WEEK 6

Learning Progress Review

API

JAVA

Charlie's Angel Team

DigitalSkola

{API}

API merupakan singkatan dari Application Programming Interface, yaitu sekumpulan definisi dan protokol untuk membangun dan mengintegrasikan aplikasi software. Singkatnya, API adalah pengembangan dan inovasi software yang memungkinkan berbagai aplikasi bertukar data dan fungsionalitas dengan mudah dan aman

Kegunaan API

- Memudahkan Membangun Aplikasi yang Fungsional
- Pengembangan Aplikasi Menjadi Lebih Efisien
- Meringankan Beban Server
- Menghandle scope yang lebih besar
- Membuat sebuah aplikasi/feature lebih versatile
- Memudahkan komunikasi antar service

- **RPC**

RPC merupakan teknologi untuk membuat komunikasi antara client side dan server side bisa dilakukan dengan konsep sederhana. RPC memiliki dua jenis, yaitu XML-RPC dan JSON-RPC. Sesuai namanya, XML-RPC menggunakan format XML sebagai media perpindahan data, sedangkan JSON-RPC menggunakan JSON untuk perpindahan data

Arsitektur API

- **SOAP**

Arsitektur API lainnya adalah SOAP (Simple Object Access Protocol). Arsitektur ini menggunakan XML (Extensible Markup Language) yang memungkinkan semua data disimpan dalam dokumen

Arsitektur API

Arsitektur API

- **REST**

REST atau Representational State Transfer adalah arsitektur API yang cukup populer karena kemudahan penggunaannya. Tak perlu coding yang panjang untuk menggunakannya.
REST menggunakan JSON sebagai bentuk datanya sehingga lebih ringan. Performa aplikasi pun menjadi lebih baik.

Status HTTP Response

- 100 : Continue
- 101 : Switching Protocols
- 103 : Early Hints
- 200 : OK
- 201 : Created
- 202 : Accepted
- 203 : Non – Authoritative Information
- 204 : No Content

- 300 : Multiple Choices
- 301 : Moved Permanently
- 302 : Found
- 400 : Bad Request
- 401 : Unauthorized
- 403 : Forbidden
- 404 : Not Found
- 500 : Internal Server Error
- 502 : Bad Gateway
- 504 : Gateway Timeout

Python Framework : Flask

Flask adalah sebuah web framework yang ditulis dengan bahasa Python dan tergolong sebagai jenis microframework. Flask berfungsi sebagai kerangka kerja aplikasi dan tampilan dari suatu web. Dengan menggunakan Flask dan bahasa Python, pengembang dapat membuat sebuah web yang terstruktur dan dapat mengatur behaviour suatu web dengan lebih mudah.

API METHODS

- GET

GET adalah method yang biasa digunakan untuk mendapatkan data dari server. Ketika kita melakukan request dengan method GET maka server akan mencarikan data yang sesuai dengan kebutuhan kita dan kemudian mengembalikan data tersebut melalui response. Inilah yang biasa disebut dengan aksi READ

API METHODS

- POST

POST adalah method yang biasa digunakan untuk membuat data baru pada database yang ada di server. Kita dapat mengirimkan data yang akan dibuat melalui body dari request yang kita kirim. Inilah yang biasa disebut dengan aksi CREATE

API METHODS

- PUT

PUT adalah method yang biasa digunakan untuk mengupdate data yang sudah ada pada database yang ada di server. Mirip seperti POST namun method ini biasanya digunakan untuk mengubah data bukan membuat. Inilah yang biasa disebut dengan aksi UPDATE

API METHODS

- PATCH

PATCH adalah method yang biasa digunakan untuk mengupdate data yang sudah ada pada database yang ada di server. Mirip seperti PUT namun method ini biasanya digunakan untuk mengupdate beberapa field dalam sebuah record saja, bukan mengupdate semua field dalam sebuah record seperti PUT.

API METHODS

- DELETE

Delete adalah method yang biasa digunakan untuk menghapus data yang sudah ada pada databse yang ada di server. Inilah yang biasa disebut dengan aksi DELETE

DigitalSkola

JAVA HISTORY

- Dikembangkan oleh James Gosling pada tahun 1991
- Sebelumnya sempat dinamai Green, kemudian diganti menjadi Oak, dan akhirnya diubah menjadi Java.
- Java pada dasarnya sudah merupakan pemrograman berorientasi objek (OOP)
- Merupakan pengembangan dari bahasa C dan C++

TINGKAT BAHASA PEMROGRAMAN

Bahasa pemrograman tingkat tinggi : C++, Java, Python

Bahasa pemrograman tingkat menengah : C, Pascal, Fortran

Bahasa pemrograman tingkat rendah : Assembly

Semakin tinggi level suatu bahasa pemrograman maka akan semakin mirip dengan bahasa manusia

Java Terminology

- JVM (Java Virtual Machine)

  Mesin yang menyediakan environment untuk mnejalankan program Java. JVM akan mengeksekusi bytecode yang dihasilkan oleh compiler pada fase running program.

- Bytecode

  Kode yang dihasilkan saat *source code* di*compile* oleh *compiler* , sehingga dapat dieksekusi oleh JVM.

- JDK (Java Development Kit)

  Merupakan set kit yang diperlukan untuk membuat dan menjalankan program Java. JVM, development tools, dan JRE termasuk kedalam JDK.

Java Terminology

- JRE (Java Runtime Environment)

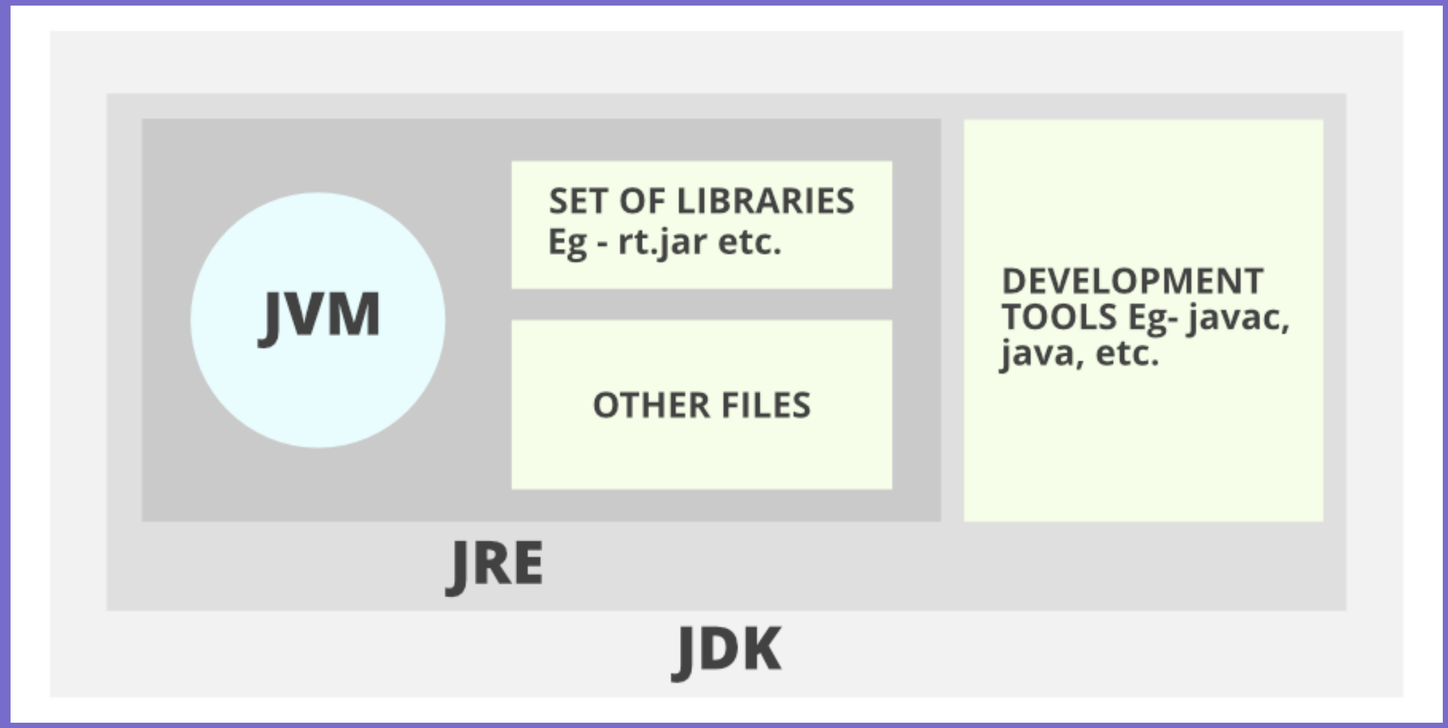
  JRE menyediakan lingkuran agar program Java dapat dieksekusi.JRE menyediakan seperangkat libraries serta file lain yang digunakan JVM saat runtime. JRE merupakan implementasi dari JVM

- Garbage Collector

  Merupakan fitur dari JVM yang berfungsi untuk menghapus dan mengigat kembali memori sebuah objek.

- ClassPath

  Merupakan parameter yang menjelaskan jalur file Java yang akan dijalankan. Classpath dapat disetting bersamaan dengan setting environmenta variables.



Source: Geeksforgeeks

Berikut adalah ilustrasi hubungan JDK, JRE, dan JVM

Main Features of Java

- **Platform independent** : Write Once Run Anywhere. Byetcode yang dihasilkan oleh compiler Java memungkinkan program Java dijalankan di platform manapun.
- **Object Oriented Programming Languange**: Semua diperlakukan sebagai objek dalam Java, sehingga cara pemrogramannya sudah berorientasi objek.
- **Simple**: Memiliki syntax yang cukup mudah dipahami dan tidak memerlukan pointer seperti pada bahasa pemrograman C++
- **Robust**: Dilengkapi fitur garbage collection, exceptional handling, dan alokasi memori yang membuat Java bersifat robust.
- **Secure**: Karena tidak memiliki pointer, Java tidak memungkinkan akses array diluar batas. Java juga memiliki environment tersendiri sehingga berjalan terpisah dengan OS

Basic Terminology

- Object: Turunan dari class yang memiliki attribut dan behaviour
- Class: Sekelompok objek yang bersifat umum
- Method: fungsi yang mendefinisikan perilaku objek
- Instance Variables: varibel non statis yang didefinisikan didalam class namun diluar method dan constructor

Java Command

Untuk menjalankan sebuah program Java, dapat menggunakan command berikut pada file yang dimaksud

```
Javac Hello.java (compile)

Java Hello.java (execute)
```

Basic Syntax Rules

DigitalSkola

Java Comments

Single Line Comment: //
contoh: //System.out.println(“ini comment“);

Multi Line Comment: /* .... */
contoh: /*System.out.println(“ini comment“);*/

Documentation Comment: /** ...*/
contoh : /** Documentation*/

Program File Name

- Penamaan program file harus sama dengan nama class (kecuali didalam file tidak ada class public).
- Ekstensi file .java

Contoh :
HelloWorld.java (valid syntax)
helloworld.java (invalid syntax)

Basic Syntax Rules

DigitalSkola

Case Sensitivity

- Java bersifat case sensitive. Sehingga Java akan memperlakukan variabel A dan a sebagai 2 variabel yang berbeda.

Contoh :
System.out.println (valid syntax)
system.out.println(invalid syntax)

Class Names

- Penamaan class dalam Java diawali dengan huruf besar
- Jika nama class lebih dari 1 kata maka kata ke-2 diawali dengan huruf besar (camel's case)atau underscore

Contoh :
class StudentApp (valid syntax)
class studentApp (invalid syntax)

Basic Syntax Rules

DigitalSkola

Method Names

- Kebalikan dari class, penamaan method diawali dengan huruf kecil.
- Jika terdiri dari 2 kata atau lebih, maka kata ke 2 dan seterusnya diawali dengan huruf besar (mixed case)

Contoh :
studentSubject (valid syntax)
StudentSubject(invalid syntax)

Identifiers

- Diawali alphabet atau underscore
- Karakter pertama dapat berupa uppercase, lowercase, ataupun underscore.
- Keyword Java tidak bisa dijadikan identifiers.

Contoh :
_90sgeneration (valid syntax)
90sgeneration (invalid syntax)

Basic Syntax Rules

DigitalSkola

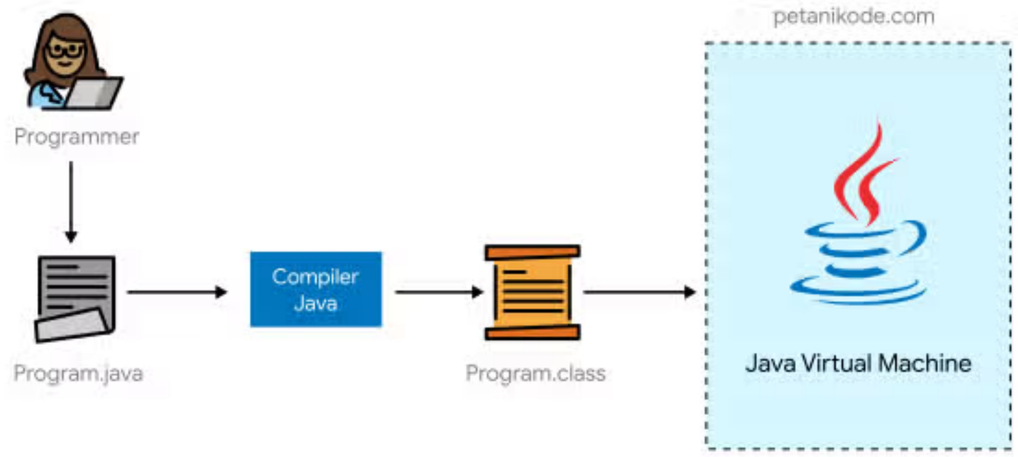
Access Modifiers

- Hak akses yang diberikan pada variabel, class, atau method.
- Terdiri dari 4 jenis : default, public, protected, private

Modifiers	Package	Subclass	World
Public	Yes	Yes	Yes
Protected	Yes	Yes	No
Default (No Modifier)	Yes	No	No
Private	No	No	No

Tabel Access Modifier

Proses pemrograman Java



1 Menulis kode program Java

Pertama Programmer menulis kode program Java dan akan menghasilkan file dengan nama **Program.java**.

2 Melakukan Compile

File **Program.java** kemudian di-compile dengan compiler (javac) dan akan menghasilkan file Program.class.

3 Menjalankan Program

File **Program.class** merupakan file yang berisi bytecode. Bytecode ini merupakan kode yang dipahami JVM. Mirip-mirip seperti bahasa assembly. Bytecode akan dieksekusi oleh JVM, sehingga program pun berjalan.

DigitalSkola

DigitalSkola

Variabel

Variabel (Variable) adalah wadah untuk penyimpanan data atau nilai sementara yang nantinya juga dapat dimanipulasi oleh program kita.

3 Jenis variabel pada pemrograman java

1 Local Variables

Local variable (variabel lokal) adalah variabel yang dideklarasikan dan hanya dapat digunakan di dalam method, constructors atau block itu sendiri.

2 Instance Variables

Instance Variable adalah variabel yang dideklarasikan di class. Tidak di dalam method, constructor atau blok apapun. Variabel ini dapat diberikan access modifier dan bisa dideklarasikan tanpa harus menentukan nilai bawaan.

3 Class/Static Variables

Seperti nama nya, “static” yang artinya statis tidak berubah-ubah. Variable ini digunakan untuk mendeklarasikan sebuah variabel yang nilainya statis dan tidak berubah-ubah secara dinamis; biasanya digunakan sebagai konstanta.

Modifier pada Java

Modifier adalah sebuah keyword yang memberikan keterangan tambahan dan ditempatkan di awal deklarasi kelas, konstruktor, atribut, maupun method.

1 Modifier Akses

Modifier akses (access modifier) adalah keyword yang menyatakan hak akses. Fungsi dari modifier akses adalah membatasi kepada siapa saja suatu obyek dapat diakses.

2 Modifier Non-Akses

Modifier akses (access modifier) adalah keyword yang menyatakan hak akses. Fungsi dari modifier akses adalah membatasi kepada siapa saja suatu obyek dapat diakses.

DigitalSkola

Flow Control Java

DigitalSkola

Decision-Making (Percabangan) statement

if-else statement

Percabangan **if-else** dipakai untuk mengeksekusi salah satu dari 2 pernyataan dari syarat tertentu yang pada pada if yang dapat bernilai benar atau salah. Sintaks:

```
if (ekspresi_boolean) {
    Pernyataan1;
} else {
    Pernyataan2;
}
```

switch

switch dipakai pada saat kita ingin memberikan kondisi dengan beberapa syarat yang identik yang masing-masing mempunyai pernyataanyang berbeda-beda. SSintaks:

```
switch (ekspresi){
    case nilai1: Pernyataan1;
    break;
    case nilai2: Pernyataan2;
    break;
    default: Pernyataan3;
}
```

Flow Control Java

DigitalSkola

Looping (Perulangan) statement

for Loop

Perulangan **for** dipakai pada saat kita melakukan perulangan dengan jumlah yang sudah diketahui pasti. Sintaks:

```
for (inisialisasi; kondisi; perubah) {
    Pernyataan;
}
```

while Loop

Perulangan **while** dipakai pada saat kita melakukan perulangan dengan jumlah yang belum diketahui pasti. Pernyataan pada while akan dikerjakan setelah pengecekan kondisi pada while bernilai true. Sintaks:

```
while (kondisi) {
    Pernyataan;
}
```

Flow Control Java

DigitalSkola

Looping (Perulangan) statement

do-while loop

Perulangan **do-while** dipakai pada saat kita melakukan perulangan dengan jumlah yang belum diketahui pasti. Pernyataan pada do akan dikerjakan terlebih dahulu, baru setelah itu dilakukan pengecekan kondisi pada while. Sintaks:

```
do {
    Pernyataan;
} while (kondisi);
```

OOP Java

Object Oriented Programing

OOP adalah membuat suatu object yang berisi data dan metode

Pemrograman berorientasi objek memiliki beberapa keunggulan dibandingkan pemrograman prosedural:

- OOP lebih cepat dan lebih mudah untuk dieksekusi
- OOP menyediakan struktur yang jelas untuk program
- OOP membantu menjaga kode Java KERING "Jangan Ulangi Sendiri", dan membuat kode lebih mudah untuk dipelihara, dimodifikasi, dan di-debug
- OOP memungkinkan untuk membuat aplikasi penuh yang dapat digunakan kembali dengan lebih sedikit kode dan waktu pengembangan yang lebih singkat

Class & Objects

Class dan Objek adalah aspek utama dari OOP

Contoh :

Class	Object
Buah	**Aple** **Pisang** **Mangga**
Mobil	**Volvo** **BMW** **Audi**

Java Inheritance
(Subclass and Superclass)

Di Java memungkinkan inheriantance atribut dan methode dari satu class ke class yang lain

Injeritance di bgai menjadi 2, yaitu

- subclass (child) - class yang inheritance dari class lain
- superclass (parent) - Class awal

Contoh

```
class Vehicle {
  protected String brand = "Ford";      // Vehicle attribute
  public void honk() {                  // Vehicle method
    System.out.println("Tuut, tuut!");
  }
}

class Car extends Vehicle {
  private String modelName = "Mustang";    // Car attribute
  public static void main(String[] args) {

    // Create a myCar object
    Car myCar = new Car();

    // Call the honk() method (from the Vehicle class) on the myCar
    object
    myCar.honk();

    // Display the value of the brand attribute (from the Vehicle class)
    and the value of the modelName from the Car class
    System.out.println(myCar.brand + " " + myCar.modelName);
```

Terima Kasih

Sampai jumpa di
Learning Progress Review
kami berikutnya! sudah berpartisipasi!

Hilda Meiranita Prastika Dewi

Nur Indrasari

Rezha Sulvian

Thasha Dinya Ainsha

Charlie's Angel Team
Week 6

DigitalSkola