

## Modu1 Pendahuluan

### Tugas Pendahuluan

1. Apa yang anda ketahui tentang Java..?
2. Apa perbedaan Java dengan Bahasa pemograman lain..?
3. Apa yang dimaksud dengan Bahasa berorientasi object..?

### Materi

Java adalah bahasa pemrograman tingkat tinggi yang dirancang untuk menjadi platform-independent dan memungkinkan pengembangan perangkat lunak yang dapat dijalankan di berbagai sistem operasi dan perangkat. Java pertama kali dikembangkan oleh James Gosling dan timnya di Sun Microsystems pada tahun 1990-an.



Java menggunakan sintaks yang mirip dengan bahasa C dan C++, membuatnya relatif mudah dipelajari bagi programmer yang sudah memiliki pengalaman dengan bahasa-bahasa tersebut. Namun, Java memiliki beberapa perbedaan penting, terutama dalam pendekatan pemrograman berorientasi objek dan manajemen memori.

Salah satu fitur paling menonjol dari Java adalah kemampuannya untuk menjalankan program pada lingkungan yang berbeda secara platform-independent. Hal ini dimungkinkan melalui penggunaan Java Virtual Machine (JVM), yang menerjemahkan kode Java menjadi bytecode yang dapat dijalankan oleh JVM di sistem target. Dengan cara ini, program Java dapat dijalankan di berbagai sistem operasi seperti Windows, macOS, Linux, dan lainnya tanpa perlu mengubah kode sumbernya.

Java juga memiliki dukungan yang kuat untuk pemrograman berorientasi objek. Setiap program Java terdiri dari satu atau lebih class, yang menggambarkan struktur dan perilaku objek. Java menyediakan fitur-fitur seperti pewarisan, enkapsulasi, dan polimorfisme yang memungkinkan pengembangan perangkat lunak yang modular, mudah dipelihara, dan dapat diubah dengan cepat.

Selain itu, Java memiliki berbagai library dan framework yang kaya, yang menyediakan fungsionalitas tambahan untuk pengembangan perangkat lunak. Hal ini memudahkan pengembang untuk membangun aplikasi yang kompleks dan berperforma tinggi dengan memanfaatkan kelas-kelas dan fungsi-fungsi yang sudah ada.

Java digunakan secara luas dalam berbagai bidang, termasuk pengembangan aplikasi desktop, pengembangan web, pengembangan permainan, pengembangan aplikasi mobile, dan banyak lagi. Java memiliki beberapa kelebihan dan kekurangan yang perlu dipertimbangkan sebelum memilih bahasa pemrograman untuk pengembangan perangkat lunak. Berikut adalah beberapa kelebihan dan kekurangan Java:

#### Kelebihan Java:

1. Platform-Independent: Program Java dapat dijalankan di berbagai sistem operasi, seperti Windows, macOS, Linux, dan lainnya, karena Java menggunakan konsep "Write Once, Run Anywhere" melalui Java Virtual Machine (JVM).
2. Pemrograman Berorientasi Objek: Java dirancang dengan pendekatan pemrograman berorientasi objek yang kuat, yang memungkinkan pengembangan perangkat lunak yang modular, mudah dipelihara, dan dapat diubah dengan cepat.
3. Keamanan: Java memiliki fitur keamanan yang kuat, seperti mekanisme sandbox yang membatasi akses program terhadap sistem operasi dan lingkungan yang dijalkannya. Hal ini membantu melindungi sistem dari ancaman keamanan yang mungkin disebabkan oleh program yang berjalan.
4. Pengelolaan Memori Otomatis: Java menggunakan mekanisme pengumpulan sampah (garbage collection) yang otomatis mengelola pembebasan memori yang tidak lagi digunakan. Hal ini membantu mengurangi kesalahan pemrograman yang berhubungan dengan manajemen memori.
5. Library dan Framework Yang Kaya: Java memiliki ekosistem yang luas dengan berbagai library dan framework yang dapat digunakan untuk membangun berbagai jenis aplikasi, seperti Spring, Hibernate, JavaFX, dan banyak lagi. Hal ini memudahkan pengembang untuk membangun aplikasi yang kompleks dengan cepat.

#### Kekurangan Java:

1. Performa: Meskipun Java telah meningkatkan kinerjanya dari waktu ke waktu, Java mungkin tidak secepat bahasa pemrograman lain yang dikompilasi ke kode mesin langsung, seperti C++. Namun, dalam sebagian besar kasus, perbedaan performa ini tidak signifikan dan bisa dikompensasi dengan kemajuan teknologi hardware.
2. Kompleksitas: Java adalah bahasa pemrograman yang cukup kompleks, terutama bagi pemula. Beberapa konsep seperti pengelolaan memori, thread, dan concurrency dapat memerlukan pemahaman yang lebih mendalam untuk menghindari kesalahan pemrograman.
3. Konsumsi Memori yang Lebih Besar: Java menggunakan JVM dan menyediakan tingkat abstraksi yang lebih tinggi, yang dapat mengakibatkan konsumsi memori yang lebih besar dibandingkan dengan bahasa pemrograman yang lebih dekat ke level sistem.
4. Waktu Pembelajaran: Karena kompleksitas dan fitur yang banyak, waktu yang dibutuhkan untuk mempelajari Java mungkin lebih lama dibandingkan dengan bahasa pemrograman lain yang lebih sederhana.
5. Awal Pemuatan yang Lambat: Karena penggunaan JVM, aplikasi Java mungkin membutuhkan waktu pemuatan awal yang lebih lama dibandingkan dengan aplikasi yang ditulis dalam bahasa pemrograman lain. Namun, sekali aplikasi dimuat, biasanya kinerja aplikasi Java stabil.

Penting untuk mempertimbangkan kelebihan dan kekurangan ini berdasarkan kebutuhan proyek Anda dan tingkat pengalaman Anda sebagai pengembang.

## 1. Kenapa Harus Belajar Java

Belajar Java memiliki beberapa alasan yang kuat. Berikut adalah beberapa alasan mengapa Anda harus mempertimbangkan untuk belajar Java:

1. **Populeritas:** Java adalah salah satu bahasa pemrograman paling populer di dunia. Banyak perusahaan besar dan organisasi menggunakan Java untuk pengembangan perangkat lunak mereka. Dengan mempelajari Java, Anda akan memiliki banyak peluang karir dan dapat terlibat dalam proyek-proyek yang signifikan.
2. **Keterbukaan Platform:** Java dirancang agar dapat dijalankan pada berbagai platform, termasuk Windows, macOS, Linux, dan bahkan perangkat seluler. Hal ini membuat Java menjadi pilihan yang kuat untuk pengembangan aplikasi yang ingin mencapai basis pengguna yang luas.
3. **Pemrograman Berorientasi Objek:** Java mendukung pemrograman berorientasi objek secara penuh. Konsep pemrograman berorientasi objek sangat penting dalam pengembangan perangkat lunak modern, karena memungkinkan modularitas, fleksibilitas, dan pemeliharaan yang lebih baik.
4. **Ekosistem yang Kuat:** Java memiliki ekosistem yang sangat kuat, dengan berbagai library, framework, dan alat yang tersedia. Anda dapat memanfaatkan kelas-kelas dan fungsi-fungsi yang sudah ada untuk mempercepat pengembangan aplikasi Anda. Beberapa contoh framework populer untuk Java adalah Spring, Hibernate, dan JavaFX.
5. **Keselamatan dan Keamanan:** Java memiliki fitur keamanan yang kuat yang dirancang untuk melindungi sistem dari ancaman keamanan yang mungkin disebabkan oleh program yang berjalan. Konsep sandbox Java membatasi akses program terhadap sistem operasi dan lingkungan di mana mereka dijalankan.
6. **Mobilitas:** Java juga sangat populer dalam pengembangan aplikasi seluler. Dengan menggunakan platform Java seperti Android, Anda dapat membuat aplikasi untuk jutaan perangkat Android di seluruh dunia.
7. **Pengembangan Web:** Java juga digunakan secara luas dalam pengembangan aplikasi web. Dengan framework seperti JavaServer Faces (JSF), JavaServer Pages (JSP), dan Spring MVC, Anda dapat membangun aplikasi web yang kuat dan skalabel.
8. **Peluang Karir yang Luas:** Karena popularitasnya yang tinggi dan adopsi yang luas, belajar Java memberi Anda peluang karir yang luas. Banyak perusahaan mencari pengembang Java yang terampil untuk mengembangkan dan memelihara perangkat lunak mereka.

Belajar Java juga memberikan landasan yang kuat untuk memahami konsep pemrograman yang lebih luas dan mempersiapkan Anda untuk belajar bahasa pemrograman lain di masa depan.

## 2. Java di dunia kerja

Java memiliki peran yang signifikan di dunia kerja dan merupakan salah satu bahasa pemrograman yang paling banyak digunakan dalam industri perangkat lunak. Berikut adalah beberapa contoh bagaimana Java digunakan dalam dunia kerja:

1. **Pengembangan Aplikasi Perusahaan:** Banyak perusahaan menggunakan Java untuk mengembangkan aplikasi bisnis yang kompleks. Java digunakan dalam pembangunan sistem manajemen database, aplikasi keuangan, sistem manajemen persediaan, dan banyak lagi.

Java memiliki keunggulan dalam kemampuan untuk mengintegrasikan dengan teknologi lain dan sistem backend yang ada.

2. Pengembangan Aplikasi Web: Java digunakan secara luas dalam pengembangan aplikasi web. Dengan menggunakan framework seperti Spring, JavaServer Faces (JSF), dan Java Persistence API (JPA), pengembang dapat membangun aplikasi web yang kuat, aman, dan skalabel.
3. Pengembangan Aplikasi Seluler: Java adalah bahasa pemrograman utama dalam pengembangan aplikasi Android. Dengan menggunakan bahasa pemrograman Java dan Android SDK, pengembang dapat membuat aplikasi untuk jutaan perangkat Android di seluruh dunia.
4. Pengembangan Permainan: Java digunakan dalam pengembangan permainan, baik permainan desktop maupun permainan seluler. Platform seperti libGDX dan JavaFX menyediakan alat dan kerangka kerja untuk mengembangkan permainan dengan Java.
5. Pengembangan Aplikasi IoT: Dalam Internet of Things (IoT), Java digunakan dalam pengembangan perangkat embedded dan sistem yang terhubung. Java menyediakan kerangka kerja yang memungkinkan pengembang untuk menghubungkan perangkat keras dengan perangkat lunak menggunakan konektivitas jaringan.
6. Pengembangan Middleware: Java juga digunakan dalam pengembangan middleware dan aplikasi server. Java EE (Enterprise Edition) menyediakan kerangka kerja dan spesifikasi untuk mengembangkan aplikasi server-side yang skalabel dan dapat diandalkan.
7. Analisis Data: Java digunakan dalam pengembangan perangkat lunak untuk analisis data dan pemrosesan besar-besaran (big data). Kerangka kerja seperti Apache Hadoop dan Apache Spark menggunakan Java untuk mengembangkan solusi analisis data yang kuat.
8. Pekerjaan di Industri IT: Keterampilan Java sangat dicari dalam industri IT. Banyak perusahaan teknologi dan perusahaan pengembang perangkat lunak mencari pengembang Java yang terampil untuk mengisi peran pengembangan perangkat lunak, pengembang web, pengembang aplikasi seluler, dan lainnya.

Java memiliki kehadiran yang kuat di berbagai industri, dan keahlian dalam Java dapat membuka banyak peluang karir. Belajar Java dan menguasai konsep dan alat terkait dapat membantu Anda membangun karir yang sukses dan memberi Anda keunggulan dalam pasar kerja yang kompetitif.

### 3. Java - Environment Setup

Untuk memulai pengembangan dengan Java, Anda perlu melakukan pengaturan lingkungan kerja yang tepat. Berikut adalah langkah-langkah umum untuk mengatur lingkungan kerja Java:

1. Unduh dan Instal JDK (Java Development Kit): JDK adalah paket perangkat lunak yang mencakup kompiler Java, runtime environment, dan berbagai alat dan utilitas untuk pengembangan Java. Anda dapat mengunduh JDK dari situs resmi Oracle (<https://www.oracle.com/java/technologies/downloads/>). Pilih versi JDK yang sesuai dengan sistem operasi Anda dan ikuti petunjuk instalasinya.
2. Konfigurasi Variabel Lingkungan (Environment Variables): Setelah menginstal JDK, Anda perlu mengkonfigurasi variabel lingkungan di sistem operasi Anda. Pada Windows, buka "System Properties" dan tambahkan path ke direktori bin JDK ke dalam variabel lingkungan "Path". Pada Linux atau macOS, buka terminal dan tambahkan path JDK ke dalam variabel lingkungan "PATH" dalam file profil shell (misalnya, ~/.bashrc atau ~/.bash\_profile).

3. Verifikasi Instalasi JDK: Buka terminal (command prompt) dan ketik perintah `java -version`. Jika Anda melihat output yang mencantumkan versi JDK yang diinstal, berarti instalasi JDK berhasil.
4. Pilih Text Editor atau Integrated Development Environment (IDE): Anda dapat menggunakan text editor biasa seperti Notepad++ atau Sublime Text untuk menulis kode Java. Namun, lebih disarankan untuk menggunakan IDE (Integrated Development Environment) seperti Eclipse, IntelliJ IDEA, atau NetBeans yang menyediakan fitur-fitur yang kuat dan alat bantu untuk pengembangan Java.
5. Buat Proyek atau Berkas Java: Dalam IDE atau text editor Anda, buat proyek atau berkas Java baru. Proyek Java biasanya terdiri dari satu atau lebih berkas Java dengan ekstensi `.java`. Setiap berkas Java harus memiliki nama yang sama dengan kelas utama di dalamnya.
6. Tulis dan Jalankan Kode Java: Mulailah menulis kode Java di dalam berkas yang telah Anda buat. Pastikan Anda menyimpan berkas dengan ekstensi `.java`. Setelah menulis kode, Anda dapat mengkompilasi berkas Java menjadi bytecode menggunakan perintah `javac` dalam terminal. Misalnya, untuk mengkompilasi berkas `Main.java`, Anda dapat menjalankan perintah `javac Main.java`. Setelah berhasil dikompilasi, Anda dapat menjalankan program menggunakan perintah `java` diikuti dengan nama kelas utama. Misalnya, `java Main`.

#### 4. Tools Java

Beberapa tools yang umum digunakan dalam pengembangan perangkat lunak dengan bahasa pemrograman Java:

1. JDK (Java Development Kit): JDK adalah paket pengembangan resmi dari Oracle yang mencakup kompiler, runtime environment, dan alat-alat pendukung untuk pengembangan aplikasi Java. JDK menyediakan peralatan penting seperti Java Compiler (`javac`), Java Runtime Environment (JRE), dan Java Development Tools (seperti debugger dan profiler).
2. IDE (Integrated Development Environment): IDE adalah perangkat lunak yang menyediakan lingkungan terpadu untuk pengembangan perangkat lunak. Beberapa IDE yang populer untuk pengembangan Java antara lain:
  - a. Eclipse: Eclipse adalah IDE open-source yang memiliki fitur lengkap untuk pengembangan Java dan mendukung juga bahasa pemrograman lainnya.
  - b. IntelliJ IDEA: IntelliJ IDEA adalah IDE yang kuat dan populer untuk pengembangan Java yang dikembangkan oleh JetBrains.
  - c. NetBeans: NetBeans adalah IDE open-source yang juga menyediakan fitur-fitur lengkap untuk pengembangan Java.
3. Build Tools:
  - a. Apache Maven: Maven adalah alat manajemen proyek yang populer untuk pengembangan Java. Maven menyediakan struktur proyek yang terorganisir, manajemen dependensi, dan memudahkan proses build, pengujian, dan distribusi proyek.
  - b. Gradle: Gradle adalah alat build modern dan fleksibel yang dapat digunakan untuk membangun, menguji, dan mengelola proyek Java dengan mudah.
4. Version Control System (VCS):
  - a. Git: Git adalah sistem kontrol versi yang populer yang digunakan oleh banyak pengembang perangkat lunak. Git memungkinkan Anda untuk melacak perubahan pada kode sumber Anda dan berkolaborasi dengan tim pengembang lainnya.
5. Debugger dan Profiler:

- a. JDK Debugger: JDK menyediakan Debugger bawaan yang dapat digunakan untuk memperbaiki bug dalam kode Java.
- b. VisualVM: VisualVM adalah alat monitoring dan profiling yang kuat yang dapat membantu dalam analisis performa dan penemuan bug pada aplikasi Java.

6. Text Editor:

- a. Notepad++: Notepad++ adalah editor teks yang ringan dan mendukung penyorotan sintaks untuk bahasa pemrograman Java.
- b. Sublime Text: Sublime Text adalah editor teks yang populer dengan banyak fitur yang dapat meningkatkan produktivitas pengembangan.

Selain itu, ada juga alat-alat tambahan seperti library/framework, pengujian, dan alat-alat bantu lainnya yang dapat digunakan sesuai dengan kebutuhan dan preferensi Anda dalam pengembangan aplikasi Java.

## 7. Java - Basic Syntax

Dalam Java, sintaks dasar adalah aturan-aturan yang mengatur cara penulisan kode Java yang valid. Berikut adalah beberapa konsep dasar sintaks Java:

1. Penamaan Variabel:

- Nama variabel harus dimulai dengan huruf atau tanda garis bawah (\_).
- Variabel bersifat case-sensitive, artinya huruf besar dan huruf kecil dibedakan.
- Nama variabel harus mengikuti aturan camel case, yaitu kata pertama dimulai dengan huruf kecil, dan setiap kata berikutnya dimulai dengan huruf besar, tanpa spasi di antaranya.

2. Mendeklarasikan Variabel:

- Untuk mendeklarasikan variabel, gunakan tipe data diikuti dengan nama variabel.
- Contoh: `int jumlah;`

3. Menginisialisasi Variabel:

- Variabel dapat diinisialisasi dengan memberikan nilai awal kepadanya pada saat deklarasi.
- Contoh: `int jumlah = 5;`

4. Tipe Data Dasar:

Beberapa tipe data dasar dalam Java antara lain: `int`, `double`, `boolean`, `char`, `String`.

- Tipe data `int` digunakan untuk menyimpan nilai bilangan bulat.
- Tipe data `double` digunakan untuk menyimpan nilai bilangan desimal.
- Tipe data `boolean` digunakan untuk menyimpan nilai `true` atau `false`.
- Tipe data `char` digunakan untuk menyimpan karakter tunggal.
- Tipe data `String` digunakan untuk menyimpan rangkaian karakter.

5. Komentar:

- Komentar digunakan untuk memberikan penjelasan atau catatan di dalam kode.
- Komentar satu baris dimulai dengan `//`, contohnya: `// Ini adalah komentar satu baris.`
- Komentar multi-baris dimulai dengan `/*` dan diakhiri dengan `*/`, contohnya:

```
1. /*
2.   Ini adalah komentar
3.   multi-baris.
4. */
```

## 6. Struktur Kontrol:

- Struktur kontrol digunakan untuk mengatur alur eksekusi program.
- Contoh struktur kontrol:
  - 1) if: Digunakan untuk menjalankan blok kode jika suatu kondisi terpenuhi.
  - 2) for: Digunakan untuk melakukan perulangan sejumlah kali yang telah ditentukan.
  - 3) while: Digunakan untuk melakukan perulangan selama kondisi tertentu terpenuhi.
  - 4) switch: Digunakan untuk melakukan pemilihan kondisi dengan beberapa kemungkinan nilai.

## 7. Fungsi atau Metode:

- Fungsi atau metode digunakan untuk mengelompokkan serangkaian pernyataan yang dapat dipanggil dari tempat lain dalam program.
- Fungsi dapat menerima argumen dan mengembalikan nilai.
- Contoh deklarasi fungsi:

```
1. public int tambah(int a, int b) {  
2.     int hasil = a + b;  
3.     return hasil;  
4. }
```

Ini hanya beberapa konsep dasar sintaks Java. Dalam pengembangan yang lebih lanjut, Anda akan menjumpai konsep-konsep lain seperti array, penggunaan objek, inheritance, dan banyak lagi.

## Contoh Program

### Contoh 1 :

Berikut adalah contoh program sederhana dalam bahasa pemrograman Java:

```
1. public class HelloWorld {  
2.     public static void main(String[] args) {  
3.         System.out.println("Hello, World!");  
4.     }  
5. }  
6.
```

Program di atas adalah program "Hello, World!" yang merupakan program sederhana yang umum digunakan sebagai langkah awal dalam mempelajari bahasa pemrograman baru.

### Penjelasan:

- **public class HelloWorld** adalah deklarasi kelas dengan nama HelloWorld.
- **public static void main(String[] args)** adalah metode utama (entry point) dari program yang akan dieksekusi saat program dijalankan.
- **System.out.println("Hello, World!")** digunakan untuk mencetak teks "Hello, World!" ke output konsol.

Untuk menjalankan program di atas, simpan kode program ke dalam file dengan nama HelloWorld.java. Kemudian, jalankan perintah `javac HelloWorld.java` untuk mengkompilasi program dan `java HelloWorld` untuk menjalankannya. Setelah itu, Anda akan melihat output "Hello, World!" dicetak di konsol.

## Contoh 2 :

Berikut adalah contoh program sederhana dalam bahasa pemrograman Java yang dilengkapi dengan komentar untuk menjelaskan setiap bagian program:

```
1. public class Calculator {
2.     // Metode utama yang akan dieksekusi saat program dijalankan
3.     public static void main(String[] args) {
4.         // Menyimpan angka 10 ke dalam variabel a
5.         int a = 10;
6.
7.         // Menyimpan angka 5 ke dalam variabel b
8.         int b = 5;
9.
10.        // Memanggil metode add() dan menyimpan hasilnya ke dalam variabel sum
11.        int sum = add(a, b);
12.
13.        // Menampilkan hasil penjumlahan
14.        System.out.println("Hasil penjumlahan: " + sum);
15.    }
16.
17.    // Metode untuk menjumlahkan dua angka
18.    public static int add(int num1, int num2) {
19.        // Menjumlahkan dua angka dan mengembalikan hasilnya
20.        int result = num1 + num2;
21.        return result;
22.    }
23. }
24.
```

### Penjelasan Program:

- **public class Calculator:** Pendeklarasian kelas dengan nama Calculator.
- **public static void main(String[] args):** Metode utama yang akan dieksekusi saat program dijalankan.
- **int a = 10; dan int b = 5;** Mendeklarasikan variabel a dan b yang bertipe data integer dan menyimpan nilai 10 dan 5.
- **int sum = add(a, b);** Memanggil metode add() dengan argumen a dan b, dan menyimpan hasil penjumlahan ke dalam variabel sum.
- **System.out.println("Hasil penjumlahan: " + sum);** Menampilkan hasil penjumlahan menggunakan metode println() dari kelas System.
- **public static int add(int num1, int num2):** Deklarasi metode add() yang menerima dua parameter num1 dan num2 bertipe data integer dan mengembalikan nilai bertipe integer.
- **int result = num1 + num2; dan return result;** Menjumlahkan dua angka yang diterima sebagai argumen dan mengembalikan hasilnya.

Komentar-komentar yang ditambahkan pada setiap baris program memberikan penjelasan tentang fungsi dan tujuan setiap bagian program. Komentar digunakan untuk membantu pembaca memahami kode program dengan lebih mudah dan menjelaskan logika di balik setiap pernyataan.



## Latihan

### Latihan Praktikum: Memperbaiki Struktur/Sintaks Program Java

Instruksi: Pada setiap latihan, Anda akan diberikan potongan kode program Java yang memiliki kesalahan struktur atau sintaks. Tugas Anda adalah memperbaiki kode tersebut sehingga program dapat berjalan tanpa kesalahan.

#### Latihan 1:

```
1. public class MyProgram
2.     public static void main(String[] args) {
3.         System.out.println("Hello, World!")
4.     }
5. }
6.
```

#### Latihan 2:

```
1. public class Calculation {
2.     public static void main(String[] args) {
3.         int x = 5;
4.         int y = 10;
5.         int sum = x + y;
6.         System.out.println("Hasil penjumlahan: " + Sum);
7.     }
8.
```

#### Latihan 3:

```
1. public class Circle {
2.     public static void main(String[] args) {
3.         double radius = 5.0
4.         double circumference = 2 * Math.PI * radius;
5.         System.out.println("Keliling lingkaran: " + Circumference);
6.     }
7. }
8.
```

Periksa kembali setiap latihan setelah memperbaiki kesalahan. Pastikan tidak ada kesalahan struktur atau sintaks yang tersisa.

**Catatan:** Latihan ini bertujuan untuk meningkatkan pemahaman Anda tentang struktur dan sintaks dasar dalam pemrograman Java. Dalam latihan sebenarnya, Anda dapat diberikan potongan kode yang lebih kompleks atau masalah yang lebih menantang.

## Tugas

### Tugas Praktikum: Program Kalkulator Sederhana

#### Instruksi:

Buatlah sebuah program kalkulator sederhana yang dapat melakukan operasi penjumlahan, pengurangan, perkalian, dan pembagian. Program harus menerima input dari pengguna dan menampilkan hasil perhitungan.

#### Spesifikasi:

1. Program harus memiliki metode utama `main()` untuk menjalankan program.
2. Program harus menerima input dari pengguna berupa dua angka dan operator matematika (+, -, \*, /).
3. Program harus memvalidasi input pengguna dan memberikan pesan kesalahan jika input tidak valid (misalnya, operator yang tidak dikenali atau input bukan angka).
4. Program harus melibatkan metode-metode terpisah untuk setiap operasi matematika (penjumlahan, pengurangan, perkalian, pembagian).
5. Program harus menampilkan hasil perhitungan ke layar.

#### Contoh Output:

```
=== KALKULATOR SEDERHANA ===  
  
Masukkan angka pertama: 5  
Masukkan operator (+, -, *, /): *  
Masukkan angka kedua: 3  
  
Hasil: 15
```

```
=== KALKULATOR SEDERHANA ===  
  
Masukkan angka pertama: 10  
Masukkan operator (+, -, *, /): /  
Masukkan angka kedua: 0  
  
Error: Tidak dapat membagi dengan nol!
```

```
=== KALKULATOR SEDERHANA ===  
  
Masukkan angka pertama: ABC  
Error: Masukkan angka yang valid!
```

#### Tugas Tambahan:

1. Tambahkan fitur agar pengguna dapat mengulang perhitungan atau mengakhiri program setelah melihat hasil.
2. Berikan opsi kepada pengguna untuk memilih operasi matematika menggunakan angka (misalnya, 1 untuk penjumlahan, 2 untuk pengurangan, dst.) dan gunakan switch-case untuk menentukan operasi yang akan dilakukan.

Silakan gunakan kreativitas Anda dalam mengimplementasikan tugas ini. Anda dapat menambahkan fitur tambahan sesuai dengan kemampuan dan minat Anda.