasannya!

.....
.....
.....
.....
.....
.....
.....
.....

6. Tipe data yang hanya memuat bilangan bulat tanpa koma yaitu tipe data ...
7. Pasangkan pernyataan mengenai tipe data bilangan bulat berikut.

7593	(1)	(a)	long
-32	(2)	(b)	short
2098491000	(3)	(c)	int
2710000000000000	(4)	(d)	byte

8. Berikut ini bilangan yang dapat ditampung pada tipe data short yaitu ... (jawaban dapat lebih dari satu)

- 32767
- 32767
- 32768
- 32768

Perhatikan kode program di bawah ini untuk soal nomor 9 – 10.

```
1 public class App {
2     public static void main(String[] args) throws Exception {
3         int bilangan1 = 074;
4         int bilangan2 = 99;
5         int bilangan3 = 081;
6         int bilangan4 = 0xa4d;
7         System.out.println(bilangan1);
8         System.out.println(bilangan2);
9         System.out.println(bilangan3);
10        System.out.println(bilangan4);
11    }
12 }
```

9. Dari kode di atas, maka variabel yang merupakan bilangan heksadesimal yaitu ...

- a. bilangan1
- b. bilangan2
- c. bilangan3
- d. bilangan4

10. Dari kode di atas, maka berdasarkan informasi berikut, berilah pernyataan **Benar** atau **Salah**.

Pernyataan	Benar/Salah
Output bilangan1 = 74	...
Output bilangan2 = 99	...
Output bilangan3 = error / exception	...
Output bilangan4 = 2637	...

11. Tipe data riil double memiliki ukuran tipe data sebesar ...
- 8 byte
 - 8 bit
 - 4 byte
 - 4 bit

Perhatikan kode program di bawah ini untuk soal nomor 12 – 14.

```
1  public class App {
2      public static void main(String[] args) throws Exception {
3          double bilangan1 = 7344;
4          System.out.println(bilangan1);
5      }
6  }
```

Berdasarkan kode program di atas, maka hasil outputnya yaitu 7344.

12. Setujukah Anda dengan pernyataan berikut?

.....
.....
.....

13. Berikan alasannya!

.....
.....
.....
.....
.....
.....

14. Ubahlah kode di atas dengan tepat untuk mendeklarasikan variabel "bilangan1" dengan tipe data float!

.....
.....
.....
.....
.....

15. Pasangkan pernyataan mengenai tipe data bilangan riil berikut.

$3,4 \times 10^{38}$	(1)
$1,8 \times 10^{308}$	(2)
$1,4 \times 10^{-45}$	(3)
$4,9 \times 10^{-324}$	(4)

(a)	batas bawah double
(b)	batas atas float
(c)	batas atas double
(d)	batas bawah float

16. Untuk memberikan baris baru pada output program, maka dapat menggunakan karakter escape sequence, yaitu

17. Apa yang Anda ketahui mengenai kode ASCII dan bagaimana penggunaannya?

.....
.....
.....
.....
.....
.....

18. Perhatikan kode program berikut ini:

```
1 public class App {  
2     public static void main(String[] args) throws Exception {  
3         char karakter1 = '\u0036';  
4         char karakter2 = '\u004f';  
5         char karakter3 = '\u006f';  
6         char karakter4 = '\u003c';  
7         System.out.println(karakter1);  
8         System.out.println(karakter2);  
9         System.out.println(karakter3);  
10        System.out.println(karakter4);  
11    }  
12 }
```

Pasangkan hasil output berikut.

karakter1	(1)	(a)	o
karakter2	(2)	(b)	<
karakter3	(3)	(c)	O
karakter4	(4)	(d)	6

Perhatikan kode program di bawah ini untuk soal nomor 19 – 21.

```
1 public class App {
2     public static void main(String[] args) throws Exception {
3         char karakter1 = 'z';
4         int konversi = (int) karakter1;
5         System.out.println(konversi);
6     }
7 }
```

19. Jelaskan menurut Anda berfungsi sebagai apakah kode di atas?

.....
.....
.....
.....
.....
.....

20. Hasil dari kode di atas yaitu ...

- a. 90
- b. 5a
- c. 122
- d. 7a

21. Untuk menghasilkan nilai 42 pada output, maka isian variabel karakter1 yang benar adalah ...

22. Di antara pilihan tersebut, panjang karakter dalam sebuah kata atau kalimat yang dapat ditampung pada tipe data string yaitu ... (jawaban dapat lebih dari satu)

- 1
- 255
- 256
- 512

23. Berdasarkan informasi berikut, berilah pernyataan **Benar** atau **Salah**.

Pernyataan	Benar/Salah
Tipe data boolean hanya dapat diisi oleh dua pilihan isian	...
Tipe data string merupakan tipe data satu-satunya yang diawali dengan huruf kapital	...
Tipe data int dapat diisi dengan karakter selain numerik	...
Tipe data double dan float akan menampilkan output yang terdapat angka dibelakang koma	...

Perhatikan kode program di bawah ini untuk soal nomor 24 – 25.

1	<code>public class App {</code>
2	<code> public static void main(String[] args) throws Exception {</code>
3	<code> String nama = "Sugeng Slamet";</code>
4	<code> String hallo = "Selamat siang \"Bang Jali\"";</code>
5	<code> System.out.println(nama);</code>
6	<code> System.out.println(hallo);</code>
7	<code> System.out.println("Selamat siang\nBang Jali\nApa kabar</code>
8	<code> hari ini?");</code>
9	<code> }</code>
9	<code>}</code>

24. Hasil dari kode untuk variabel hallo yaitu ...

- a. "Selamat siang Bang Jali"
- b. Selamat siang Bang Jali
- c. Selamat siang "Bang Jali"
- d. Selamat siang \"Bang Jali\""

25. Bagaimanakah hasil output untuk baris 7?

.....

.....

.....

.....

.....

.....

SOAL PRAKTIKUM

Modifikasilah kode program pada Tabel 3.8 yang berisi keterangan singkat dari mahasiswa masing-masing, berupa nama, NIM, dan kelas, seperti pada kode berikut.

1	public class App {
2	public static void main(String[] args) throws Exception {
3	String nama = "Nama Mahasiswa";
4	String NIM = "NIM Mahasiswa";
5	String kelas = "Kelas Mahasiswa";
6	}
7	}

Lalu tampilkan ke layar menjadi hasil keluaran berikut **hanya** dengan menggunakan satu perintah "*system.out.println*" saja!

Contoh hasil keluaran:

Nama: Budi Setiabudi

NIM: 322140203

Kelas: 1ITA2



`<html>`
`<head>`
`font-size: 100%; <meta charset="UTF-8">`
`<title>title</title>`
`</head>`

BAB 4

DEKLARASI

INPUT

dan OUTPUT

`ArrowIcon />`
`h4>`
`eslintConfig`
`=> {`
`"extends": [`
`n /> "react-app",`
`"react-app/"`
`closeOnEscapeKeyDown);`

4.1. Input dan Output

Setiap program Java akan membuka 3 pernyataan untuk menerima input dan membuka output, diantaranya:

- `System.out`, pernyataan untuk menampilkan tampilan dari program.
- `System.in`, pernyataan untuk menerima input ke program.
- `System.err`, pernyataan untuk menampilkan kesalahan dari program.

Untuk pernyataan output itu sendiri terdapat beberapa macam pernyataan, diantaranya:

- **`System.out.print (expression)`**, untuk mencetak `expression` ke standar output dan cursor penulisan terdapat pada kolom selanjutnya.
- **`System.out.println (expression)`**, untuk mencetak `expression` ke standar output dan cursor penulisan terdapat pada baris selanjutnya.
- **`System.out.printf (string format, expression)`**, untuk menampilkan `expression` sesuai dengan format ke standar output.

Pada string format, terdapat beberapa jenis string format, diantaranya:

- Integer: `%d`, `%o`, `%x` atau `%X`
- String & Character: `%s`, `%c`
- Floating point: `%e` atau `%E`, `%f`
- Boolean: `%b`
- Hash value: `%h`
- Simbol: `%%`
- New Line: `%n`

Untuk membuat spasi pada string format, lebarnya ditentukan oleh parameter sebelum diisi string format.

Contoh:

- `%4d` : untuk menampilkan integer dengan spasi 4 karakter
- `%6.4f` : untuk menampilkan floating point dengan spasi 6 karakter dan 4 angka dibelakang koma.

Sebagai praktikum, perhatikan tabel kode program berikut. Menurut Anda, bagaimana hasil outputnya?

Tabel 4.1. Kode Program Menampilkan Kalimat ke Layar

1	<code>public class App {</code>
2	<code> public static void main(String[] args) throws Exception {</code>
3	<code> System.out.print("Selamat datang ");</code>
4	<code> System.out.println("Ini akan ditampilkan di sebelah</code>
5	<code>string sebelumnya");</code>
6	<code> System.out.println("Ini akan ditampilkan pada baris</code>
7	<code>berikutnya");</code>
8	
9	<code> //printf</code>
10	<code> System.out.printf("Data Integer: %4d\n", 750);</code>
11	<code> System.out.printf("Data Float: %6.4f\n", 2.7183);</code>
12	<code> System.out.printf("Data Character: %c\n", 'c');</code>
13	<code> System.out.printf("Data String: %s\n", "Ini adalah</code>
	<code>string");</code>
	<code> }</code>
	<code>}</code>

4.2. Menerima *Input* dari *Keyboard* dengan *Scanner*

Menggunakan *scanner* sebagai *input* dari *keyboard* sudah menjadi hal yang umum dalam membuat sebuah program berbasis *Java*. *Scanner* berasal dari paket `java.util`. Untuk metode pembacaan data terdapat beberapa macam, diantaranya:

- `nextBoolean` (untuk tipe data boolean)
- `nextByte`
- `nextShort`
- `nextInt`
- `nextLong`
- `nextFloat`
- `nextDouble`
- `next` (untuk tipe data string)
- `nextLine().charAt(0)` (untuk tipe data character / char)

Langkah-langkah penulisan program penggunaan *Scanner* yaitu:

- Mengimpor pustaka java kelas `Scanner` (ditulis di atas `public class`):
`import java.util.Scanner;`
- Membuat instansiasi:
`Scanner sc = new Scanner(System.in);`
- Menerima masukan:
`int bil1 = sc.nextInt();`
`double bil2 = sc.nextDouble();`

Sebagai praktikum, perhatikan tabel kode program berikut. Menurut Anda, bagaimana hasil outputnya?

Tabel 4.2. Kode Program Menerima *Input* dengan *Scanner*

1	<code>import java.util.Scanner;</code>
2	
3	<code>public class App {</code>
4	<code> public static void main(String[] args) throws Exception {</code>
5	<code> Scanner sc = new Scanner(System.in);</code>
6	

7	System.out.println("Masukkan bilangan 1: ");
8	int bil1 = sc.nextInt();
9	System.out.println("Masukkan bilangan 2: ");
10	double bil2 = sc.nextDouble();
11	
12	System.out.println("Bilangan 1: " + bil1 + "\nBilangan 2:
	" + bil2);
13	}
14	}

4.3. Menerima *Input* dari *Keyboard* dengan *BufferedReader*

Selain menggunakan kelas Scanner, program Java juga dapat membaca masukan dengan kelas *BufferedReader*. *BufferedReader* membaca masukan sebagai sebuah string. Untuk menggunakan tipe data lain, perlu mengonversi tipe data string menjadi tipe data yang diperlukan. Fungsi-fungsi konversi:

- `Integer.parseInt(input)` -> mengubah string menjadi integer.
- `Double.parseDouble(input)` -> mengubah string menjadi double.
- `Float.parseFloat(input)` -> mengubah string menjadi float.

Langkah-langkah penulisan program penggunaan *BufferedReader* yaitu:

- Mengimpor pustaka java kelas *BufferedReader* dan *InputStreamReader* (ditulis di atas public class):
import java.io.BufferedReader;
import java.io.InputStreamReader;
- Membuat instansiasi:
BufferedReader reader = new BufferedReader(new
InputStreamReader (System.in));

- Menerima masukan:

```
int bil1 = Integer.parseInt (reader.readLine());  
double bil2 = Double.parseDouble  
(reader.readLine());
```

Sebagai praktikum, perhatikan tabel kode program berikut. Menurut Anda, bagaimana hasil outputnya?

Tabel 4.3. Kode Program Menerima *Input* dengan *BufferedReader*

1	import java.io.BufferedReader;
2	import java.io.InputStreamReader;
3	
4	public class App {
5	public static void main(String[] args) throws Exception {
6	BufferedReader reader = new BufferedReader((new
	InputStreamReader(System.in));
7	
8	System.out.println("Masukkan bilangan 1: ");
9	int bil1 = Integer.parseInt(reader.readLine());
10	System.out.println("Masukkan bilangan 2: ");
11	double bil2 = Double.parseDouble(reader.readLine());
12	
13	System.out.println("Bilangan 1: " + bil1 + "\nBilangan 2:
	" + bil2);
14	}
15	}

ASESMEN

Jawablah pertanyaan di bawah ini dengan benar!

1. Pasangkan makna dari format string integer berikut.

%d	(1)	(a)	oktal
%o	(2)	(b)	heksadesimal (lowercase)
%x	(3)	(c)	heksadesimal (uppercase)
%X	(4)	(d)	desimal

2. Pernyataan yang digunakan untuk menampilkan kesalahan dari sebuah program yaitu ...
3. Perintah output yang mencetak hasil output dan kursor ditampilkan pada baris yang sama yaitu ...
- System.out print
 - System.out printa
 - System.out println
 - System.out printf
4. Berikut ini yang termasuk format string floating point yaitu ...
(jawaban dapat lebih dari satu)
- %f
 - %x
 - %E
 - %e

5. Jelaskan menurut Anda perbedaan format string %e, %E dan %f.

.....
.....
.....
.....
.....

Perhatikan kode program di bawah ini untuk soal nomor 6 – 7.

```
1 public class App {  
2     public static void main(String[] args) throws Exception {  
3         System.out.printf("Data 1: %2X\n", 160);  
4         System.out.printf("Data 2: %3.3E\n", 1067.7567);  
5         System.out.printf("Data 3: %3o\n", 100);  
6         System.out.printf("Data 4: %3.4f\n", 3.14159);  
7     }  
8 }
```

6. Program di atas akan dijalankan. Berdasarkan informasi berikut, berilah pernyataan **Benar** atau **Salah**.

Pernyataan	Benar/Salah
Data 1: A0	...
Data 2: 1.067e+03	...
Data 3: 144	...
Data 4: 3.1416	...

7. Jika data 2 diubah menjadi System.out.printf("Data 2: %1.3E\n", 261059.874); maka hasil outputnya yaitu ...

- a. 2.61060e+05
- b. 2.610e+03
- c. 2.611e+05
- d. 2.611e+03

Perhatikan pernyataan berikut ini:

Untuk membuat input dari *keyboard* menggunakan *Scanner* ataupun *BufferedReader*, maka di atas public class, harus mengimpor pustaka pada Java dengan kode berikut:

```
import java.util.*
```

8. Setujukah Anda dengan pernyataan berikut?

.....
.....
.....
.....
.....

9. Berikan alasannya!

.....
.....
.....
.....
.....

Perhatikan pernyataan berikut ini:

Metode pembacaan data untuk membuat input dari keyboard dengan menggunakan *Scanner*, maka harus menyesuaikan tipe data yang dijadikan variabel input. Misal variabel "input" beripe data short, maka pembacaan datanya yaitu `nextLong()`

10. Setujukah Anda dengan pernyataan berikut?

.....
.....
.....
.....
.....

11. Berikan alasannya!

.....
.....
.....
.....

12. Metode pembacaan data untuk tipe data char yaitu ...

Perhatikan kode program di bawah ini untuk soal nomor 13 – 14.

```
1 import java.util.Scanner;
2
3 public class App {
4     public static void main(String[] args) throws Exception {
5         Scanner input = new Scanner(System.in);
6
7         System.out.println("Masukkan bilangan 1: ");
8         byte bil1 = (1);
9         System.out.println("Masukkan bilangan 2: ");
10        double bil2 = (2);
11
12        System.out.println("Bilangan 1: " + bil1 + "\nBilangan 2:
13        " + bil2);
14    }
15 }
```

13. Kode yang tepat untuk mengisi (1) yaitu ...
14. Variabel bil1 dan bil2 yang memungkinkan untuk diinput di keyboard yaitu ... *(jawaban dapat lebih dari satu)*
- 103 dan 25000
 - 650 dan 110
 - 128 dan 4.0333
 - 127 dan 0.473433
15. Pada BufferedReader, tipe data default sebagai masukan yaitu...
- a. int
 - b. double
 - c. boolean
 - d. string
16. Menurut Anda, bagaimana cara untuk mengonversikan tipe data pada BufferedReader?

.....

.....

.....

.....

17. Impor library Java untuk membuat input dari keyboard dengan menggunakan BufferedReader yaitu ... *(jawaban dapat lebih dari satu)*
- import java.util.BufferedReader
 - import java.util.InputStreamReader
 - import java.util.Scanner
 - import java.util.*

18. Jelaskan perbedaan fungsi import BufferedReader dan InputStreamReader.

.....

.....

.....
.....

Perhatikan kode program di bawah ini untuk soal nomor 19 – 20.

```
1 import java.io.BufferedReader;
2 import java.io.InputStreamReader;
3
4 public class App {
5     public static void main(String[] args) throws Exception {
6         BufferedReader reader = new BufferedReader((new
7             InputStreamReader(System.in)));
8
9         short var1 = (1) (reader.readLine());
10        long var2 = (2) (reader.readLine());
11        double var3 = (3) (reader.readLine());
12        int var4 = (4) (reader.readLine());
13    }
14 }
```

19. Berdasarkan informasi berikut, berilah pernyataan **Benar** atau **Salah**.

Pernyataan	Benar/Salah
(1) = short.parseShort()	...
(2) = Long.parseLong()	...
(3) = Double.parseDouble()	...
(4) = Int.parseInt()	...

20. Pasangkan inputan yang cocok untuk masing-masing variabel tersebut.

var1	(1)
var2	(2)
var3	(3)
var4	(4)

(a)	125000
(b)	10000
(c)	3.1415926
(d)	-1500000000000

SOAL PRAKTIKUM

Ulangi soal praktikum pada bab 3 yang berisi informasi mengenai Nama, NIM, dan kelas mahasiswa dengan mengubah kode program menjadi sebuah program yang dapat mengisikan *input* dari *keyboard*. NIM Ganjil gunakan Scanner, NIM Genap gunakan BufferedReader.

Contoh hasil keluaran:

Nama: Budi Setiabudi (diketik di bagian output)

NIM: 322140203 (diketik di bagian output)

Kelas: 1ITA2 (diketik di bagian output)

Selamat Datang

Budi Setiabudi

322140203

1ITA2



BAB 5

FUNGSI OPERATOR



5.1. Operator Pada Pemrograman

Operator adalah sebuah simbol atau tanda yang apabila diletakkan pada dua buah operan dapat menghasilkan sebuah hasil. Sebagai contoh, tanda tambah (+) jika diletakkan di antara kedua angka, maka akan menghasilkan angka lain yang merupakan hasil penjumlahan dari kedua angka tersebut. Tanda '+' inilah yang disebut operator.

Terdapat beberapa jenis operator, diantaranya:

- **Unary**, yaitu operator yang hanya melibatkan 1 operan
 - Contoh: -16
- **Binary**, yaitu operator yang melibatkan 2 operan
 - Contoh: $12 * 4$
- **Ternary**, yaitu operator yang melibatkan 3 operan
 - Contoh: $(a > b) ? c : d$

Operator yang digunakan dalam pemrograman diantaranya:

- **Operator Aritmetik**, yang biasa digunakan dalam operasi hitung matematika. Contoh: $4 + 8$.
- **Operator Relasi**, yang biasa digunakan sebagai perbandingan dua nilai. Contoh: $a > b$.
- **Operator Logika Boolean**, yang biasa digunakan untuk mengaitkan dua kondisi menjadi sebuah kondisi. Contoh: kondisi1 AND kondisi2.
- **Operator Assignment**, yang biasa digunakan untuk memberi tugas pada variabel tertentu. Contoh: $a += 7$.

5.2. Operator Aritmetik

Terapat beberapa operator aritmetik, diantaranya:

- Operator penjumlahan (+)
- Operator pengurangan (-)
- Operator perkalian (*)
- Operator pembagian (/)
- Operator sisa bagi / modulo (%)

CATATAN:

Hasil pembagian selalu menghasilkan nilai pembulatan ke bawah. Misal $16 / 9 = 1,777\dots$. Akan tetapi di pemrograman Java, bilangan dibelakang koma dihilangkan dan output akan menjadi "1".

Namun apabila ingin menampilkan hasil pembagi dengan desimal, maka tipe data perlu diubah ke tipe double atau float sebagai berikut:

```
double bagi = 16.0 / 9.0;
```

```
System.out.println(bagi);
```

Output: 1.7777777777777777 (hingga 16 angka dibelakang koma)

Sebagai praktikum, perhatikan tabel kode program berikut. Menurut Anda, bagaimana hasil outputnya?

Tabel 5.1. Kode Program Operator Aritmetik

```
1 public class App {
2     public static void main(String[] args) throws Exception {
3         int tambah = 30 + 18;
4         int kurang = 20 - 25;
5         int kali = 13 * 12;
6         int bagi = 35 / 10;
7         int sisa_bagi = 38 % 6;
8
9         System.out.println(tambah);
10        System.out.println(kurang);
11        System.out.println(kali);
12        System.out.println(bagi);
13        System.out.println(sisa_bagi);
14    }
15 }
```

5.3. Operator Ternary

Jenis operator ternary "?:" merupakan bentuk pendek dari fungsi if-then-else. Secara umum, format ternary adalah:

"kondisi ? jika_benar : jika_salah"

- Tanda operator "?:" akan melakukan evaluasi dari "kondisi".
- Jika bernilai "true" operator mengembalikan nilai "jika_benar".
- Jika bernilai "false" operator mengembalikan nilai "jika_salah".

Contoh: $x = (14 > 8) ? 25 : 20$

Nilai x di atas akan bernilai 25, karena kondisi $(14 > 8)$ adalah true.

Sebagai praktikum, perhatikan tabel kode program berikut. Menurut Anda, bagaimana hasil outputnya?

Tabel 5.2. Kode Program Operator Ternary

1	public class App {
2	public static void main(String[] args) throws Exception {
3	int a, b;
4	String ternary;
5	a = 5 * 9;
6	b = 20 + 26;
7	ternary = (a > b) ? "true" : "false";
8	System.out.println(ternary);
9	}
10	}

5.4. Operator Kenaikan (++) dan Penurunan (--)

Operator kenaikan (++), berfungsi untuk menambah suatu bilangan dengan 1, contoh:

```
int i = 1;
i++;
System.out.println(i);
```

Output: 2

Operator penurunan (--), berfungsi untuk mengurangi suatu bilangan dengan 1, contoh:

```
int i = 1;
i--;
System.out.println(i);
```

Output: 0

Sebagai praktikum, perhatikan tabel kode program berikut. Menurut Anda, bagaimana hasil outputnya?

Tabel 5.3. Kode Program Operator ++

1	public class App {
2	public static void main(String[] args) throws Exception {
3	int a = 1;
4	System.out.println("a = " + a);
5	System.out.println(++a = " + ++a);
6	System.out.println("a = " + a);
7	
8	int b = 1;
9	System.out.println("b = " + b);
10	System.out.println("b++ = " + b++);

11	System.out.println("b = " + b);
12	
13	int c = 1;
14	System.out.println("c = " + c);
15	System.out.println("c-- = " + c--);
16	System.out.println("c = " + c);
17	}
18	}

5.5. Operator Relasi

Terdapat beberapa operator relasi, diantaranya:

- Operator sama dengan (==)
- Operator tidak sama dengan (!=)
- Operator lebih besar (>)
- Operator lebih besar atau sama (>=)
- Operator lebih kecil (<)
- Operator lebih kecil atau sama (<=)

Sebagai praktikum, perhatikan tabel kode program berikut.
Menurut Anda, bagaimana hasil outputnya?

Tabel 5.4. Kode Program Operator Relasi

1	public class App {
2	public static void main(String[] args) throws Exception {
3	
4	System.out.println((6 == 6));
5	System.out.println((8 != 4*2));
6	System.out.println((12+5 > 3*5));
7	System.out.println((20 >= 3*7));
8	System.out.println((16*7 < 115-3));
9	System.out.println((10*8 <= 9*9));
10	}
11	}

5.6. Operator Logika Boolean

Terdapat beberapa operator logika boolean, diantaranya:

- Operator dan (&&), tabel kebenaran (B = benar, S = salah):

Kondisi 1	Kondisi 2	Hasil operator dan (&&)
B	B	B
B	S	S
S	B	S
S	S	S

- Operator atau (||), tabel kebenaran (B = benar, S = salah):

Kondisi 1	Kondisi 2	Hasil operator atau ()
B	B	B
B	S	B
S	B	B
S	S	S

- Operator not (!), tabel kebenaran (B = benar, S = salah):

Kondisi	Hasil operator not (!)
B	S
S	B

Sebagai praktikum, perhatikan tabel kode program berikut.
Menurut Anda, bagaimana hasil outputnya?

Tabel 5.5. Kode Program Operator Logika Boolean

1	<code>public class App {</code>
2	<code> public static void main(String[] args) throws Exception {</code>
3	<code> //AND</code>
4	<code> System.out.println((6*5 == 3*10) && (40-18 > 5*4));</code>
5	<code> System.out.println((2*7+10 <= 3*8) && (20-10 > 30/3));</code>
6	<code> System.out.println((12+1*3 == 39) && (11*2+9 != 62/2));</code>

```

7
8     //OR
9     System.out.println((6*5 == 3*10) || (40-18 > 5*4));
10    System.out.println((2*7+10 <= 3*8) || (20-10 > 30/3));
11    System.out.println((12+1*3 == 39) || (11*2+9 != 62/2));
12
13    //NOT
14    System.out.println(!(6*5 == 3*10));
15    System.out.println(!(20-10 > 30/3));
16  }
17  }

```

5.7. Operator Assignment

Operator **assignment** atau **pemberi nilai** adalah operator untuk memanipulasi atau menginisialisasi nilai pada sebuah variabel. Operator assignment identik dengan simbol sama dengan '=', dan dapat digabung dengan simbol operator aritmetik (+, -, *, /, %) maupun operator bitwise (&, |, ^, <<=, >>=).

Penggunaan operator assignment pada java digunakan untuk menghitung dua buah variabel serta mengembalikan hasilnya secara langsung. Contoh: diberikan dua variabel integer, yakni X = 10, Y = 15. Untuk menjumlahkan dua variabel tersebut, pada umumnya ditulis X = X + Y. Tetapi dengan menggunakan operator assignment, cukup ditulis sebagai X += Y.

Sebagai praktikum, perhatikan tabel kode program berikut. Menurut Anda, bagaimana hasil outputnya?

Tabel 5.6. Kode Program Operator Assignment 1

```

1  public class App {
2      public static void main(String[] args) throws Exception {
3          int a, b;
4          a = 25;

```

5	b = 30;
6	a = 25 + 30;
7	System.out.println(a);
8	
9	int c, d;
10	c = 25;
11	d = 30;
12	c += d;
13	System.out.println(c);
14	}
15	}

Tabel 5.7. Kode Program Operator Assignment 2

1	public class App {
2	public static void main(String[] args) throws Exception {
3	int a, b, c, d, e, hasil;
4	a = 75;
5	b = 24;
6	c = 16;
7	d = 4;
8	e = 11;
9	hasil = 0;
10	
11	hasil += a;
12	System.out.println("Nilai pertama: " + hasil);
13	hasil -= b;
14	System.out.println("Nilai kedua: " + hasil);
15	hasil *= c;
16	System.out.println("Nilai ketiga: " + hasil);
17	hasil /= d;
18	System.out.println("Nilai keempat: " + hasil);
19	hasil %= e;

20	System.out.println("Nilai kelima: " + hasil);
21	}
22	}

ASESMEN

Jawablah pertanyaan di bawah ini dengan benar!

1. Jenis operator untuk "22 + 44 / 4" yaitu ...
 - a. unary
 - b. binary
 - c. ternary
 - d. quadnary
2. Pasangkan operator yang cocok sesuai jenisnya tersebut.

*	(1)	(a)	assignment
	(2)	(b)	logika boolean
>=	(3)	(c)	aritmetik
-=	(4)	(d)	relasi

Perhatikan kode program di bawah ini untuk soal nomor 3 – 5.

```
1 public class App {
2     public static void main(String[] args) throws Exception {
3         int a, b, c;
4         String ternary;
5         a = 10;
6         b = 40;
7         c = 25 * 2;
8         ternary = ((a + b) > c) ? "true" : "false";
9         System.out.println(ternary);
10    }
11 }
```

3. Bagaimanakah hasil output dari kode program di atas?

.....
.....
.....
.....
.....

4. Berikan alasannya!

.....
.....
.....
.....
.....

5. Agar menghasilkan output true, maka nilai a, b, c yang memungkinkan yaitu ... (*jawaban dapat lebih dari satu*)

- 20, 30, dan $100 / 4$
- 40, 45, dan $17 * 5 - 1$
- 30, 33, dan $7 * 9$
- 22, 78, dan $9 * 11$

Perhatikan kode program di bawah ini untuk soal nomor 6 – 8.

1	public class App {
2	public static void main(String[] args) throws Exception {
3	int oper1 = 22 + 77 - 33;
4	int oper2 = 20 / 5 + 10;
5	int oper3 = 33 + 10 * 3;
6	int oper4 = 27 - 100 / 10;
7	int oper5 = 85 / 4;
8	int oper6 = 144 % 10;
9	

```

10     System.out.println(oper1);
11     System.out.println(oper2);
12     System.out.println(oper3);
13     System.out.println(oper4);
14     System.out.println(oper5);
15     System.out.println(oper6);
16 }
17 }

```

6. Hasil output untuk variabel oper6 yaitu ...
7. Agar hasil output untuk variabel oper5 menjadi 21.25, maka baris 7 harus diubah menjadi ... (*jawaban dapat lebih dari satu*)
- double oper5 = 85.0 / 4
 - double oper5 = 85.0 / 4.0
 - double oper5 = 85 / 4.0
 - double oper5 = 85 / 4
8. Apabila program dijalankan, berdasarkan informasi berikut, berilah pernyataan **Benar** atau **Salah**.

Pernyataan	Benar/Salah
Variabel oper1 = 66	...
Variabel oper2 = 14	...
Variabel oper3 = 129	...
Variabel oper4 = 17	...

Perhatikan kode program di bawah ini untuk soal nomor 9 – 11.

```
1 public class App {
2     public static void main(String[] args) throws Exception {
3         int a = 1;
4         System.out.println("a = " + a);
5         System.out.println("++a = " + ++a);
6         System.out.println("a = " + a);
7
8         int b = 1;
9         System.out.println("b = " + b);
10        System.out.println("b++ = " + b++);
11        System.out.println("b = " + b);
12
13        int c = 1;
14        System.out.println("c = " + c);
15        System.out.println("c-- = " + c--);
16        System.out.println("c = " + c);
17
18        int d = 1;
19        System.out.println("d = " + d);
20        System.out.println("--d = " + --d);
21        System.out.println("d = " + d);
22    }
23 }
```

9. Pasangkan hasil output urutan dari masing-masing variabel berikut.

variabel a	(1)	(a)	1, 1, 0
variabel b	(2)	(b)	1, 2, 2
variabel c	(3)	(c)	1, 1, 2
variabel d	(4)	(d)	1, 0, 0

10. Variabel manakah yang dieksekusi untuk operator kenaikan atau penurunan?

.....
.....
.....
.....
.....

11. Berikan alasannya!

.....
.....
.....
.....
.....

12. Hasil output untuk operator relasi dan logika boolean yaitu ...

13. Tanda operator pada Java untuk "tidak sama dengan" yaitu ...

- a. !=
- b. =!
- c. ≠
- d. !!=

14. Berdasarkan informasi berikut, berilah pernyataan **Benar** (untuk true) atau **Salah** (untuk false).

Pernyataan	Benar/Salah
System.out.println((100 != 5*2*10))	...
System.out.println((30*7 > 220))	...
System.out.println((30+14 == 5*9))	...

System.out.println((108+70/2-40 > 79-5*2+17*2))	...
---	-----

15. Agar hasil outputnya true, maka operator yang tepat untuk melengkapi kode System.out.println((56/2/7 ... 200%19-6)) yaitu ... *(jawaban dapat lebih dari satu)*

- >
- ==
- >=
- <=

16. Pernyataan yang bernilai true di bawah ini yaitu ...

- a. System.out.println((27 == 4 * 7))
- b. System.out.println((32/4 != 2*3))
- c. System.out.println((12*5 < 47+20-7))
- d. System.out.println((70/2+10 <= 4*11))

Perhatikan pernyataan berikut ini:

Menurut tabel kebenaran, operator AND memberikan pernyataan bahwa semua kondisi harus bernilai TRUE untuk menghasilkan nilai TRUE. Apabila salah satu saja bernilai FALSE, maka hasilnya akan menjadi FALSE.

17. Setujukah Anda dengan pernyataan berikut?

.....

.....

.....

.....

.....

18. Berikan alasannya!

.....
.....
.....
.....
.....

19. Operator logika yang menghasilkan nilai yang berkebalikan dengan kondisi yaitu ...

20. Berdasarkan informasi berikut, berilah pernyataan **Benar** (untuk true) atau **Salah** (untuk false).

Pernyataan	Benar/Salah
System.out.println((30*7 > 220) && (24+9*9 != 297))	...
System.out.println((27*5-12 > 7*19-10) (6*8+12 == 6*10))	...
System.out.println((20+8/2+6*5 < 53) (90-72/6 != 78))	...
System.out.println((36*2-19 < 52) !(100-100/10 == 25+5*3))	...

21. Berikut ini pernyataan yang bernilai true yaitu ... (*jawaban dapat lebih dari satu*)

- System.out.println((17 >= 9+8) && (33-10 == 46/2))
- System.out.println((10+7*2 != 34) || (50*2 == 98+1))
- System.out.println(!(37+3 < 20+20));
- System.out.println((30+16/2 >= 40-2) && (55/5 == 9+3));

Perhatikan kedua kode program di bawah ini untuk soal nomor 22 – 23.

```
1 public class App {  
2     public static void main(String[] args) throws Exception {  
3         int bil1, bil2;  
4         bil1 = 25;  
5         bil2 = 30;  
6         bil1 *= bil2;  
7         System.out.println(bil1);  
8     }  
9 }
```

```
1 public class App {  
2     public static void main(String[] args) throws Exception {  
3         int bil1, bil2;  
4         bil1 = 25;  
5         bil2 = 30;  
6         bil1 = bil1 * bil2;  
7         System.out.println(bil1);  
8     }  
9 }
```

22. Apakah kedua kode program di atas memiliki hasil output yang sama?

.....
.....
.....
.....
.....

23. Berikan alasannya!

.....
.....
.....
.....
.....

Perhatikan kedua kode program di bawah ini untuk soal nomor 24 – 25.

1	public class App {
2	public static void main(String[] args) throws Exception {
3	int a, b, c, d, hasil;
4	a = 127;
5	b = 392;
6	c = 17;
7	d = 10;
8	hasil = 0;
9	
10	hasil += a; System.out.println(hasil);
11	hasil -= b; System.out.println(hasil);
12	hasil %= c; System.out.println(hasil);
13	hasil *= d; System.out.println(hasil);
14	}
15	}

24. Jenis operator pada kode program di atas yaitu ...

25. Pasangkan hasil output berikut sesuai dengan eksekusi variabelnya.

a	(1)
b	(2)
c	(3)
d	(4)

(a)	-265
(b)	-100
(c)	127
(d)	-10

SOAL PRAKTIKUM

1. Buatlah sebuah program dengan bahasa Java yang menerima sebuah masukan berupa derajat Celcius dan menampilkan hasil bilangan masukan itu menjadi derajat Fahrenheit (Rumus Celcius ke Fahrenheit: $(9 * \text{Celcius} / 5) + 32$).
2. Buatlah sebuah program dengan bahasa Java yang menerima tiga buah masukan bilangan bulat berupa panjang, lebar, dan tinggi sebuah balok, dan akan menghasilkan nilai luas permukaan balok **tanpa tutup**. Rumus: $p * l + 2 * l * t + 2 * p * t$
3. Buatlah sebuah program dengan bahasa Java yang menerima dua buah bilangan bulat berupa *integer* lalu menampilkan hasil penjumlahan kedua bilangan, pengurangan kedua bilangan, perkalian kedua bilangan, dan pembagian kedua bilangan (masukan pembagi tidak boleh nol).

PETUNJUK Pengerjaan:

Gunakan 3 angka terakhir NIM Anda, lalu hitunglah dengan modulo 3. Setelah itu, kerjakan soal sesuai nomor pada hasil perhitungan modulo tersebut.

Contoh: NIM Anda = 322140203, maka 3 digit terakhir yaitu 203. Hasil dari $203 \% 3 = 2$. Maka Anda mengerjakan soal nomor 2. Apabila hasilnya 0, maka Anda mengerjakan soal nomor 3.



BAB 6

**FUNGSI
PERCABANGAN
(Bagian 1)**

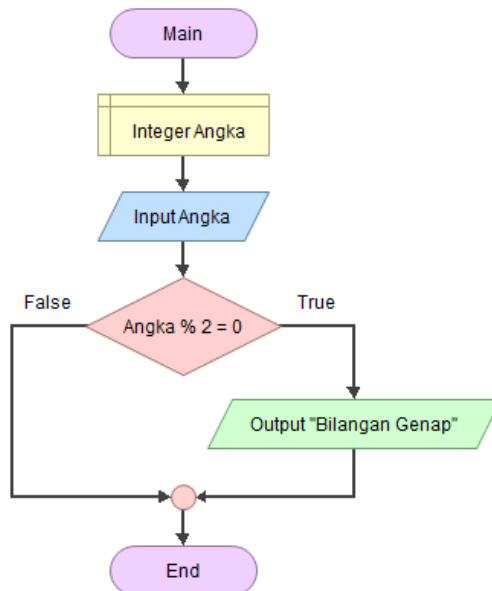
6.1. Percabangan

Percabangan adalah sebuah blok program yang menyatakan bahwa sebuah tindakan akan dijalankan apabila kondisi terpenuhi. Jika satu saja kondisi tidak terpenuhi maka tindakan tidak akan dijalankan. Syntax untuk percabangan dalam pemrograman Java yaitu if, if-else, dan switch-case.

6.2. Percabangan Satu Kondisi

Blok program IF untuk satu kondisi berarti hanya ada satu blok aksi yang dilakukan jika persyaratan terpenuhi. Dengan kata lain, sebuah aksi hanya dapat dikerjakan jika suatu persyaratan terpenuhi, namun sebaliknya jika persyaratan tidak terpenuhi, maka jalannya program tidak akan melakukan apa-apa.

Sebagai contoh, kita akan memasukkan sebuah bilangan, lalu bilangan tersebut diperiksa apakah bilangan tersebut termasuk bilangan genap atau bukan. Maka dalam pemetaan algoritma dalam bentuk flowchart digambarkan sebagai berikut.



Karena hanya berupa if satu kondisi, maka untuk nilai false tidak ada reaksi apa-apa, sehingga tidak ada nilai outputnya.

Sebagai praktikum, perhatikan tabel kode program berikut. Menurut Anda, bagaimana hasil outputnya?

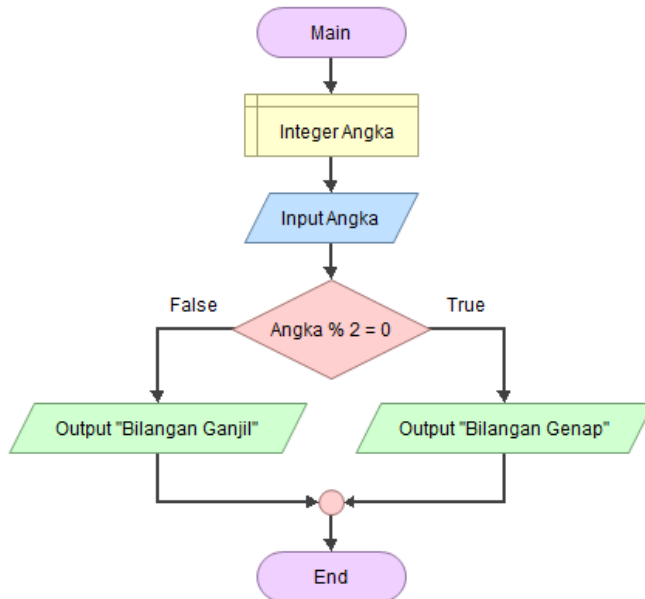
Tabel 6.1. Kode Program Percabangan Satu Kondisi

1	public class App {
2	public static void main(String[] args) throws Exception {
3	Scanner sc = new Scanner(System.in);
4	System.out.println("Masukkan sebuah angka: ");
5	int bilangan = sc.nextInt();
6	
7	if(bilangan%2 == 0) {
8	System.out.println("Angka " + bilangan + " adalah
	bilangan genap");
9	}
10	}
11	}

6.3. Percabangan Dua Kondisi

Blok program IF-ELSE dipergunakan untuk menyatakan suatu percabangan dua kondisi, dimana ada dua blok aksi yang akan dipilih untuk dikerjakan jika kondisi aksi terpenuhi. Saat pembacaan program sampai pada IF-ELSE, maka akan diperiksa persyaratannya, jika syarat terpenuhi maka aksi yang dijalankan adalah pada blok IF, namun jika tidak terpenuhi maka aksi yang dijalankan adalah pada blok ELSE.

Sebagai contoh, kita akan memasukkan sebuah bilangan, lalu bilangan tersebut diperiksa apakah bilangan tersebut termasuk bilangan genap atau ganjil. Maka dalam pemetaan algoritma dalam bentuk flowchart digambarkan sebagai berikut.



Karena ada dua kondisi, maka ada reaksi pada if (true) dan else (false). Dari flowchart di atas, akan menampilkan output “bilangan ganjil” dan “bilangan genap”.

Sebagai praktikum, perhatikan tabel kode program berikut. Menurut Anda, bagaimana hasil outputnya?

Tabel 6.2. Kode Program Percabangan Dua Kondisi

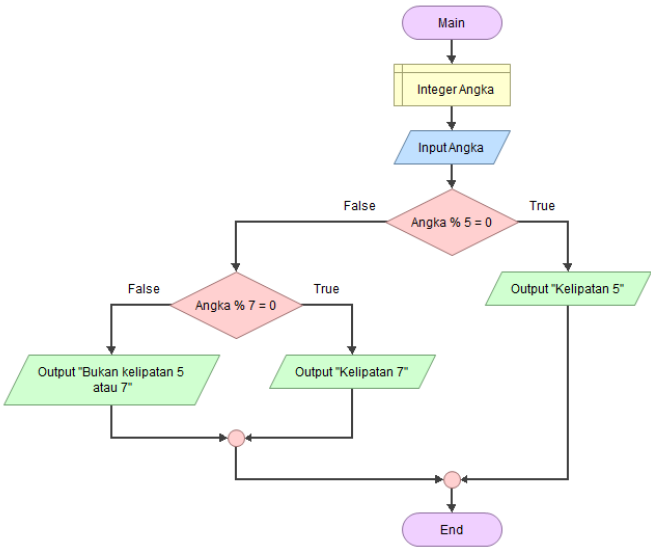
1	<code>import java.util.*;</code>
2	
3	<code>public class App {</code>
4	<code>public static void main(String[] args) throws Exception {</code>
5	<code>Scanner sc = new Scanner(System.in);</code>
6	<code>System.out.println("Masukkan nilai ujian: ");</code>
7	<code>int nilai = sc.nextInt();</code>
8	
9	<code>if(nilai >= 70) {</code>
10	<code>System.out.println("Selamat, Anda lulus ujian!");</code>


```
11     } else {
12         System.out.println("Maaf, Anda gagal!");
13     }
14 }
15 }
```

6.4. Percabangan Bertingkat (> 2 Kondisi)

Percabangan bertingkat terjadi apabila persyaratan / kondisi percabangan lebih dari dua. Percabangan bertingkat memiliki formasi "if – else if – else", dimana terdapat default 3 buah kondisi (masing-masing 1 kondisi untuk if, else if, dan else). Apabila terdapat 4 kondisi, maka yang perlu ditambah adalah jumlah "else if" nya menjadi 2. Sehingga diperoleh untuk jumlah "else if" = jumlah kondisi – 2.

Sebagai contoh, kita akan memasukkan sebuah bilangan, lalu bilangan tersebut diperiksa apakah bilangan tersebut merupakan kelipatan 5. Jika tidak, periksa kembali apakah merupakan kelipatan 7. Maka dalam pemetaan algoritma dalam bentuk flowchart digambarkan sebagai berikut.



Untuk if bertingkat, setiap yang bernilai false, akan diberi simbol if untuk memeriksa kondisi berikutnya jika kondisi sebelumnya tidak terpenuhi.

Sebagai praktikum, perhatikan tabel kode program berikut. Menurut Anda, bagaimana hasil outputnya?

Tabel 6.3. Kode Program Percabangan Bertingkat

```
1 import java.util.*;
2
3 public class App {
4     public static void main(String[] args) throws Exception {
5         Scanner sc = new Scanner(System.in);
6         System.out.println("Masukkan nilai: ");
7         int nilai = sc.nextInt();
8
9         if(nilai >= 85 && nilai <= 100) {
10            System.out.println("Nilai A");
11        } else if (nilai >= 70 && nilai < 85) {
12            System.out.println("Nilai B");
13        } else {
14            System.out.println("Nilai C");
15        }
16    }
17 }
```

Karena materi ini terbagi menjadi dua bagian, maka untuk asesmen dan tugas praktikum akan diberikan pada bab selanjutnya.



BAB 7

FUNGSI PERCABANGAN (Bagian 2)

7.1. Percabangan *If* Bersarang

Blok program *If* bersarang berisikan suatu keadaan dimana struktur kondisi terdapat sturktur kondisi juga. Suatu program mengizinkan percabangan *If* berada di dalam percabangan lainnya dan tidak membatasi jumlah percabangan tersebut. Jika diibaratkan, *if* bersarang ibarat sebuah bab dan didalamnya terdapat beberapa sub-bab, sementara setiap sub-bab terdapat beberapa sub-sub-bab, dan seterusnya.

Sebagai praktikum, perhatikan tabel kode program berikut. Menurut Anda, bagaimana hasil outputnya?

Tabel 7.1. Kode Program Percabangan *If* Bersarang

1	<code>import java.util.*;</code>
2	
3	<code>public class App {</code>
4	<code> public static void main(String[] args) throws Exception {</code>
5	<code> Scanner sc = new Scanner(System.in);</code>
6	<code> System.out.println("Masukkan nilai try out masing-masing</code>
	<code>mata pelajaran!!");</code>
7	<code> double b_indo, b_inggris, matematika, ipa, rata_rata;</code>
8	<code> System.out.print("Masukkan nilai Bahasa Indonesia: ");</code>
9	<code> b_indo = sc.nextDouble();</code>
10	<code> System.out.print("Masukkan nilai Bahasa Inggris: ");</code>
11	<code> b_inggris = sc.nextDouble();</code>
12	<code> System.out.print("Masukkan nilai Matematika: ");</code>
13	<code> matematika = sc.nextDouble();</code>
14	<code> System.out.print("Masukkan nilai IPA: ");</code>
15	<code> ipa = sc.nextDouble();</code>
16	
17	<code> rata_rata = (b_indo + b_inggris + matematika + ipa)/4;</code>
18	
19	

```

    if ((b_indo >= 0 && b_indo <= 10) && (b_inggris >= 0
20 && b_inggris <= 10) && (matematika >= 0 && matematika
    <= 10) && (ipa >= 0 && ipa <= 10)) {
        if (b_indo >= 4 && b_inggris >= 4 && matematika >=
21 4 && ipa >= 4) {
22     if (rata_rata >= 5.5) {
23         System.out.println("Selamat, Anda lulus!!");
24         System.out.println("Rata-rata: " + rata_rata);
25     } else {
26         System.out.println("Maaf, Anda tidak lulus!!");
27         System.out.println("Rata-rata: " + rata_rata);
28     }
        } else {
29         System.out.println("Maaf, Anda tidak lulus, ada nilai
    pelajaran yang di bawah 4");
30         System.out.println("Rata-rata: " + rata_rata);
31     }
32     } else {
33         System.out.println("Masukan nilai harus pada rentang 0
    - 10!");
34     }
35 }
36 }

```

7.2. Perintah *Switch-Case*

Kondisi switch-case adalah kondisi percabangan yang bertujuan untuk membandingkan isi dari variabel dengan beberapa nilai. Perintah yang wajib ada pada switch case yaitu:

- Perintah **switch**, yaitu terdapat nama variabel yang hendak diperiksa.
- Perintah **case**, yaitu perintah yang memiliki satu buah atau lebih untuk setiap nilai yang hendak diperiksa.

- Setiap blok case, terdapat perintah **break** untuk memutus program apabila kondisi terpenuhi.
- Perintah **default**, yaitu perintah yang akan dijalankan apabila semua kondisi case tidak terpenuhi.

Susunan kode program untuk switch-case yaitu:

```
switch (nama_variabel) {  
    case 'nilai 1':  
        (menjalankan program untuk nama_variabel == nilai 1)  
        break;  
    case 'nilai 2':  
        (menjalankan program untuk nama_variabel == nilai 2)  
        break;  
    ...  
    default:  
        (menjalankan program jika tidak ada kondisi yang terpenuhi)  
}
```

Susunan switch-case mirip dengan if bertingkat berikut:

```
if (nama_variabel == 1) {  
    (menjalankan program untuk nama_variabel == nilai 1)  
}  
else if (nama_variabel == 2) {  
    (menjalankan program untuk nama_variabel == nilai 2)  
}  
...  
else {  
    (menjalankan program jika tidak ada kondisi yang terpenuhi)  
}
```

Sebagai praktikum, perhatikan tabel kode program berikut.
Menurut Anda, bagaimana hasil outputnya?

Tabel 7.2. Kode Program *Switch-Case*

```
1 import java.util.*;
2
3 public class App {
4     public static void main(String[] args) throws Exception {
5         Scanner sc = new Scanner(System.in);
6         byte menu;
7         double bil1, bil2, tambah, kurang, kali, bagi;
8         System.out.print("Masukkan bilangan 1: ");
9         bil1 = sc.nextDouble();
10        System.out.print("Masukkan bilangan 2: ");
11        bil2 = sc.nextDouble();
12        System.out.println("Pilih menu: \n1. Penjumlahan \n2.
Pengurangan \n3. Perkalian \n4. Pembagian");
13        menu = sc.nextByte();
14
15        switch (menu) {
16            case 1:
17                tambah = bil1 + bil2;
18                System.out.println("Hasil penjumlahan = " +
19 tambah);
20                break;
21            case 2:
22                kurang = bil1 - bil2;
23                System.out.println("Hasil pengurangan = " +
24 kurang);
25                break;
26            case 3:
27                kali = bil1 * bil2;
```

```
28         System.out.println("Hasil perkalian = " + kali);
29         break;
30     case 4:
31         if (bil2 != 0) {
32             bagi = bil1 / bil2;
33             System.out.println("Hasil pembagian = " + bagi);
34         } else {
35             System.out.println("Pembagian dengan nol tidak
36             terdefinisi!");
37         }
38         break;
39     default:
40         System.out.println("Menu yang dimasukkan
41         salah!");
42         break;
43     }
44 }
```


ASESMEN

Jawablah pertanyaan di bawah ini dengan benar!

1. Berikut ini syntax yang digunakan pada percabangan yaitu ...
(jawaban dapat lebih dari satu)

- if
- if-else
- for
- switch-case

2. Untuk percabangan satu kondisi apabila persyaratan tidak dipenuhi maka hasil outputnya yaitu ...

3. Bagaimanakah algoritma untuk percabangan satu kondisi?

.....

.....

.....

.....

.....

Perhatikan kode program di bawah ini untuk soal nomor 4 – 5.

1	import java.util.*
2	
3	public class App {
4	public static void main(String[] args) throws Exception {
5	Scanner sc = new Scanner(System.in);
6	System.out.println("Masukkan usia Anda: ");
7	int usia = sc.nextInt();
8	
9	if(usia >= 17) {
10	

	System.out.println("Usia Anda " + usia + " tahun,
11	Anda boleh membuat KTP");
12	}
13	}
	}

4. Berdasarkan informasi berikut, berilah pernyataan **Benar** atau **Salah**.

Pernyataan	Benar/Salah
Program tersebut merupakan percabangan satu dimensi.	...
Jika memasukkan usia 17,5 tahun, maka outputnya "Usia Anda 17,5 tahun, Anda boleh membuat KTP".	...
Syarat yang diperoleh dari program di atas yaitu usia minimal 17 tahun untuk membuat KTP.	...
Jika memasukkan usia kurang dari 17 tahun, maka tidak ada hasil pada output.	...

5. Proses persyaratan pada program di atas terdapat pada baris ...
- Baris 1
 - Baris 3-5
 - Baris 9-11
 - Baris 5-7

Perhatikan kode program di bawah ini untuk soal nomor 6 – 8.

1	import java.util.*
2	
3	public class App {
4	public static void main(String[] args) throws Exception {
5	Scanner sc = new Scanner(System.in);
6	System.out.println("Masukkan sebuah angka: ");

```
7     int angka = sc.nextInt();
8
9     if(angka % 2 == 0 && angka > 200) {
10        System.out.println("Angka yang dimasukkan
memenuhi syarat");
11    }
12 }
13 }
```

Berdasarkan kode di atas, angka 200 termasuk angka yang memenuhi syarat.

6. Setujukah Anda dengan pernyataan berikut?

.....
.....
.....
.....
.....

7. Berikan alasannya!

.....
.....
.....
.....
.....

8. Apabila angka 261 memenuhi syarat, maka baris 9 diubah menjadi ... (*jawaban dapat lebih dari satu*)

- if(angka % 2 == 1 && angka > 200)
- if(angka % 3 == 0 && angka > 200)
- if(angka % 4 == 1 && angka >= 261)
- if(angka % 2 == 1 && angka > 261)

Perhatikan kode program di bawah ini untuk soal nomor 9 – 10.

```
1  import java.util.*
2
3  public class App {
4      public static void main(String[] args) throws Exception
5  {
6      Scanner sc = new Scanner(System.in);
7      System.out.println("Masukkan sebuah angka: ");
8      int angka = sc.nextInt();
9
10     if(1) {
11         System.out.println("Angka yang dimasukkan
12     memenuhi syarat");
13     }
14 }
```

9. Untuk membuat persyaratan agar angka antara 100 dan 200 memenuhi syarat, maka pernyataan yang tepat untuk angka (1) yaitu ...
- angka > 100 && angka < 200
 - angka >= 100 && angka < 200
 - angka >= 100 && angka <= 200
 - angka > 100 && angka <= 200
10. Pasangkan pernyataan untuk mengisi angka (1) di kode tersebut sesuai dengan persyaratannya.

angka%10 == 2	(1)	(a)	angka maksimal 430
angka <= 430	(2)	(b)	angka harus berakhiran 2
angka%6 == 0 && angka > 300	(3)	(c)	angka harus selain 10 sampai 50
angka < 10 angka > 50	(4)	(d)	Angka berkelipatan 6 diatas 300

11. Syntax pada percabangan dua kondisi yang digunakan sebagai alternatif lain apabila kondisi persyaratan tidak terpenuhi yaitu...

12. Bagaimanakah algoritma untuk percabangan dua kondisi?

.....

.....

.....

.....

.....

Perhatikan kode program di bawah ini untuk soal nomor 13 – 15.

```

1  import java.util.*
2
3  public class App {
4      public static void main(String[] args) throws Exception {
5          Scanner sc = new Scanner(System.in);
6          System.out.println("Masukkan nama: ");
7          String nama = sc.next();
8          System.out.println("Masukkan nomor ujian: ");
9          int no_ujian = sc.nextInt();

```

```

10     System.out.println("Masukkan jumlah benar: ");
11     int benar = sc.nextInt();
12     System.out.println("Masukkan jumlah salah: ");
13     int salah = sc.nextInt();
14
15     System.out.println("----- ");
16     System.out.println("HASIL UJIAN MASUK SMA NEGERI
17 SWASTA");
18     System.out.println("----- ");
19
20     System.out.println("Jumlah benar: " + benar);
21     System.out.println("Jumlah salah: " + salah);
22     System.out.println("Jumlah kosong: " + (50-benar-
23 salah));
24
25     int nilai = benar * 4 - salah;
26     System.out.println("Nilai Anda: " + nilai);
27
28     if(nilai >= 100) {
29         System.out.println("Selamat " + nama + ", Anda
dinyatakan lulus ujian masuk");
30     } else {
31         System.out.println("Maaf " + nama + ", Anda
dinyatakan tidak lulus ujian masuk");
32     }
    }
}

```

13. Berdasarkan informasi berikut, berilah pernyataan **Benar** atau **Salah**.

Pernyataan	Benar/Salah
Seorang peserta calon siswa SMA Negeri Swasta dinyatakan lulus jika peserta menjawab sekurang-kurangnya 25 soal dengan benar tanpa ada jawaban yang salah.	...
Skor maksimal ujian masuk SMA Negeri Swasta yaitu 200 poin.	...
Jono mengikuti ujian masuk dengan hasil 27 jawaban benar dan 9 jawaban salah. Maka Jono dinyatakan lulus ujian SMA Negeri Swasta.	...
Hana mengikuti ujian masuk dan menjawab semua soal dengan perbandingan jumlah jawaban benar dan salah yaitu 3 : 2. Maka Hana dinyatakan lulus ujian SMA Negeri Swasta.	...

14. Eka dan Doni mengikuti ujian masuk SMA Negeri Swasta. Keduanya memiliki nilai yang sama dan bernilai 12 lebihnya dari ambang batas kelulusan. Eka mampu mengerjakan beberapa soal dengan benar tanpa kesalahan satu soal pun, sementara Doni memiliki perbandingan jawaban benar dan salah yaitu 2 : 1. Total jawaban kosong dari Eka dan Doni yaitu sebanyak ... soal.
- 2
 - 22
 - 16
 - 24

15. Jika ada syarat tambahan, yaitu peserta lulus ujian masuk apabila peserta sebanyak-banyaknya mempunyai 15 jawaban yang salah, maka kode pada baris 26 dimodifikasi menjadi ...

Perhatikan kode program di bawah ini untuk soal nomor 16 – 17.

```
1 import java.util.*
2
3 public class App {
4     public static void main(String[] args) throws Exception {
5         Scanner sc = new Scanner(System.in);
6         System.out.println("Masukkan total harga: ");
7         int harga = sc.nextInt();
8         System.out.println("Masukkan total item barang: ");
9         int item = sc.nextInt();
10
11         System.out.println("----- ");
12         System.out.println("RINCIAN TRANSAKSI");
13         System.out.println("----- ");
14
15         System.out.println("Total harga: " + harga);
16         System.out.println("Total item barang: " + item);
17
18         if(harga >= 500000 || item > 7) {
19             System.out.println("Anda mendapat potongan harga
20 sebesar 10%");
21             System.out.println("Anda cukup membayar " + (1));
22         } else {
23             System.out.println("Anda tidak mendapat diskon");
24         }
25     }
```


16. Kode yang tepat untuk mengisi angka (1) pada baris 20 yaitu ...
(jawaban dapat lebih dari satu)

- harga * 0.9
- harga – (harga * 0.1)
- harga * (1 – 0.1)
- harga + (harga * 0.1)

17. Pasangkan beberapa transaksi tersebut dengan harga yang dibayar sesuai dengan persyaratan yang ditetapkan.

harga 600.000, 7 item	(1)	(a)	450.000
harga 750.000, 9 item	(2)	(b)	470.000
harga 470.000, 10 item	(3)	(c)	675.000
harga 500.000, 8 item	(4)	(d)	600.000

Perhatikan kode program di bawah ini untuk soal nomor 18 – 20.

```

1  import java.util.*
2
3  public class App {
4      public static void main(String[] args) throws Exception {
5          Scanner sc = new Scanner(System.in);
6          System.out.println("Masukkan panjang lapangan: ");
7          double panjang = sc.nextDouble();

```

```
8      System.out.println("Masukkan lebar lapangan: ");
9      double lebar = sc.nextDouble();
10     double keliling = (1);
11     double harga;
12
13     if(2) {
14         harga = (3);
15         System.out.println("Total harga: " + harga);
16     } else {
17         harga = (4);
18         System.out.println("Total harga: " + harga);
19     }
20 }
21 }
```

Program di atas yaitu program menghitung total biaya pemasangan pagar lapangan. Apabila keliling lapangan kurang dari 100 m, harga pemasangan pagar Rp10.000 per meter. Sedangkan apabila lebih atau sama dengan 100 m, harga pemasangan pagar Rp15.000 per meter.

18. Lengkapi kode program bernomor (1) sampai (4) di atas.

.....
.....
.....
.....
.....

19. Berdasarkan informasi berikut, berilah pernyataan **Benar** atau **Salah**.

Pernyataan	Benar/Salah
Jika lapangan berbentuk persegi, maka panjang sisi minimum untuk biaya pemasangan pagar Rp15.000 per meter yaitu 25 m.	...
Jika panjang kebun 20 m dan lebar kebun 29 m, maka harga pemasangan pagar yaitu Rp1.470.000.	...
Jika kebun berbentuk persegi dengan luas 400 m ² , maka berlaku ketentuan biaya pemasangan pagar sebesar Rp10.000 per meter.	...
Jika kebun berbentuk persegi dengan luas 900 m ² , maka harga pemasangan pagar yaitu Rp1.800.000.	...

20. Menurut Anda, mengapa variabel panjang, lebar, keliling, dan harga termasuk ke dalam tipe data double?

.....

.....

.....

.....

.....

21. Banyaknya perintah "else if" yang digunakan apabila ingin membuat 50 kondisi percabangan adalah ...

- a. 49
- b. 48
- c. 50
- d. 51

22. Urutan formasi untuk percabangan lebih dari dua kondisi yaitu...

Perhatikan kode program di bawah ini untuk soal nomor 23 – 24.

```
1 import java.util.*
2
3 public class App {
4     public static void main(String[] args) throws Exception {
5         Scanner sc = new Scanner(System.in);
6         System.out.println("Masukkan poin membership: ");
7         int poin = sc.nextInt();
8
9         if (poin >= 100 && poin < 200) {
10            System.out.println("Anda merupakan member
11 silver");
12        } else if (poin >= 200 && poin < 500){
13            System.out.println("Anda merupakan member
14 gold");
15        } else if (poin >= 500){
16            System.out.println("Anda merupakan member
17 diamond");
18        } else {
19            System.out.println("Anda merupakan member
classic");
        }
    }
}
```

23. Pasangkan beberapa poin membership berikut sesuai dengan tingkatannya.

88 poin	(1)
375 poin	(2)
100 poin	(3)
550 poin	(4)

(a)	diamond
(b)	classic
(c)	gold
(d)	silver

24. Jika terdapat aturan terbaru bahwa member diamond diberikan bagi customer yang memiliki membership poin minimal 500, dengan masa aktif membership lebih dari 2 tahun, maka baris program yang perlu diubah yaitu ...

- a. 9
- b. 11
- c. 13
- d. 15

Perhatikan kode program di bawah ini untuk soal nomor 25 – 27.

```
1 import java.util.*
2
3 public class App {
4     public static void main(String[] args) throws Exception {
5         Scanner sc = new Scanner(System.in);
6         System.out.println("Jumlah halaman yang dicetak: ");
7         int halaman = sc.nextInt();
8         int harga;
9
10        if (1) {
11            harga = (2)
12            System.out.println("Harga cetak = " + harga);
13        } else if (3) {
```

```
14     harga = (4)
15     System.out.println("Harga cetak = " + harga);
16     } else if (5) {
17     harga = (6)
18     System.out.println("Harga cetak = " + harga);
19     } else {
20     (7)
21     }
22 }
23 }
```

Program di atas yaitu program menghitung total biaya pencetakan kertas tanpa warna (hitam putih). Harga per lembar kertas yaitu Rp300. Apabila mencetak lebih dari 500 lembar kertas, harga cetak sebesar Rp275 per lembar. Apabila mencetak lebih dari 1000 lembar kertas, harga cetak sebesar Rp250 per lembar. Penghitungan harga berlaku apabila memasukkan jumlah halaman minimal 0, jika dibawah itu akan menampilkan keterangan yang menyatakan bahwa jumlah halaman yang dimasukkan salah.

25. Lengkapi kode program bernomor (1) sampai (7) di atas.

.....

.....

.....

.....

.....

26. Berdasarkan informasi berikut, berilah pernyataan **Benar** atau **Salah**.

Pernyataan	Benar/Salah
Harga cetak 460 lembar kertas lebih murah daripada harga cetak 503 lembar kertas.	...
Harga cetak 495 lembar kertas sama dengan harga cetak 540 lembar kertas.	...
Harga cetak 980 lembar kertas lebih murah daripada harga cetak 1040 lembar kertas.	...
Harga cetak 995 lembar kertas lebih mahal daripada harga cetak 1080 lembar kertas.	...

27. Pasangkan banyaknya lembar kertas yang dicetak berikut sesuai dengan harganya.

400	(1)	(a)	jumlah tidak sesuai
750	(2)	(b)	262.500
-100	(3)	(c)	120.000
1050	(4)	(d)	206.250

Perhatikan kode program di bawah ini untuk soal nomor 28 – 30.

```

1  Import java.util.*
2
3  public class App {
4      public static void main(String[] args) throws Exception
5  {
6      Scanner sc = new Scanner(System.in);
7      System.out.println("Masukkan angka: ");
8      int angka = sc.nextInt();
9
10     if(angka%7 == 0 && angka%9 == 0) {
11

```

```
12     System.out.println(angka + " adalah kelipatan 7
13 dan 9");
14     } else if (angka%7 == 0) {
15         System.out.println(angka + " adalah kelipatan 7");
16     } else if (angka%9 == 0) {
17         System.out.println(angka + " adalah kelipatan 9");
18     } else {
19         System.out.println(angka + " bukan kelipatan 7
maupun 9");
    }
    }
}
```

Berdasarkan kode di atas, maka bilangan positif terkecil untuk pernyataan "bilangan kelipatan 7 dan 9" yaitu 63.

28. Setujukah Anda dengan pernyataan berikut?

.....
.....
.....
.....
.....

29. Berikan alasannya!

.....
.....
.....
.....
.....

30. Pasangkan bilangan berikut sesuai dengan pernyataannya.

90	(1)	(a)	kelipatan 7 dan 9
70	(2)	(b)	kelipatan 7
252	(3)	(c)	bukan kelipatan 7 dan 9
87	(4)	(d)	kelipatan 9

31. Menurut Anda, apakah perbedaan antara percabangan bersarang dengan percabangan bertingkat?

.....

.....

.....

.....

.....

Perhatikan kode program di bawah ini untuk soal nomor 32 – 34.

```
1  Import java.util.*
2
3  public class App {
4      public static void main(String[] args) throws Exception {
5          Scanner sc = new Scanner(System.in);
6          System.out.println("Masukkan angka: ");
7          int angka = sc.nextInt();
8
9          if(angka > 0 && angka <= 200) {
10             if (angka %3 == 0) {
11
```

```

12         System.out.println(angka + " adalah kelipatan
13 3");
14     } else {
15         System.out.println(angka + " bukan kelipatan
16 3");
17     }
18     } else if (angka > 200 && angka <= 500) {
19         if (angka %2 == 0 && angka %5 == 0) {
20             System.out.println(angka + " adalah kelipatan 2
21 dan 5");
22         } else {
23             System.out.println(angka + " bukan kelipatan 2
24 dan 5");
25         }
26     } else {
27         System.out.println("Angka harus 1 sampai 500");
28     }
29 }

```

32. Apabila memasukkan bilangan 300 pada output, maka hasil yang ditampilkan yaitu ...
- 300 adalah kelipatan 2 dan 5
 - 300 adalah kelipatan 3
 - 300 bukan kelipatan 2 dan 5
 - Angka yang dimasukkan harus 1 sampai 500
33. Bilangan terkecil yang memungkinkan untuk menampilkan output kelipatan 2 dan 5 yaitu ...

34. Pasangkan bilangan berikut sesuai dengan pernyataannya.

120	(1)	(a)	kelipatan 2 dan 5
300	(2)	(b)	bukan kelipatan 3
130	(3)	(c)	angka harua 1 sampai 500
520	(4)	(d)	kelipatan 3

Perhatikan kode program di bawah ini untuk soal nomor 35 – 37.

```
1  Import java.util.*
2
3  public class App {
4      public static void main(String[] args) throws Exception {
5          Scanner sc = new Scanner(System.in);
6          System.out.println("Masukkan jari-jari: ");
7          double jarijari = sc.nextDouble();
8          System.out.println("Masukkan tinggi: ");
9          double tinggi = sc.nextDouble();
10         double volume;
11
12         if(1) {
13             if (2) {
14                 volume = 22 / 7 * jarijari * jarijari * tinggi;
15                 System.out.println("Volume drum: " + volume);
16             } else {
17                 volume = 3.14 * jarijari * jarijari * tinggi;
18                 System.out.println("Volume drum: " + volume);
19             }
16         }
```

```
20     } else {
21         System.out.println("jari-jari dan volume harus diatas
22 nol");
23     }
24 }
}
```

Program di atas menghitung volume drum minyak. Jika jari-jari atau tinggi kelipatan 7, menggunakan konstanta phi $22/7$. Jika bukan kelipatan 7, maka menggunakan konstanta phi 3,14. Perlu diingat bahwa masukan jari-jari dan tinggi harus diatas nol.

35. Lengkapi kode program bernomor (1) sampai (2) di atas.

.....
.....
.....
.....
.....

36. Input jari-jari dan tinggi yang memungkinkan untuk menggunakan konstanta phi $22/7$ yaitu ... (*jawaban dapat lebih dari satu*)

- jarijari = 12, tinggi = 35
- jarijari = 10, tinggi = 20
- jarijari = 77, tinggi = 40
- jarijari = 56, tinggi = 49

37. Jika diketahui panjang jari-jari suatu tabung 3,5 dm dan tinggi tabung 7,7 dm, maka berdasarkan program di atas, volume tabung yaitu ... dm^3 .

- a. 296,45
- b. 94,325

- c. 296,1805
- d. 13,475

Perhatikan kode program di bawah ini untuk soal nomor 38 – 40.

```
1  Import java.util.*
2
3  public class App {
4      public static void main(String[] args) throws Exception {
5          Scanner sc = new Scanner(System.in);
6          System.out.print("Masukkan total harga: ");
7          int harga = sc.nextInt();
8          System.out.print("Pilih metode pembayaran:\n1.
Tunai\n2. QRIS\nPilihan Anda: ");
9          int pilihan = sc.nextInt();
10         int harga_final;
11
12         if(pilihan == 1) {
13             if (harga > 300000 && harga <= 500000) {
14                 harga_final = harga * 0.9;
15                 System.out.println("Harga yang dibayar: " +
16 harga_final);
17             } else if (harga > 500000) {
18                 harga_final = harga * 0.8;
19                 System.out.println("Harga yang dibayar: " +
20 harga_final);
21             } else {
22                 harga_final = harga;
23                 System.out.println("Harga yang dibayar: " +
24 harga_final);
25             }
26         } else if(pilihan == 2) {
```

```

27     if (harga > 300000 && harga <= 500000) {
28         harga_final = harga * 0.85;
29         System.out.println("Harga yang dibayar: " +
30 harga_final);
31     } else if (harga > 500000) {
32         harga_final = harga * 0.75;
33         System.out.println("Harga yang dibayar: " +
34 harga_final);
35     } else {
36         harga_final = harga;
37         System.out.println("Harga yang dibayar: " +
38 harga_final);
39     }
40     } else {
41         System.out.println("Pilihan pembayaran tidak
tersedia");
42     }
43 }

```

38. Berdasarkan informasi berikut, berilah pernyataan **Benar** atau **Salah**.

Pernyataan	Benar/Salah
Untuk transaksi diatas Rp300.000, pihak supermarket memberikan potongan sebesar 15% bagi pembayaran via QRIS.	...
Transaksi diatas Rp500.000, pihak supermarket memberikan potongan 1/5 harga bagi pembayaran tunai.	...

Transaksi dibawah Rp300.000, pihak supermarket memberikan harga penuh bagi pembayaran via QRIS.	...
Transaksi diatas Rp500.000, pihak supermarket memberikan pembayaran 1/4 harga bagi pembayaran QRIS.	...

39. Agar variabel pilihan pada baris 9 menjadi efektif, maka tipe data dapat diganti menjadi tipe data ...
40. Pasangkan transaksi dan metode pembayaran berikut sesuai harga akhirnya.

350.000, QRIS	(1)	(a)	360.000
600.000, tunai	(2)	(b)	297.500
540.000, QRIS	(3)	(c)	405.000
400.000, tunai	(4)	(d)	480.000

41. Pada perintah switch-case, perintah yang wajib ada pada setiap blok case untuk menghentikan program apabila kondisi terpenuhi yaitu ...
42. Urutan kode program fungsi percabangan switch-case yang benar adalah ...
- switch – break – default – case
 - switch – case – default – break
 - switch – case – break – default
 - switch – default – case – break

Perhatikan kode program di bawah ini untuk soal nomor 43 – 45.

```
1  Import java.util.*
2
3  public class App {
4      public static void main(String[] args) throws Exception {
5          Scanner sc = new Scanner(System.in);
6          System.out.print("Masukkan panjang sisi: ");
7          double sisi = sc.nextDouble();
8          System.out.print("1. Hitung volume\n2. Hitung luas
permukaan\nPilihan Anda: ");
9          byte pilihan = sc.nextByte();
10         int hasil;
11
12         if (1) {
13             switch(pilihan) {
14                 case 1:
15                     hasil = (2);
16                     System.out.println("Volume kubus: " + hasil);
17                     break;
18                 case 2:
19                     hasil = (3);
20                     System.out.println("Luas permukaan kubus: " +
21 hasil);
22                     break;
23                 default:
24                     System.out.println("Pilihan menu tidak
25 tersedia");
26                     break;
27             }
28         } else {
29
```


30	<pre>System.out.println("Panjang sisi harus lebih dari nol!"); } } }</pre>
----	--

43. Lengkapi kode program bernomor (1) sampai (3) di atas.

.....
.....
.....
.....
.....

44. Variabel sisi yang harus dimasukkan apabila hasil pada pilihan 1 dan 2 bernilai sama yaitu ...

45. Jika memasukkan 10 pada variabel sisi dan 3 pada variabel pilihan, maka hasil yang ditampilkan adalah ...

- a. Volume kubus: 1000
- b. Luas permukaan kubus: 1000
- c. Luas permukaan kubus: 600
- d. Pilihan menu tidak tersedia

Perhatikan pernyataan berikut ini:

Perintah switch-case mirip dengan percabangan bertingkat. Namun, pada switch-case, tidak diizinkan menggunakan operator perbandingan, ataupun operator logika.

46. Setujukah Anda dengan pernyataan berikut?

.....
.....
.....
.....
.....

47. Berikan alasannya!

.....
.....
.....
.....
.....

Perhatikan kode program di bawah ini untuk soal nomor 48 – 50.

```
1  Import java.util.*
2
3  public class App {
4      public static void main(String[] args) throws Exception
5  {
6      Scanner sc = new Scanner(System.in);
7      System.out.print("Pilih jenis kamar: \n1. Regular\n2.
8  Executive\n3. VVIP: ");
9      byte kamar = sc.nextByte();
10     System.out.print("Masukkan lama menginap: ");
11     int hari = sc.nextInt();
12     System.out.print("Masukkan banyak extra bed: ");
13     int extra = sc.nextInt();
14     int tarif;
15     if (hari > 0 && extra >= 0) {
```

```

16         switch(kamar) {
17             case 1:
18                 tarif = hari * 250000 + extra * 45000;
19                 System.out.println("Tarif penginapan: " +
20 tarif);
21                 break;
22             case 2:
23                 tarif = hari * 325000 + extra * 55000;
24                 System.out.println("Tarif penginapan: " +
25 tarif);
26                 break;
27             case 3:
28                 tarif = hari * 400000 + extra * 75000;
29                 System.out.println("Tarif penginapan: " +
30 tarif);
31                 break;
32             default:
33                 System.out.println("Pilihan menu tidak
34 tersedia");
35                 break;
36         }
        } else {
            System.out.println("Isian hari dan extra bed tidak
sesuai");
        }
    }
}

```

48. Berdasarkan informasi berikut, berilah pernyataan **Benar** atau **Salah**.

Pernyataan	Benar/Salah
Setiap tingkatan jenis kamar, harga penginapan bertambah Rp75.000 untuk setiap malamnya.	...
Setiap tingkatan jenis kamar, harga extra bed bertambah Rp20.000 untuk setiap malamnya.	...
Apabila memasukkan variabel kamar dengan 4, maka akan menjalankan perintah default.	...
Apabila memasukkan variabel hari dengan 0 dan variabel extra dengan 0, maka tidak ada perhitungan tarif yang dilakukan.	...

49. Budi ingin menginap di hotel tersebut. Dengan budget maksimal Rp1.000.000, maka alternatif Budi untuk memilih kamar beserta lama ia menginap adalah ... *(jawaban dapat lebih dari satu)*

- kamar = 3, hari = 2, extra = 2
- kamar = 1, hari = 4, extra = 0
- kamar = 2, hari = 3, extra = 1
- kamar = 2, hari = 2, extra = 2

50. Pasangkan kamar yang dipilih, lama penginapan, dan extra bed sesuai dengan tarif yang harus dibayarkan pengunjung.

kamar = 3, hari = 1, extra = 3	(1)	(a)	2.005.000
kamar = 1, hari = 7, extra = 0	(2)	(b)	590.000

kamar = 2, hari = 6, extra = 1	(3)
kamar = 1, hari = 2, extra = 2	(4)

(c)	1.750.000
(d)	625.000

SOAL PRAKTIKUM

1. Buatlah sebuah program dengan bahasa Java yang menerima masukan tiga buah bilangan bulat berupa panjang sisi segitiga, kemudian periksalah apakah segitiga itu termasuk segitiga sama sisi atau bukan.
2. **(NIM Ganjil)** Buatlah sebuah program dengan kasus sebagai berikut:
Sebuah LBB TOEFL baru saja mengadakan tes TOEFL kepada mahasiswa untuk menjanging manakah yang dinyatakan lulus atau harus mengikuti kursus tambahan. Ketentuan dari kasus ini antara lain:
 - Rentang nilai tes TOEFL yaitu 310 sampai 677.
 - Angka terakhir nilai tes TOEFL harus 0, 3, atau 7. (**HINT:** gunakan mod 10)
 - Jika nilai tes minimal 477 maka dinyatakan lulus, jika nilai tes 400 sampai 473 maka wajib mengikuti kursus pengayaan, dan jika nilai dibawah 400 maka wajib mengikuti kursus pemula dan pengayaan.
3. **(NIM Genap)** Buatlah sebuah program dengan kasus sebagai berikut:

Pak Suparman ingin membuat sebuah kolam renang di lapangan belakang rumahnya. Ia ingin membuat sebuah informasi mengenai ukuran kolam beserta durasi pengisian air, dan biaya pemasangan keramik. Ketentuan dari kasus ini antara lain:

- Program ini terdiri dari 2 menu, yaitu menu "durasi pengisian air" dan menu "biaya pemasangan keramik"
- Menu nomor 1 yaitu menu "durasi pengisian air". Menu ini berisi informasi tentang volume kolam renang (berdasarkan inputan panjang, lebar, kedalaman kolam), beserta durasi pengisian air (berdasarkan debit air dalam satuan liter per detik).
- Menu nomor 2 yaitu menu "biaya pemasangan keramik". Menu ini berisi informasi tentang biaya pemasangan keramik pada kolam renang dengan asumsi biaya pemasangan Rp80.000 per meter persegi.



BAB 8

FUNGSI PERULANGAN (Bagian 1)

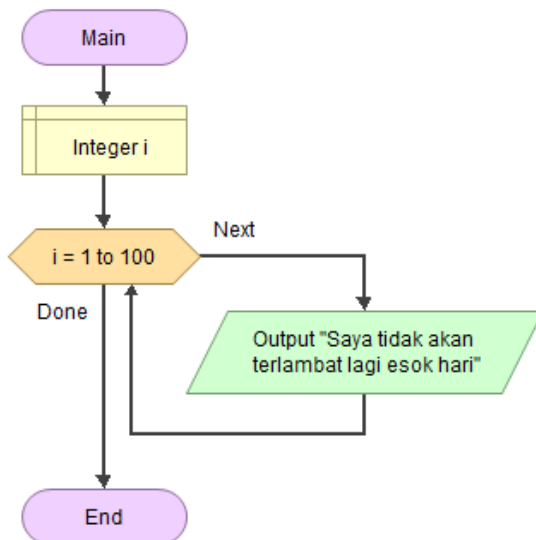
```
eslintConfig  
=> {  
  "extends": [  
    "react-app",  
    "react-app/"  
  ],  
  closeOnEscapeKeyDown);
```

8.1. Perulangan (Looping)

Perulangan (*Looping*) yaitu suatu blok program yang bertugas untuk melakukan kegiatan secara berulang dari sebuah proses sesuai dengan yang diinginkan. Bagian-bagian perulangan yang harus dipenuhi yaitu:

- **Inisialisasi**, yaitu tahap persiapan membuat kondisi awal sebelum dilakukan perulangan.
- **Proses**, yaitu tahap dilakukannya sebuah proses yang perlu dilakukan secara berulang-ulang.
- **Iterasi**, terjadi di dalam perulangan yang merupakan proses penambahan agar perulangan dapat terus berjalan.
- **Terminasi**, yaitu kondisi penghentian dari perulangan agar tidak menjadi perulangan tanpa henti.

Sebagai contoh, siswa datang terlambat ke sekolah dan menerima hukuman menuliskan pernyataan "Saya tidak akan terlambat lagi esok hari" sebanyak 100 kali. Algoritma yang digambarkan berupa flowchart yaitu sebagai berikut:



8.2. Perulangan For

Perintah perulangan *for* digunakan untuk membuat suatu proses pengulangan yang **sudah jelas perlu diulang berapa kali**. Dengan kata lain, jumlah perulangan yang diperlukan sudah diketahui dari si pembuat program, mulai dari angka awal berapa ke angka akhir berapa, batasannya sudah jelas.

Ada dua fungsi penghitungan pada perintah *for*, yaitu:

- Perulangan increment (++)

Kode program:

```
for (int i = 1; i <= 10; i++) {  
    System.out.print(i + " ");  
}
```

Output: 1 2 3 4 5 6 7 8 9 10

- Perulangan decrement (--)

Kode program:

```
for (int i = 12; i >= 0; i--) {  
    System.out.print(i + " ");  
}
```

Output: 12 11 10 9 8 7 6 5 4 3 2 1 0

Sebagai praktikum, perhatikan tabel kode program berikut. Menurut Anda, bagaimana hasil outputnya?

Tabel 8.1. Kode Program Perulangan For

1	public class App {
2	public static void main(String[] args) throws Exception {
3	for (int a = 1; a <= 10; a++) {
4	System.out.println("Selamat datang");
5	}
6	}
7	}

Tabel 8.2. Kode Program Perulangan For Hitung Maju

1	public class App {
2	public static void main(String[] args) throws Exception {
3	for (int a = 1; a <= 20; a++) {
4	System.out.print(a + " ");
5	}
6	System.out.println();
7	for (int a = 20; a >= 1; a--) {
8	System.out.print(a + " ");
9	}
10	}
11	}

Tabel 8.3. Kode Program Perulangan For Lompat Bilangan

1	public class App {
2	public static void main(String[] args) throws Exception {
3	for (int a = 1; a <= 20; a = a + 2) {
4	System.out.print(a + " ");
5	}
6	System.out.println();
7	for (int a = 20; a >= 1; a = a - 2) {
8	System.out.print(a + " ");
9	}
10	}
11	}

Karena materi ini terbagi menjadi dua bagian, maka untuk asesmen dan tugas praktikum akan diberikan pada bab selanjutnya.



BAB 9

FUNGSI PERULANGAN (Bagian 2)

9.1. Perulangan While

While, yaitu sebuah pernyataan looping pada Java yang digunakan untuk mengeksekusi kode program sampai kondisi tertentu. Potongan kode program while ditulis seperti ini:

```
while (boolean_expression) {  
    //statement  
}
```

Dari potongan kode tersebut, statement / pernyataan akan terus berulang selama boolean_expression bernilai "true". Untuk menghentikan perulangan, maka perlu mengubah boolean_expression menjadi "false".

Sebagai praktikum, perhatikan tabel kode program berikut. Menurut Anda, bagaimana hasil outputnya?

Tabel 9.1. Kode Program Perulangan While

1	public class App {
2	public static void main(String[] args) throws Exception {
3	int ulang = 1;
4	
5	while (ulang <= 20) {
6	System.out.print(ulang + " ");
7	ulang++;
8	}
9	}
10	}

9.2. Perulangan Do-While

Konsep do-while sama seperti while. Bedanya pada perintah do-while, sebuah pernyataan dieksekusi dulu baru dilakukan perulangan. Kode program do-while:

```
do {
```

```
    //statement
} while (boolean_expression)
```

Sebagai praktikum, perhatikan tabel kode program berikut. Menurut Anda, bagaimana hasil outputnya?

Tabel 9.2. Kode Program Perulangan Do-While

1	public class App {
2	public static void main(String[] args) throws Exception {
3	int ulang = 0;
4	do {
5	System.out.println(ulang);
6	ulang++;
7	} while (ulang < 10);
8	}
9	}

9.3. Nested Loop

Perintah looping dapat dibuat secara bertingkat seperti halnya perintah if. Pada umumnya, perintah nested loop digunakan untuk membuat formasi bangun datar seperti persegi, persegi panjang, segitiga, dan sebagainya.

Sebagai praktikum, perhatikan tabel kode program berikut. Menurut Anda, bagaimana hasil outputnya?

Tabel 9.3. Kode Program Nested Loop

1	public class App {
2	public static void main(String[] args) throws Exception {
3	for (int a = 1; a <= 10; a++) {
4	for (int b = 1; b <= 5; b++) {
5	System.out.print("*");
6	}

7	System.out.println();
8	}
9	}
10	}

ASESMEN

Jawablah pertanyaan di bawah ini dengan benar!

1. Berikut ini syntax yang digunakan pada perulangan yaitu ...
(jawaban dapat lebih dari satu)
 - while
 - do-while
 - for
 - switch-case

2. Kondisi penghentian dari perulangan agar tidak terjadi perulangan terus menerus yaitu ...
 - a. terminasi
 - b. inisialisasi
 - c. proses
 - d. iterasi

Perhatikan kode program di bawah ini untuk soal nomor 3 – 5.

1	public class App {
2	public static void main(String[] args) throws Exception {
3	for (int a = 1; a < 10; a++) {
4	System.out.println("Materi Algoritma
5	Pemrograman");

6	}
7	}
	}

Kode di atas akan menampilkan output berupa kalimat "Materi Algoritma Pemrograman" sebanyak 10 kali.

3. Setujukah Anda dengan pernyataan berikut?

.....
.....
.....
.....
.....
.....

4. Berikan alasannya!

.....
.....
.....
.....
.....
.....

5. Dari kode program di atas, tentukan mana yang termasuk inisialisasi, proses, iterasi, dan terminasi.

.....
.....
.....
.....
.....
.....

Perhatikan kode program di bawah ini untuk soal nomor 6 – 8.

```
1 public class App {
2     public static void main(String[] args) throws Exception {
3         for (int a = 1; a <= 100; a++) {
4             System.out.print(a + " ");
5         }
6     }
7 }
```

6. Berdasarkan informasi berikut, berilah pernyataan **Benar** atau **Salah**.

Pernyataan	Benar/Salah
Program di atas menampilkan hasil berupa angka berurutan dari 1 sampai 100.	...
Jika nilai a diubah menjadi 0, maka ada 100 angka yang ditampilkan.	...
Jika operator perbandingan diubah menjadi a >= 100, maka tidak ada hasilnya atau output kosong.	...
Batas atas dari output pada program di atas yaitu angka 100.	...

7. Selain menggunakan kondisi perulangan pada baris ke-3, alternatif lain untuk menampilkan 100 angka berurutan adalah ... (jawaban dapat lebih dari satu)

- (int a = 1; a < 101; a++)
- (int a = 0; a < 101; a++)
- (int a = 0; a < 100; a++)
- (int a = 1; a < 100; a++)

8. Ubahlah kode di atas agar menampilkan 100 bilangan
beurutan secara menurun.

.....
.....
.....
.....
.....
.....

Perhatikan kode program di bawah ini untuk soal nomor 9 – 10.

```
1 public class App {  
2     public static void main(String[] args) throws Exception {  
3         for (byte a = 1; a <= 100; a--) {  
4             System.out.print(a + " ");  
5         }  
6     }  
7 }
```

Kode di atas akan menampilkan bilangan berurutan secara menurun
dari 1 sampai tak terhingga.

9. Setujukah Anda dengan pernyataan berikut?

.....
.....
.....
.....
.....
.....

10. Berikan alasannya!

.....
.....
.....
.....
.....
.....

Perhatikan kode program di bawah ini untuk soal nomor 11 – 15.

```
1 public class App {  
2     public static void main(String[] args) throws Exception {  
3         for (int a = 10; a <= 35; a = a + 2) {  
4             System.out.print(a + " ");  
5         }  
6         System.out.println();  
7         for (int a = 32; a > 0; a = a - 2) {  
8             System.out.print(a + " ");  
9         }  
10    }  
11 }
```

11. Fungsi dari perintah $a = a + 2$ pada baris 3 yaitu ...

12. Jumlah bilangan yang ditampilkan pada kode baris 3 sampai 5 yaitu ...

- a. 17
- b. 12
- c. 13
- d. 11

13. Berdasarkan informasi berikut, berilah pernyataan **Benar** atau **Salah**.

Pernyataan	Benar/Salah
Program pada baris 3 sampai 5 menampilkan bilangan genap dari 10 sampai 34.	...
Program pada baris 7 sampai 9 menampilkan bilangan genap dari 32 sampai 0.	...
Fungsi dari perintah $a = a - 2$ pada baris 7 yaitu bilangan lompat dua secara mundur.	...
Jumlah bilangan yang ditampilkan pada kode baris 7 sampai 9 yaitu 16.	...

14. Ubahlah kode di atas menjadi bilangan lompat tiga secara menurun dari 50 sampai 25.

.....

.....

.....

.....

.....

.....

15. Pasangkan perintah kode tersebut sesuai dengan jenis perintahnya.

<code>int a = 32</code>	(1)
<code>a >= 0</code>	(2)
<code>a = a - 2</code>	(3)
<code>System.out.println(a + " ")</code>	(4)

(a)	Iterasi
(b)	Proses
(c)	Terminasi
(d)	Inisialisasi

Perhatikan kode program di bawah ini untuk soal nomor 16 – 20.

```
1 public class App {
2     public static void main(String[] args) throws Exception {
3         for (int a = 6; a <= 35; a++) {
4             if (a % 4 == 0) {
5                 System.out.print(a + " ");
6             }
7         }
8         System.out.println();
9         for (int a = 70; a >= 30; a--) {
10            if (a % 3 == 0) {
11                System.out.print(a + " ");
12            }
13        }
14    }
15 }
```

16. Pasangkan pernyataan berikut sesuai dengan jawaban yang tepat.

Batas atas untuk baris 3 sampai 7	(1)	(a)	69
Batas atas untuk baris 9 sampai 13	(2)	(b)	32
Batas bawah untuk baris 3 sampai 7	(3)	(c)	8
Batas bawah untuk baris 9 sampai 13	(4)	(d)	30

17. Syarat tambahan yang dipenuhi untuk menampilkan hasil dari kode program pada baris 9 sampai 13 yaitu ...
- a. bilangan kelipatan 3
 - b. bilangan kelipatan 4
 - c. bilangan lompat 3
 - d. bilangan lompat 4

18. Ubahlah kode di atas menjadi bilangan kelipatan 6 dari 70 sampai 100.

.....

.....

.....

.....

.....

.....

19. Hasil output untuk kode program baris 9 sampai 13 yaitu ...

20. Jumlah bilangan yang ditampilkan pada kode baris 3 sampai 7 yaitu ...
- a. 7
 - b. 8
 - c. 9
 - d. 6

Perhatikan kode program di bawah ini untuk soal nomor 21 – 24.

1	import java.util.*;
2	
3	public class App {
4	public static void main(String[] args) throws Exception {
5	Scanner sc = (1);

6	System.out.println("Tentukan nilai awal: ");
7	int bilawal = (2);
8	System.out.println("Tentukan nilai akhir: ");
9	int bilakhir = (3);
10	int total = 0;
11	
12	for (4) {
13	total = (5);
14	}
15	
16	System.out.println("Jumlah bilangan berurutan dari " + bilawal + " sampai " + bilakhir + " adalah " + total);
17	}
18	}

Kode di atas berisi tentang perulangan for yang menampilkan hasil jumlah bilangan berurutan dari bilangan awal sampai bilangan akhir yang ditentukan dari user.

21. Lengkapi kode program bernomor (1) sampai (3) di atas.

.....

.....

.....

.....

.....

.....

22. Perintah yang tepat untuk mengisi nomor (4) yaitu ... (*jawaban dapat lebih dari satu*).
- (int a = bilawal; a <= bilakhir; a++)
 - (int a = bilakhir; a >= bilawal; a--)
 - (int a = bilakhir; a > bilawal; a--)
 - (int a = bilawal; a < (bilakhir + 1); a++)

23. Perintah yang tepat untuk mengisi nomor (5) yaitu ...
- a. $0 + a$
 - b. `total++`
 - c. `total - a`
 - d. `total + a`
24. Jika nilai variabel `bilawal = 1` dan `bilakhir = 100`, maka variabel `total` akan bernilai ...

Perhatikan kode program di bawah ini untuk soal nomor 25 – 28.

```
1  import java.util.*;
2
3  public class App {
4      public static void main(String[] args) throws Exception {
5          Scanner sc = (1);
6          System.out.print("Tentukan banyak bilangan: ");
7          int banyakbil = (2);
8          double total = (3);
9
10         for (int a = 1; a <= banyakbil; a++) {
11             System.out.print("Masukkan bilangan ke " + a + ": ");
12             double bilangan = (4);
13             total = (5);
14         }
15
16         double ratarata = (6);
17         System.out.println("Nilai rata-ratanya adalah " + ratarata);
18     }
19 }
```

Kode di atas berisi tentang perulangan for yang menampilkan hasil rata-rata dari banyaknya bilangan yang dimasukkan oleh user.

25. Lengkapi kode program bernomor (1) sampai (4) di atas.

.....

.....

.....

.....

.....

.....

26. Perintah yang tepat untuk mengisi nomor (5) yaitu ...

- a. bilangan + a
- b. total + bilangan
- c. total – bilangan
- d. bilangan++

27. Perintah yang tepat untuk mengisi nomor (6) yaitu ...

28. Berdasarkan informasi berikut, berilah pernyataan **Benar** atau **Salah**.

Pernyataan	Benar/Salah
User memasukkan beberapa bilangan sesuai jumlah yang dikehendaki dengan variabel banyakbil.	...
Variabel total akan menghitung jumlah dari beberapa bilangan yang dimasukkan user.	...
Apabila bagian inisialisasi diganti menjadi a = 0, maka agar jumlah iterasinya sama,	...

bagian terminasi diganti dengan a < banyakbil.	
Jika user memasukkan variabel banyakbil = 3, dan ketiga bilangan itu 70, 74, dan 65, maka hasil rata-ratanya yaitu 70.	...

29. Menurut Anda, bagaimanakah perbedaan antara while dengan do-while?

.....

.....

.....

.....

.....

.....

30. Urutan komponen perulangan pada while yaitu ...

- a. inisialisasi – terminasi – iterasi – proses
- b. inisialisasi – terminasi – proses – iterasi
- c. inisialisasi – iterasi – proses – terminasi
- d. inisialisasi – proses – iterasi – terminasi

31. Urutan komponen perulangan pada do-while yaitu ...

- a. inisialisasi – terminasi – iterasi – proses
- b. inisialisasi – terminasi – proses – iterasi
- c. inisialisasi – iterasi – proses – terminasi
- d. inisialisasi – proses – iterasi – terminasi

32. Perhatikan kode program berikut:

1	public class App {
2	public static void main(String[] args) throws Exception {
3	for (byte a = 1; a <= 100; a--) {

```
4      System.out.print(a + " ");
5      }
6      }
7      }
```

Ubahlah kode program di atas dengan menggunakan while (NIM Ganjil) maupun do-while (NIM Genap).

.....

.....

.....

.....

.....

.....

Perhatikan kode program di bawah ini untuk soal nomor 33 – 36.

```
1  import java.util.*;
2
3  public class App {
4      public static void main(String[] args) throws Exception {
5          Scanner sc = new Scanner(System.in);
6          int total = (1);
7          int item = (2);
8          char jawaban;
9
10         do {
11             System.out.print("Masukkan harga: ");
12             int harga = (3);
13             total (4);
14             item (5);
15             System.out.print("Apakah ada tambahan barang? ");
16             jawaban = (6);
```

17	} while (7);
18	System.out.println("Total harga = " + total);
19	System.out.println("Total item = " + item);
20	}
21	}

Kode di atas berisi tentang perulangan perulangan do-while untuk menghitung berapa banyak item barang yang dibeli beserta total harganya. Perulangan terjadi apabila user memilih huruf "y" pada pertanyaan "Apakah ada tambahan barang?"

33. Lengkapi kode program bernomor (1) sampai (2) di atas.

.....

.....

.....

.....

.....

.....

34. Perintah yang tepat untuk mengisi nomor (5) yaitu ... (*jawaban dapat lebih dari satu*).

- item + 1 = item
- item++
- item += 1
- item = item + 1

35. Pasangkan nomor kode yang hilang berikut sesuai dengan isian yang tepat.

(3)	(1)
(4)	(2)

- (a) jawaban == 'y' || jawaban == 'Y'
- (b) Sc.nextLine().charAt(0)

(6)	(3)
(7)	(4)

(c)	sc.nextInt()
(d)	+= harga

36. Pada program do-while di atas, yang termasuk ke dalam bagian iterasi yaitu baris ...
37. Untuk membuat nested loop 5 dimensi, maka banyaknya perintah for yaitu sebanyak ... buah.
- 4
 - 5
 - 6
 - 10

Perhatikan kode program di bawah ini untuk soal nomor 38 – 40.

1	public class App {
2	public static void main(String[] args) throws Exception {
3	for (int a = 1; a <= 18; a++) {
4	for (int b = 1; b <= 10; b++) {
5	System.out.print("*");
6	}
7	System.out.println();
8	}
9	}
10	}

38. Program di atas yaitu membuat formasi bangun datar persegi panjang dengan tanda bintang (*). Yang menandakan jumlah kolom pada formasi tersebut terletak pada baris ..., sedangkan penanda jumlah baris pada formasi tersebut terletak pada baris....

39. Ubahlah kode di atas agar formasi bangun datar menjadi sebagai berikut.

```
*  
**  
***  
****  
*****
```

.....
.....
.....
.....
.....
.....

40. Apabila ingin membuat formasi persegi panjang sebanyak 300 tanda bintang, maka baris 3 dan 4 dapat diubah menjadi...
(jawaban dapat lebih dari satu).

- (int a = 1; a <= 30; a++) dan (int b = 1; int b <= 10; b++)
- (int a = 1; a <= 1; a++) dan (int b = 1; int b <= 300; b++)
- (int a = 1; a <= 15; a++) dan (int b = 1; int b <= 20; b++)
- (int a = 1; a <= 25; a++) dan (int b = 1; int b <= 10; b++)

SOAL PRAKTIKUM

Soal praktikum Perulangan dapat Anda scan pada QR Code di bawah ini:



PETUNJUK Pengerjaan:

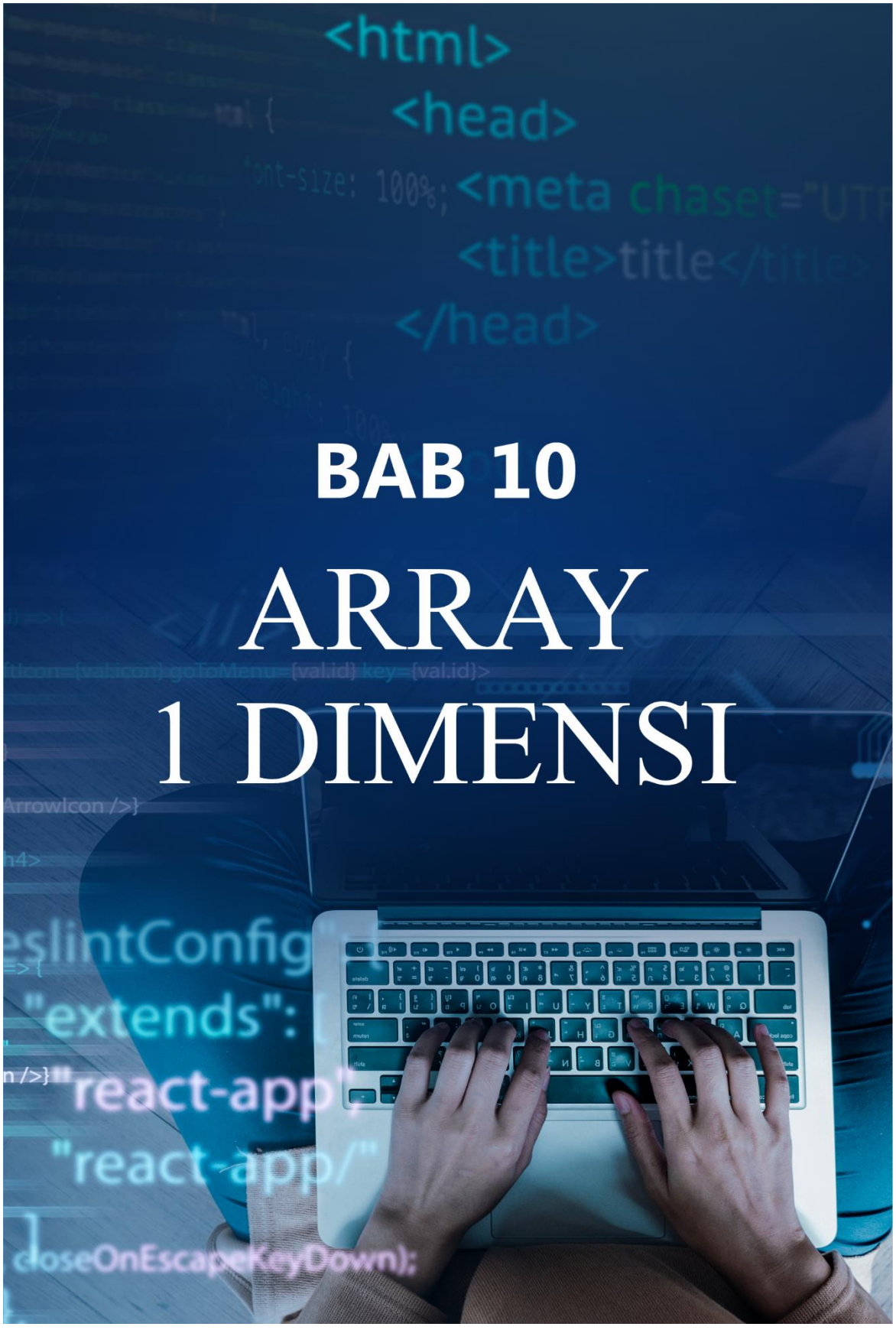
Gunakan 3 digit terakhir NIM Anda lalu hitunglah dengan modulo 12. Misal NIM Anda 322140203, maka 3 digit terakhir yaitu 203. Hasil dari $203 \% 12 = 11$. Tabel di bawah ini menunjukkan nomor soal / room yang harus Anda kerjakan sesuai hasil perhitungan.

Hasil NIM % 12	Nomor Soal / Room
1	1, 13, 25
2	2, 14, 26
3	3, 15, 27
4	4, 16, 28
5	5, 17, 29
6	6, 18, 30
7	7, 19, 31
8	8, 20, 32
9	9, 21, 33
10	10, 22, 34
11	11, 23, 35
0	12, 24, 36



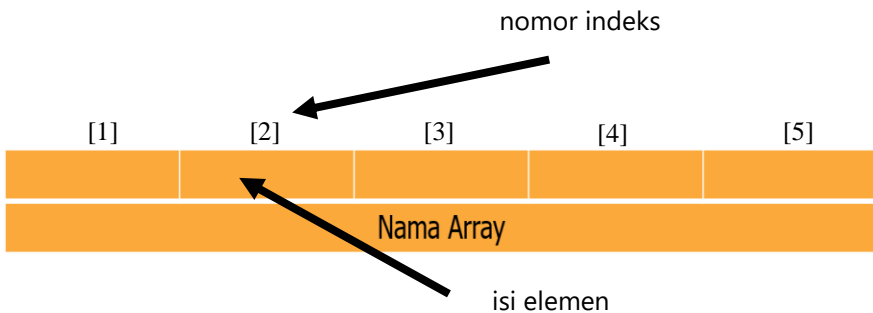
BAB 10

ARRAY 1 DIMENSI



10.1. Konsep Array

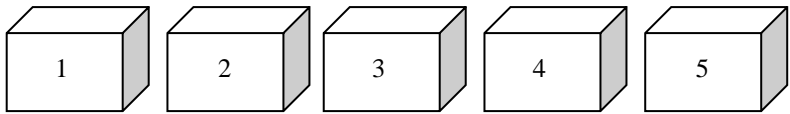
Secara garis besar, array dapat didefinisikan sebagai tabel di mana tabel merupakan sekumpulan elemen yang dapat diakses untuk setiap elemennya. Array menjadi hal yang penting dalam penyimpanan data, karena apabila array tidak ada, maka semisal memerlukan 5 tempat untuk 5 nilai, maka secara tidak langsung harus membuat 5 macam variabel yang berbeda. Hal tersebut dapat mempersulit pengaksesan data. Ilustrasi mengenai array ditunjukkan pada Gambar 10.1.



Gambar 10.1. Ilustrasi Array

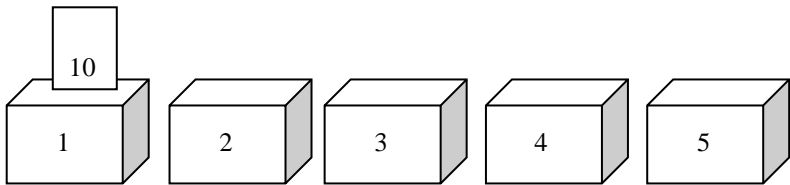
Dalam logika pemikiran manusia, array dianalogikan sebagai beberapa kotak penampungan (variabel) yang diikat menjadi satu sebagai kesatuan, dimana tiap kotak mempunyai nomor yang didefinisikan sebagai nomor indeks. Urutan analogi dari array yaitu sebagai berikut:

- Misalkan ada 5 buah kotak kosong dalam satu ikatan. Kelima kotak kosong itu berisikan nilai yang sesuai dengan tipe kotak itu sendiri (analogi sebagai tipe data).



Bahasa algoritmiknya yaitu: `ikatan_integer : array [1..5] of integer`

- Misalkan kotak bernomor 1 itu diisi dengan memasukkan nilai dari kotak tersebut:



Maka dapat disimpulkan bawa kotak ke-1 berisi angka 10.

Nomor kotak tersebut sering disebut dengan **indeks**.

Bahasa algoritmiknya: `ikatan_integer1 <- 10`

Secara umum, array dapat dibagi menjadi sebagai berikut:

- *Array* satu dimensi
- *Array* dua atau lebih dimensi (untuk hal ini disebut *array* multi dimensi)

10.2. Array Satu Dimensi

Sesuai dengan namanya, *array* satu dimensi yaitu *array* yang terdiri dari beberapa baris dan 1 kolom atau beberapa kolom dan 1 baris. Intinya dimensi dari *array* satu dimensi yaitu (1 x n) atau (n x 1). Misalkan ingin membuat *array* berupa tinggi badan 5 orang mahasiswa, maka berdasarkan subbab 10.1, maka langkah-langkah analoginya apabila ditulis dalam kode pemrograman *Java* yaitu sebagai berikut:

Langkah 1, deklarasi *array*:

Bahasa algoritmik: tinggi_badan : array [1..5] of **integer**

Bahasa pemrograman *Java*:

1	int[] tinggi_badan;
2	tinggi_badan = new int[5];

Atau dapat ditulis:

1	int tinggi_badan[];
2	tinggi_badan = new int[5];

Atau dapat ditulis:

1	int[] tinggi_badan = new int[5];
---	----------------------------------

Langkah 2, pengaksesan *array*:

Misalkan tinggi mahasiswa 1 170 cm, mahasiswa 2 160 cm, dan mahasiswa 3 177 cm, maka dapat ditulis sebagai berikut:

Bahasa algoritmik:

tinggi_badan1 <- 170

tinggi_badan2 <- 160

tinggi_badan3 <- 177

Bahasa pemrograman *Java*:

1	tinggi_badan[0] = 170;
2	tinggi_badan[1] = 160;
3	tinggi_badan[2] = 177;

Perlu diketahui, bahwa dalam bahasa pemrograman *Java*, indeks *array* dimulai dari 0 sampai dengan jumlah indeks – 1. Untuk menampilkan hasil pengaksesan *array* di output, maka penulisannya yaitu:

1	System.out.println(tinggi_badan[0]);
2	System.out.println(tinggi_badan[1]);
3	System.out.println(tinggi_badan[2]);

Maka hasil outputnya yaitu:

170
160
177

Namun, apabila pada bagian `System.out.println` diubah menjadi `(tinggi_badan[3])`, maka hasil outputnya yaitu **0**, dikarenakan indeks ke-3 tidak ada isinya. Untuk tipe data lain, hasil output untuk menampilkan array yang tidak ada isinya yaitu sebagai berikut:

- Tipe data numerik (int, byte, short, long): 0
- Tipe data riil (double, float): 0.0
- Tipe data char: (tidak ada isinya)
- Tipe data string: null

Jika pada bagian `System.out.println` diubah menjadi `(tinggi_badan[5])`, maka hasil outputnya yaitu galat **ArrayIndexOutOfBoundsException**. Arti dari galat tersebut yaitu program tidak dapat menampilkan indeks *array* yang melebihi daya tampung yang ditentukan sebelumnya. Pada bagian ini, daya tampung *array* yang ditentukan yaitu sebanyak 5 elemen, sementara `tinggi_badan[5]` itu merupakan indeks ke-6 (**INGAT**: Bahasa pemrograman *Java* penomoran indeks dimulai dari 0), sehingga melebihi daya tampung yang ditentukan dan terjadilah galat tersebut.

Array juga dapat dideklarasikan sekaligus diinisialisasi dengan kode penulisan sebagai berikut:

```
1 int tinggi_badan[] = {154, 160, 176, 163, 171, 170};
```

Dari kode diatas, maka dapat diketahui bahwa array `tinggi_badan` memiliki 6 buah indeks. Maka apabila ingin menampilkan tinggi badan pada indeks ke-3, maka outputnya yaitu 163, karena aturan indeks ke-0 tetap berlaku.

Sebagai praktikum, perhatikan tabel kode program berikut. Menurut Anda, bagaimana hasil outputnya?

Tabel 10.1. Kode Program Array

```
1 public class App {
2     public static void main(String[] args) throws Exception {
3         int [] no_undian = new int [10];
4         no_undian[3] = 298;
5         System.out.println(no_undian[3]);
6     }
7 }
```

Tabel 10.2. Kode Program Array

```
1 public class App {
2     public static void main(String[] args) throws Exception {
3         int [] no_undian = new int [10];
4         no_undian[1] = 298;
5         no_undian[2] = 106;
6         no_undian[3] = 305;
7
8         for (int a = 0; a < no_undian.length; a++) {
9             System.out.println(no_undian[a]);
10        }
11    }
12 }
```

Tabel 10.3. Kode Program Array

```
1 public class App {
2     public static void main(String[] args) throws Exception
3     {
4         int [] no_undian = {59, 102, 115, 134, 176, 146};
5
6         for (int a = 0; a < no_undian.length; a++) {
7             System.out.println(no_undian[a]);
8         }
9     }
}
```

Karena materi ini terbagi menjadi dua bagian, maka untuk asesmen dan tugas praktikum akan diberikan pada bab selanjutnya.

```
<html>
```

```
<head>
```

```
font-size: 100%; <meta charset="UTF
```

```
<title>title</title>
```

```
</head>
```

BAB 11

ARRAY

MULTI DIMENSI

```
rowIcon />
```

```
slintConfig
```

```
"extends": [
```

```
>] "react-app",
```

```
"react-app/"
```

```
closeOnEscapeKeyDown);
```



11.1 Array 2 Dimensi

Array dua dimensi yaitu *array* yang memiliki dua atau lebih kolom dengan beberapa baris, atau dua atau lebih baris dengan beberapa kolom. Ilustrasinya sebagai sebuah tabel yang berukuran $m \times n$ dimana m = jumlah baris dan n = jumlah kolom. Gambar 11.1 dibawah ini merupakan ilustrasi dari *array* dua dimensi.

indeks 1,1	←	3	7	2	...	10	→	indeks n,1
		11	8	5	...	2		
		13	5	4	...	17		
		⋮	⋮	⋮	⋮	⋮		
indeks 1,n	←	20	10	2	...	13	→	indeks n,n

Gambar 11.1. Ilustrasi *Array* Dua Dimensi

Proses penelusuran pada matriks *array* atau yang disebut sebagai scanning dilakukan dari tiap baris dan tiap baris ditelusuri tiap kolomnya. Gambar 11.2 merupakan matriks *array* ukuran 3×4 .

3	7	2	13
11	8	5	14
13	5	4	10

Gambar 11.2 Matriks ukuran 3×4

Berdasarkan Gambar 11.2, maka penelusuran dari matriks tersebut akan mendapatkan urutan jalan $3 - 7 - 2 - 13 - 11 - 8 - 5 - 14 - 13 - 5 - 4 - 10$, untuk setiap baris ditelusuri tiap kolomnya. Berikut merupakan algoritma untuk membuat matriks ukuran 3×4

(12 elemen), beserta mengisi *array* dan menampilkan *array* dengan bahasa algoritmik beserta kode program *Javanya*.

Langkah 1, deklarasi *array*:

Bahasa algoritmik:

matriks : array [1..3, 1..4] of **integer**

Bahasa pemrograman *Java*:

1	import java.util.Scanner;
2	
3	public class App{
4	public static void main(String[] args) throws Exception {
5	
6	int matriks [][] = new int[3][4];

Langkah 2, membuat variabel baris dan kolom:

Bahasa algoritmik:

baris : **integer**

kolom : **integer**

Bahasa pemrograman *Java*:

1	int baris;
2	int kolom;

Langkah 3, membuat perulangan untuk mengisi masing-masing baris dan kolom:

Bahasa algoritmik:

for baris <- 1 to 3 do

for kolom <- 1 to 4 do

input (matriks_{baris, kolom})

 (end for)

(end for)

Bahasa pemrograman *Java*:

1	Scanner sc = new Scanner (System.in);
2	
3	for (baris = 0, baris = matriks.length; baris++ {
4	for (kolom = 0, kolom = matriks[baris].length; kolom++ {
5	System.out.print("masukkan angka:");
6	matriks[baris][kolom] = sc.nextInt();
7	}
8	}

Langkah 4, membuat perulangan untuk menampilkan *array*:

Bahasa algoritmik:

for baris <- 1 to 3 do

for kolom <- 1 to 4 do

output (matriks_{baris, kolom})

 (end for)

output ("\n")

(end for)

Bahasa pemrograman *Java*:

1	for (baris = 0, baris = matriks.length; baris++ {
2	for (kolom = 0, kolom = matriks[baris].length; kolom++ {

```
3   System.out.print(matriks[baris][kolom] + " ");
4   }
5   System.out.println("");
6   }
7   }
8   }
```

Maka output yang dihasilkan, apabila pengisian array sesuai dengan matriks pada Gambar 2.3 yaitu:

```
3 7 2 13
11 8 5 14
13 5 4 10
```

Kode program di atas tidak hanya untuk *array* dua dimensi, bisa juga untuk *array* tiga dimensi, empat dimensi, dan seterusnya, sehingga disebut sebagai *array* multi dimensi. Apabila ingin membuat *array* tiga dimensi, maka deklarasinya menjadi sebagai berikut:

```
1   import java.util.Scanner;
2
3   public class App{
4   public static void main(String[] args) throws Exception {
5
6   int matriks [][][] = new int[3][4][5];
```

Contoh di atas apabila ingin membuat matriks 3 x 4 sebanyak 5 lapisan (*layer*). Maka jumlah elemen yang dapat diisi yaitu 3 x 4 x 5 = 60 elemen.

Sebagai praktikum, perhatikan tabel kode program berikut. Menurut Anda, bagaimana hasil outputnya?

Tabel 11.1. Kode Program Array 2 Dimensi

```

1 public class App {
2     public static void main(String[] args) throws Exception {
3         String [][] buah = {"Mangga", "Jambu"}, {"Apel",
4         "Rasberi"}, {"Jeruk", "Pepaya"};
5         System.out.println(buah[0][1]);
6         System.out.println(buah[1][3]);
7         System.out.println(buah[2][1]);
8     }
9 }

```

Tabel 11.2. Kode Program Array 3 Dimensi

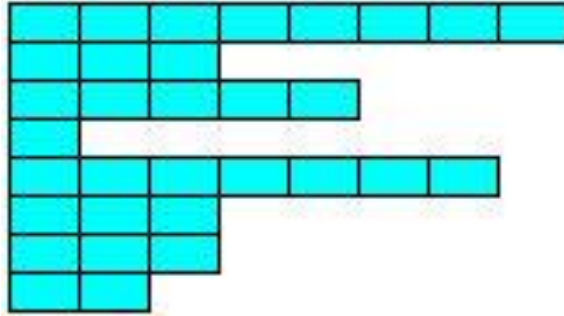
```

1 public class App {
2     public static void main(String[] args) throws Exception {
3         int [][][] data = new int [2][3][3];
4         int angka = 0;
5
6         for (int a = 0; a < data.length; a++) {
7             for (int b = 0; b < data[a].length; b++) {
8                 for (int c = 0; c < data[a][b].length; c++) {
9                     angka++;
10                    data[a][b][c] = angka;
11                    System.out.print(data[a][b][c] + "\t");
12                }
13                System.out.println();
14            }
15            System.out.println();
16        }
17    }
18 }

```

11.2 Jagged Array

Jagged array yaitu *array* dua dimensi yang memiliki jumlah kolom yang tidak sama untuk tiap barisnya ataupun jumlah baris yang tidak sama untuk tiap kolomnya. Ilustrasi *jagged array* ditunjukkan pada Gambar 11.3.



Gambar 11.3. *Jagged Array*

Jika dilihat dari Gambar 11.3, maka baris ke-1 memiliki 8 kolom, baris ke-2 memiliki 3 kolom, baris ke-3 memiliki 5 kolom, demikian seterusnya dimana jumlah kolom berbeda-beda dari tiap barisnya. Untuk membuat *jagged array*, sesuai Gambar 11.3, berikut langkah-langkahnya.

Langkah 1, membuat *array* dan jumlah barisnya:

Bahasa pemrograman *Java*:

```
1 | int jagged [][] = new int[8][];
```

Langkah 2, menentukan jumlah kolom untuk tiap barisnya:

Bahasa pemrograman *Java*:

```
1 | jagged[0] = new int[8];  
2 | jagged[1] = new int[3];  
3 | jagged[2] = new int[5];  
4 | jagged[3] = new int[1];
```

5	jagged[4] = new int[7];
6	jagged[5] = new int[3];
7	jagged[6] = new int[3];
8	jagged[7] = new int[2];

Sebagai praktikum, perhatikan tabel kode program berikut. Menurut Anda, bagaimana hasil outputnya?

Tabel 11.3. Kode Program Jagged Array

1	public class App {
2	public static void main(String[] args) throws Exception {
3	int [][] angka = new int [3][];
4	angka[0] = new int [2];
5	angka[1] = new int [5];
6	angka[2] = new int [4];
7	
8	for (int baris = 0; baris < angka.length; baris++) {
9	int nilaiBaris = baris;
10	for (int kolom = 0; kolom < angka[baris].length;
11	kolom++) {
12	angka[baris][kolom] = nilaiBaris;
13	}
14	}
15	
16	for (int baris = 0; baris < angka.length; baris++) {
17	for (int kolom = 0; kolom < angka[baris].length;
18	kolom++) {
19	System.out.print(angka[baris][kolom]);
20	}
21	System.out.print("\n");
22	}

	}
	}

ASESMEN

Jawablah pertanyaan di bawah ini dengan benar!

1. Penomoran pada array disebut juga ...
 - a. indeks
 - b. elemen
 - c. variabel
 - d. cell

2. Berdasarkan informasi berikut, berilah pernyataan **Benar** atau **Salah**.

Pernyataan	Benar/Salah
Array merupakan solusi untuk menyimpan sejumlah data dengan tipe yang sama tanpa harus membuat variabel yang sangat banyak sehingga lebih praktis.	...
Tiap petak tabel pada array disebut elemen.	...
Array satu dimensi hanya terdiri atas satu kolom dan beberapa baris.	...
Penomoran indeks array pada pemrograman Java selalu diawali dari 1.	...

3. Ukuran matriks untuk array satu dimensi yaitu ...

4. Penulisan kode program untuk membuat array bertipe integer sebanyak 10 elemen yaitu ... (*jawaban dapat lebih dari satu*).

- `int [] array;`
`array = new int [10];`
- `int [] array = new int [10];`
- `int array [];`
`array = 10;`
- `int array [] = new int [10];`

Perhatikan kode program di bawah ini untuk soal nomor 5 – 9.

```
1 public class App {
2     public static void main(String[] args) throws Exception {
3         int [] jarak = new int [9];
4
5         jarak[0] = 1500;
6         jarak[1] = 3000;
7         jarak[4] = 2900;
8         jarak[6] = 2400;
9         jarak[7] = 3200;
10
11         System.out.println(jarak[1]);
12     }
13 }
```

5. Output yang ditampilkan pada baris 11 yaitu ...

- a. 1500
- b. 3000
- c. 2900
- d. 2400

6. Pasangkan pernyataan berikut sesuai outputnya.

indeks ke-1	(1)	(a)	1500
elemen ke-5	(2)	(b)	3000
indeks ke-3	(3)	(c)	0
elemen ke-1	(4)	(d)	2900

7. Apabila pada baris 11 diubah menjadi "System.out.println (jarak[7]);", maka pasanglah hasil yang sesuai dengan tipe datanya.

double	(1)	(a)	null
char	(2)	(b)	0.0
short	(3)	(c)	(output kosong)
string	(4)	(d)	0

8. Apakah output yang muncul jika pada baris 11 diubah menjadi "System.out.println (jarak[9]);"?

.....

.....

.....

.....

.....

.....

9. Berikan alasannya!

.....

.....

.....

.....

.....

Perhatikan kode program di bawah ini untuk soal nomor 10 – 12.

```

1 public class App {
2     public static void main(String[] args) throws Exception {
3         String [] nama = {"Amir", "Budi", "Tiara", "Dedi", "Eca",
4 "Karina", "Listi", "Ratih", "Tono"};
5
6         System.out.println(nama[4]);
7     }
8 }

```

10. Berdasarkan informasi berikut, berilah pernyataan **Benar** atau **Salah**.

Pernyataan	Benar/Salah
Array "nama" bertipe String.	...
Array "nama" berjumlah 9 elemen.	...
Hasil output pada baris 6 yaitu Dedi	...
Elemen ke-2 dari array tersebut yaitu Budi.	...

11. Apabila pada baris 11 diubah menjadi "System.out.println (nama[6]);", maka output yang muncul yaitu ...

12. Hasil output akan menjadi "ArrayIndexOutOfBoundsException", jika baris 11 diubah menjadi ... (*jawaban dapat lebih dari satu*).

- System.out.println (nama[9]);
- System.out.println (nama[0]);
- System.out.println (nama[10]);
- System.out.println (nama[-1]);

Perhatikan kode program di bawah ini untuk soal nomor 13 – 16.

```
1 import java.util.*;
2
3 public class App {
4     public static void main(String[] args) throws Exception {
5         Scanner sc = (1);
6         System.out.println("Masukkan banyak angka: ");
7         int banyak_angka = (2);
8         int[] angka = new (3)[(4)];
9         int max = 0;
10
11         for (int a = 0; a < (5); a++) {
12             System.out.print("Masukkan angka ke-" + angka [a]
13 + ": ");
14             angka[a] = (6);
15         }
16
17         for (int a = 0; a < (7); a++) {
18             if (angka[a] > (8)) {
19                 max = (9);
20             }
21         }
22         System.out.println();
23         System.out.println("Nilai tertinggi adalah: " + (10);
24     }
}
```

13. Lengkapi kode program bernomor (1) sampai (7) di atas.

.....
.....
.....
.....
.....

14. Apakah perbedaan dari perulangan for pada baris 11-14 dengan baris 16-19?

.....
.....
.....
.....
.....

15. Pernyataan yang tepat untuk mengisi nomor (8) di atas yaitu ...

- a. 0
- b. angka[a-1]
- c. max
- d. angka[a]

16. Pernyataan yang tepat untuk mengisi nomor (10) di atas yaitu...

Perhatikan kode program di bawah ini untuk soal nomor 17 – 20.

1	import java.util.*;
2	
3	public class App {
4	public static void main(String[] args) throws Exception {
5	Scanner sc = (1);
6	System.out.print("Masukkan jumlah kata: ");
7	int jumlah_kata = (2);

```

8      String kata [] = new (3)[(4)];
9      int count = (5);
10
11     for (int a = 0; a < (6); a++) {
12         System.out.print("Masukkan kata: ");
13         kata[a]= (7);
14     }
15
16     for (int a = 0; a < (8); a++) {
17         if (9){
18             System.out.println("Kata yang mengandung huruf
19             A adalah " + (10));
20         }
21     }
22 }

```

17. Lengkapi kode program bernomor (1) sampai (6) di atas.

.....

.....

.....

.....

.....

18. Pernyataan yang tepat untuk mengisi nomor (10) di atas yaitu...

19. Pernyataan yang tepat untuk mengisi nomor (7) di atas yaitu ...

- a. sc.next();
- b. sc.nextLine();
- c. sc.nextString();
- d. sc.nextChar();

20. Bagaimanakah penulisan kondisi if yang tepat untuk mengisi nomor (9)?

.....

21. Berdasarkan informasi berikut, berilah pernyataan **Benar** atau **Salah**.

Pernyataan	Benar/Salah
Array dua dimensi yaitu array yang berisi beberapa baris dan beberapa kolom.	...
Ukuran matriks untuk array dua dimensi yaitu (m x n).	...
Jumlah elemen untuk array dua dimensi ditentukan dengan hasil perkalian dari elemen baris (m) dan elemen kolom (n).	...
Jika jumlah elemen baris (m) sebanyak 6 elemen dan elemen kolom (n) sebanyak 1 elemen, maka array tersebut disebut array dua dimensi.	...

22. Diketahui kode program:

```
int array [][] = new int [m][n]
```

Apabila array tersebut berisikan 40 elemen, maka nilai m dan n yang memungkinkan yaitu ... (jawaban dapat lebih dari satu).

- m = 10, n = 4
- m = 1, n = 40
- m = 2, n = 25
- m = 5, n = 8

23. Array yang memiliki jumlah kolom yang berbeda di tiap barisnya yaitu ...
- jagged array
 - array satu dimensi
 - array dua dimensi
 - array multi dimensi

Perhatikan kode program di bawah ini untuk soal nomor 24 – 26.

```
1 import java.util.*;
2
3 public class App {
4     public static void main(String[] args) throws Exception {
5         int [][] angka = new int [3][];
6         angka[0] = new int [4];
7         angka[1] = new int [3];
8         angka[2] = new int [2];
9
10        for (int baris = 0; baris < angka.length; baris++) {
11            int nilaiBaris = baris+3;
12            for (int kolom = 0; kolom < angka[baris].length;
13 kolom++) {
14                angka[baris][kolom] = nilaiBaris;
15            }
16        }
17
18        for (int baris = 0; baris < angka.length; baris++) {
19            for (int kolom = 0; kolom < angka[baris].length;
20 kolom++) {
21                System.out.print(angka[baris][kolom]);
22            }
23            System.out.print("\n");
```

24	}
	}
	}

24. Jumlah elemen yang dibentuk dari array pada kode di atas yaitu...

25. Hasil output untuk baris ke-1 yaitu ...
- a. 3 3 3 3
 - b. 3 3 3
 - c. 4 4 4
 - d. 0 0 0 0

26. Ubahlah kode di atas sehingga menghasilkan output sebagai berikut:

1 1 1
3
5 5 5 5
7
9 9

.....
.....
.....
.....
.....

Perhatikan pernyataan berikut ini:

Apabila diberikan kode sebagai berikut:

```
String [][][][] array = new String [1][1][1][1][1][1];
```

Maka array tersebut merupakan array 6 dimensi.

27. Setujukah Anda dengan pernyataan berikut?

.....
.....
.....
.....
.....

28. Berikan alasannya!

.....
.....
.....
.....
.....

29. Berdasarkan informasi berikut, berilah pernyataan **Benar** atau **Salah**.

Pernyataan	Benar/Salah
Array multi dimensi merupakan array yang memiliki dimensi dua atau lebih.	...
Jumlah dimensi pada array bergantung pada banyaknya karakter kurung kotak “[]” pada deklarasi array.	...
Kode String [][][][] array = new String [1][1][1][1][1][1] berisikan 6 elemen array.	...

Banyaknya karakter kurung kotak “[]” pada deklarasi array harus sama dengan saat membuat objek array.

...

Perhatikan kode program di bawah ini untuk soal nomor 30 – 32.

```
1 public class App {  
2     public static void main(String[] args) throws Exception {  
3         double matriks [][][][] = new Double [3][6][5][1][2];  
4     }  
5 }
```

30. Banyaknya elemen yang dibentuk dari array multi dimensi di atas yaitu ...

- a. 180
- b. 90
- c. 17
- d. 36512

31. Pasangkan objek array berikut sesuai dengan jumlah elemen yang dibentuk.

[2][1][2][1][2]	(1)	(a)	120
[10][10][10]	(2)	(b)	1000
[4][6][5][2]	(3)	(c)	8
[5][4][3][2][1]	(4)	(d)	240

32. Array pada program di atas merupakan array ... dimensi.

Perhatikan kode program di bawah ini untuk soal nomor 33 – 36.

```

1  import java.util.*;
2
3  public class App {
4      public static void main(String[] args) throws Exception {
5          String [][] sayur = {"Timun", "Brokoli"}, {"Selada",
6              "Terong"}, {"Wortel", "Sawi"};
7          System.out.println(sayur[0][1]);
8          System.out.println(sayur[1][3]);
9          System.out.println(sayur[2][1]);
10     }
11 }

```

33. Berdasarkan informasi berikut, berilah pernyataan **Benar** atau **Salah**.

Pernyataan	Benar/Salah
Hasil output untuk baris 8 yaitu sawi.	...
Ukuran matriks pada array di atas yaitu (2 x 3).	...
Jumlah elemen pada array di atas yaitu sebanyak 6 elemen.	...
Selada berada pada array sayur[1][0].	...

34. Hasil output untuk baris 7 yaitu ...

35. Ubahlah kode di atas sehingga menghasilkan output sebagai berikut:

```

Timun           Brokoli
Selada          Terong
Wortel          Sawi

```

.....

.....
.....
.....

36. Pasangkan output berikut sesuai dengan objek array yang ditentukan.

Timun	(1)	(a)	sayur[2][0]
Brokoli	(2)	(b)	sayur[0][0]
Wortel	(3)	(c)	sayur[1][1]
Terong	(4)	(d)	sayur[0][1]

Perhatikan kode program di bawah ini untuk soal nomor 37 – 40.

```
1 import java.util.*;
2 public class App {
3     public static void main(String[] args) throws Exception {
4         Scanner sc = (1);
5         String [] namaBarang;
6         int n;
7         System.out.print("Masukkan Jumlah Item: ");
8         n = (2);
9         namaBarang = new (3);
10        String status = "";
11
12        int [][] data = new int [(4)][2];
13        for (int a = 0; a < (5); a++) {
14            System.out.print("Barang ke-" + (a+1) + ": ");
15            namaBarang[a] = (6);
16            System.out.print("Harga satuan: ");
17            data[a][0] = (7);
18            System.out.print("Stok: ");
19            data[a][1] = (8);
```

```

20     }
21
22     System.out.println();
23     System.out.println("-----
-----");
24     System.out.println("No\tNama Barang\tHarga
Satuan\tStok\t\tStatus");
25     System.out.println("-----
-----");
26     for (int a = 0; a < (9); a++) {
27         System.out.print((a+1) + "\t" + namaBarang[(10)] +
28 "\t\t");
29         for (int b = 0; b < 2; b++) {
30             System.out.print(data(11) + "\t\t");
31         }
32         if (12) {
33             status = "Tambah stok";
34         } else {
35             status = "Tidak tambah stok";
36         }
37         System.out.println(13);
38     }
39     System.out.println("-----
-----");
40 }

```

Kode di atas untuk membuat list barang di suatu supermarket dan melakukan pengecekan stok barang apabila sisa stok barang kurang dari 100 pcs, maka perlu dilakukan tambahan stok barang.

37. Lengkapi kode program bernomor (1) sampai (8) di atas.

.....
.....
.....
.....
.....

38. Kode perulangan pada kode di atas yang berfungsi sebagai tempat pengisian elemen array multi dimensi yaitu ...

- a. baris 4 – 10
- b. baris 12 – 20
- c. baris 22 – 25
- d. baris 26 – 37

39. Pernyataan yang tepat untuk mengisi nomor (12) yaitu ...

40. Pasangkan nomor kode yang hilang berikut sesuai dengan isiannya yang tepat.

(9)	(1)
(10)	(2)
(11)	(3)
(13)	(4)

(a)	[a][b]
(b)	status
(c)	namaBarang.length
(d)	[a]

SOAL PRAKTIKUM

Soal praktikum Array dapat Anda scan pada QR Code di bawah ini:



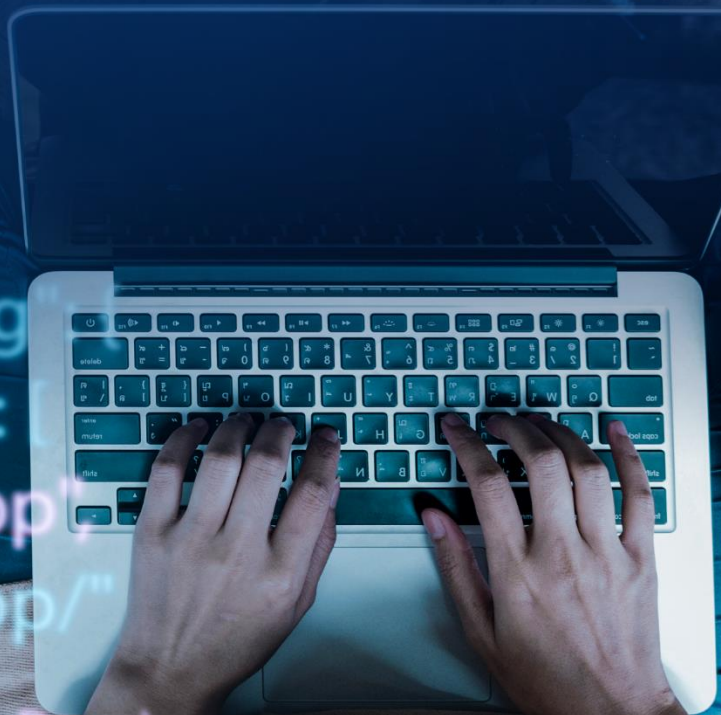
PETUNJUK Pengerjaan:

Gunakan 3 digit terakhir NIM Anda lalu hitunglah dengan modulo 7. Misal NIM Anda 322140203, maka 3 digit terakhir yaitu 203. Hasil dari $203 \% 7 = 0$. Tabel di bawah ini menunjukkan nomor soal / room yang harus Anda kerjakan sesuai hasil perhitungan.

Hasil NIM % 7	Nomor Soal / Room
1	37, 44
2	38, 45
3	39, 46
4	40, 47
5	41, 48
6	42, 49
0	43, 50

BAB 12

PROYEK AKHIR



Penilaian akhir pada matakuliah ini dilakukan setelah menempuh serangkaian materi Algoritma Pemrograman pada buku ini. Penilaian dilakukan dengan membuat sebuah proyek berupa sebuah program yang mengandung beberapa materi dari Algoritma Pemrograman, yaitu percabangan, looping, dan array.

Petunjuk pelaksanaan proyek akhir yaitu sebagai berikut:

1. Bagilah kelas Anda menjadi kelompok beranggotakan 2-3 orang.
2. Silakan membuat sebuah program mengenai informasi apapun (data bisa diperoleh dari narasumber ataupun secara daring). Untuk mempermudah, buatlah gambaran berupa tabel terlebih dahulu, kemudian identifikasi tipe data yang cocok dari masing-masing atribut / variabelnya.
3. Nilai tambahan apabila program yang dibuat selain dari beberapa studi kasus di buku ini (pada bagian asesmen dan soal praktikum).

Perencanaan jadwal proyek akhir:

1. Pekan ke-12 perkuliahan: Menentukan tim proyek beserta judul proyek yang diusulkan.
2. Pekan ke-13 perkuliahan: Mempresentasikan usulan judul proyek, dengan format: Judul proyek, deskripsi singkat proyek, dan jenis struktur data apa yang digunakan.
3. Pekan ke-14 perkuliahan: Pembimbingan proyek kepada dosen pengampu matakuliah (opsional).
4. Pekan ke-15 perkuliahan: Mempresentasikan kemajuan proyek. Usahakan progres minimum 60%.
5. Pekan ke-16 (UAS): Mempresentasikan hasil akhir proyek.

Jadwal tersebut dilakukan pada saat kegiatan tatap muka. Sementara kegiatan praktikum ditiadakan agar mahasiswa lebih memfokuskan penyelesaian proyek akhir tersebut.

Selain mempresentasikan hasil akhir proyek, Anda diminta untuk mengumpulkan laporan proyek sebagai tanda bukti (hard copy jilid biasa), beserta lembar progres mingguan yang dicantumkan pada halaman terakhir laporan proyek. Berikut ini format laporan dan lembar progres mingguan.

LAPORAN PROYEK AKHIR ALGORITMA PEMROGRAMAN

<Judul Proyek>

<Logo Institusi>

diajukan untuk penilaian tugas akhir
matakuliah Algoritma Struktur Data

NAMA ANGGOTA:

<NAMA dan NIM anggota kelompok>

Dosen Pengampu:

<Nama Dosen Pengampu Matakuliah beserta gelar>

<NAMA FAKULTAS/PRODI>

<NAMA INSTITUSI>

TAHUN ...

Format isi Laporan

BAB 1: Pendahuluan

Bab ini berisi mengenai deskripsi dari proyek yang Anda buat, fitur apa saja yang ada dalam proyek tersebut, beserta batasan dari proyek tersebut.

BAB 2: Hasil

Bab ini berisi hasil akhir dari proyek yang Anda buat, yaitu berupa kode program dan output programnya.

BAB 3: Penutup

Bab ini berisi kesimpulan dari proyek yang Anda buat, beserta rencana keberlanjutan dari proyek tersebut (jika ada).

Format lembar progres proyek

LEMBAR PROGRES PROYEK

Nama Kelompok: 1.(NIM.....)

2.(NIM.....)

3.(NIM.....)

Kelas :

Judul Proyek :

.....

Dosen Pengampu:

No	Tanggal	Kegiatan	TTD Dosen Pengampu
1			
2			
3			
4			
5			

Keterangan: Tabel bisa ditambahkan sesuai dengan pengerjaan proyek masing-masing tim.

REFERENSI

- Rosa, A. S. (2018a). *Logika Algoritma dan Pemrograman Dasar*. Modula.
- Rosa, A. S. (2018b). *Struktur Data: Terapan Dalam Berbagai Bahasa Pemrograman: Pascal, C, C++, Dan Java*. Modula.

