# **MODUL I : PENGENALAN REST API & SPRING BOOT**



**Kompetensi Pengetahuan**

1. Peserta dapat memahami Konsep Rest API
2. Peserta dapat memahami Spring Boot

**Kompetensi Keterampilan**

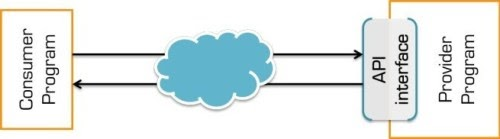
1. Peserta dapat melakukan instalasi environment yakni Java, Maven, Postman, Git, dan IntelliJ Community

## **REST API**

Dalam memahami konsep sebuah Rest API, perlu dipahami terlebih dahulu mengenai sebuah API dan implementasinya

### **API**

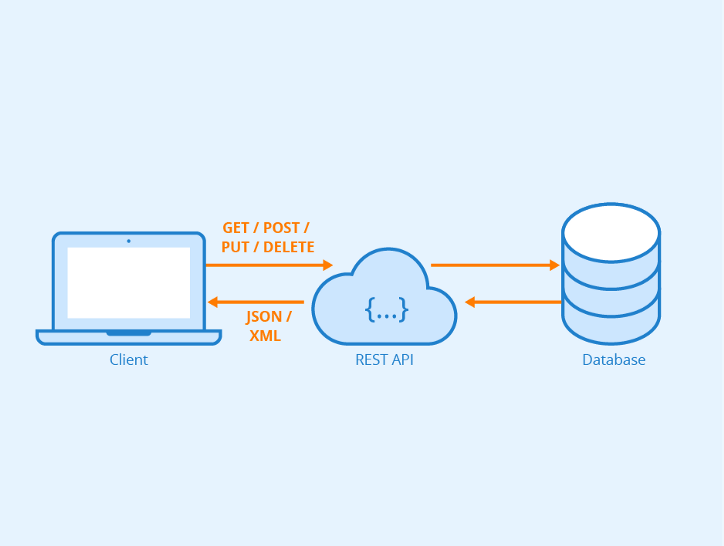
API atau Application Programming Interface adalah sebuah interface yang dapat menghubungkan aplikasi atau code satu dengan aplikasi atau kode lainnya (Sharma, 2021). Adapun tujuan API sebagai perantara antar berbagai aplikasi berbeda, baik dalam satu platform yang sama atau lintas platform sesuai gambar dibawah (Lawrence, 2020).



Selain tujuan diatas, terdapat tujuan lain penggunaan API yakni untuk mempercepat proses pengembangan aplikasi dengan cara menyediakan sebuah function yang terpisah sehingga para developer tidak perlu lagi membuat fitur yang serupa (Romadhoni, 2020). Proses ini akan mendefinisikan jenis panggilan atau permintaan yang dapat dibuat, cara pemanggilan, format data yang harus digunakan, serta konvensi yang harus diikuti.

### **REST API**

Representational State Transfer adalah desain arsitektur dari sebuah API dalam merancang sebuah aplikasi yang berbasis jaringan terdistribusi (Varanasi dan Bartkov, 2021). Pada arsitektur REST, REST server menyediakan resources (sumber daya/data) dan REST client mengakses dan menampilkan resource tersebut untuk penggunaan selanjutnya. Setiap resource diidentifikasi oleh URIs (Universal Resource Identifiers) atau global ID. Resource tersebut direpresentasikan dalam bentuk format teks, JSON atau XML. Pada umumnya formatnya menggunakan JSON dan XML.

****

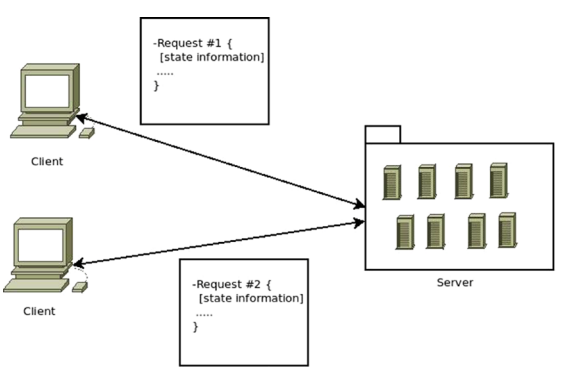
1. **Prinsip REST API**

Berdasarkan Masse (2011) terdapat 6 prinsip dasar dalam sebuah REST API yakni

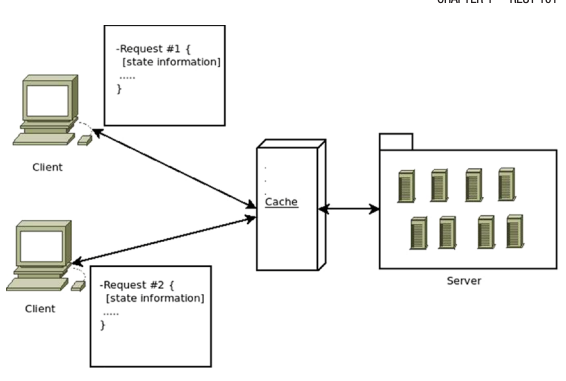
1. **Client-server** : Diimplementasikan dengan memisahkan antara client dan server sehingga komponen client atau server dapat dikembangkan secara independen dan mempunya skalabilitas yang tinggi.



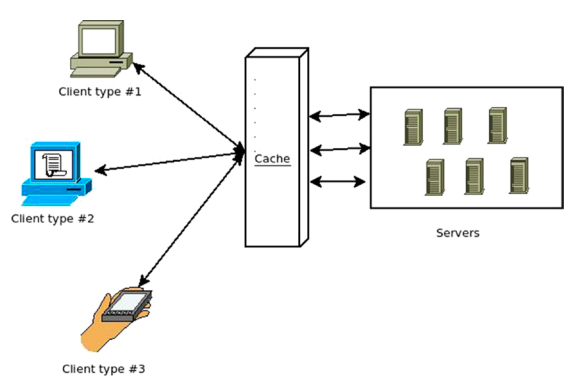
1. **Stateless** : Server tidak perlu mengingat dari data client sehingga dari client akan membutuhkan data-data yang diperlukan untuk komunikasi ke server.



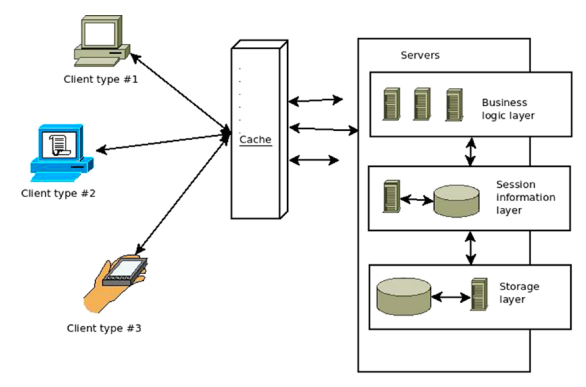
1. **Cache** : Mampu mendeklarisan web server untuk melakukan penyimpanan data sementara (cache) dari setiap data response. Dengan adanya cache ini maka waktu yang dibutuhkan interaction antara client dan server menjadi lebih cepat karena cukup memanggil data yang telah disimpan sebelumnya.



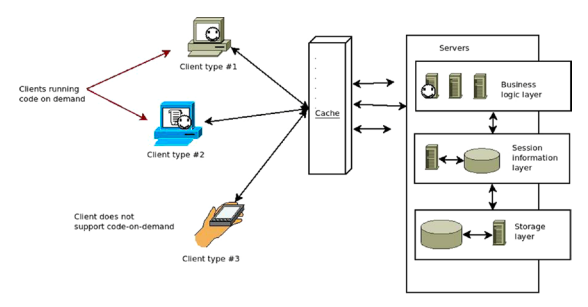
1. **Uniform Interface** : *client*, *server*, dan jaringan diantaranya berdasarkan pada keseragaman interface yang dimiliki. Jika salah satu dari komponen tersebut mempunyai interface yang berbeda dengan standar yang ditetapkan, maka komunikasi web sistem tersebut rusak.



1. **Layered system** : Di antara client dan server terdapat beberapa hierarchy layer seperti gateways, firewalls, dan proxies. Adapun layer-layer ini dapat ditambahkan, diubah, dihapus, ataupun diurutkan kembali secara transparan.



1. **Code on demand** : web server dapat melakukan transfer program sementara yang dapat dieksekusi, seperti script atau plugin ke client



1. **Cara Kerja REST API**

Sebuah client mengirimkan sebuah data atau request melalui HTTP Request dan kemudian server merespon melalui HTTP Response. REST memiliki standarisasi dalam pemakaian, yaitu URL dan HTTP method. HTTP method digunakan untuk mengetahui kegunaan dari URL yang diakses, sehingga mempermudah dalam penulisan URL. Adapun standard-standard yang bisa digunakan ketika akan mendesain dan membuat sebuah Rest API adalah sebagai berikut menurut Doglio (2018)

1. **Jenis HTTP method yang sering digunakan :**

* **GET**, digunakan untuk membaca sebuah record atau daftar record.
* **POST**, digunakan untuk menambah sebuah record.
* **PUT**, digunakan untuk mengubah semua field dalam sebuah record.
* **PATCH**, digunakan untuk mengubah beberapa field dalam sebuah record.
* **DELETE**, digunakan untuk menghapus sebuah sebuah record.

1. **Komponen dari HTTP Request :**

* **Verb**, HTTP method yang digunakan misalnya GET, POST, DELETE, PUT dll.
* **Uniform Resource Identifier (URI)** untuk mengidentifikasikan lokasi resource pada server.
* **HTTP Version**, menunjukkan versi dari HTTP yang digunakan, contoh HTTP v1.1.
* **Request Heade**r, berisi metadata untuk HTTP Request. Contoh, type client/browser, format yang didukung oleh client, format dari body pesan, setting cache dll.
* **Request Body**, konten dari data.

1. **Response Code**

Response Code adalah standar kode respons yang digunakan untuk menginformasikan suatu permintaan. Berikut contohnya :

* **1xx:** Berisi Informasi dan hanya digunakan di versi HTTP 1.1.
* **2xx:** Ketika request berhasil dan mendapatkan content yang dibutuhkan dari server.
* **3xx:** Klien harus melakukan tindakan tambahan untuk menyelesaikan permintaan.
* **4xx:** Data yang dikirimkan oleh client (request) mempunyai format yang salah atau kurang sesuai.
* **5xx:** Terdapat kendala di dalam server internal..

1. **Response Format**

Standarisasi struktur format response yang konsisten setiap kali permintaan biasanya berbentuk JSON atau XML. Contoh response dari server untuk diberikan ke client adalah seperti berikut

{

"status" : true,

"message" : "Success",

"data": {

"id" : 1,

"nama" : "Buku kisah nabi-nabi",

"harga" : 50000,

"category" : {

"id" : 3,

"nama" : "buku"

}

}

}

Jika setiap layanan mengikuti standar yang ada makan komunikasi satu sama lain akan sangat mudah karena cukup membaca pola yang ada tanpa harus konfirmasi terlebih dahulu. Adapun pada modul ini akan menggunakan standard response atau pola seperti diatas.

1. **Public API**

Menurut Dewi (2021) *Public* API merupakan jenis API terbuka yang tersedia untuk digunakan oleh *developer* atau bisnis eksternal dan menurut bri (2021) public API merupakan API yang dapat diakses oleh siapa pun untuk platform apa pun. Beberapa contoh public API yang bisa digunakan oleh developer dapat dilihat dari list berikut <https://github.com/public-apis/public-apis>. Tujuan penggunaan open API sendiri bisa menambah kebutuhan data ataupun kebutuhan fungsi-fungsi yang belum ada di dalam system yang dikembangkan.

## **Spring Boot**

Sebelum membahas mengenai Spring boot perlu diketahui lebih dahulu mengenai teknologi **Spring** dan framework-framework yang ada di dalam Spring.

### **Spring**

Berdasarkan Sharma (2021) **Spring** merupakan salah satu framework yang dikembangkan menggunakan bahasa Java. Framework ini mempunyai tujuan agar mempermudah para *programmer* dalam membuat sebuah aplikasi Java. Hal ini berdasarkan dari website official spring (spring.io, 2022) framework spring mempunyai fokus dalam hal kecepatan, sederhana, dan produktif. Adapun Beberapa Fitur yang disediakan Spring Framework adalah sebagai berikut.

1. Dependency Injection
2. Aspect Oriented Programming
3. Spring MVC and Restful Web Service
4. Support koneksi database, dll

### **Maven**

Merupakan build automation tools yang akan membantu manajemen project yang berbentuk file **pom.xml** (https://maven.apache.org, 2022)**,** yang mempunya tujuanseperti :

* Manage dependency (library) secara otomatis
* Manage build project kedalam binary application (jar) secara automate
* Melakukan automate testing secara otomatis

Selain maven, **Gradle** merupakan alternative lain untuk melakukan build automation tools.

1. **Lifecycle in Maven**

Secara sederhana **lifecycle** dapat disebut sebuah **task**, adapun **lifecycle** dalam maven adalah

1. **Validate**

Melakukan validasi project apakah sudah sesuai serta mengecek kebutuhan semua informasi sudah tersedia atau belum

1. **Compile**

Melakukan proses compile dari sebuah source code

1. **Test**

Menguji source code yang sudah tercompile dengan mengguanakn framework unit test yang sesuai, adapun test file tersebut tidak perlu di deploy ataupun di jadikan sebuah package

1. **Package**

Menjadikan sebuah package dalam format terdistribusi seperti jar dari source yang sudah tercompile

1. **Verify**

Menjalankan pengecekan atas hasil integration test yang telah dijalankan untuk memastikan quality dari test terukur

1. **Install**

Melakukan install package kedalam local repository untuk digunakan sebagai dependency sehingga bisa digunakna di project lainnya

1. **Deploy**

Sama seperti install hanya saja hasil package tersebut diberikan kedalam remote server sehingga penggunaan package tidak hanya bersifat local namun sudah public

1. **CLI Spring Boot**

Berikut meruapakn perintah - perintah yang dapat digunakan untuk CLI Maven yang sering digunakan beserta deskripsinya

| **Command** | **Description** |
| --- | --- |
| mvn --version | Menampilkan versi maven yang digunakan |
| mvn clean | Melakukan clear pada directory target |
| mvn package | Membuild project kedalam sebuah package JAR |
| mvn clean package | Melakukan clear pada directory lalu melakukan build project kedalam sebuah package JAR |
| mvn verify | Menjalankan semua integration test di dalam project |
| mvn clean verify | Melakukan clear pada directory lalu melakukan Menjalankan semua integration test di dalam project |
| mvn install | Melakukan build project berdasarkan Maven Pom file lalu hasil Jar tersebut diinstall di local maven repository |
| mvn clean install | Melakukan clear pada directory lalu melakukan build project berdasarkan Maven Pom file lalu hasil Jar tersebut diinstall di local maven repository |

### **Spring Boot**

Berdasarkan website official spring (spring.io, 2021) **Spring Boot** merupakan salah satu jenis framework dari Spring. Di dalam Spring Boot, akan lebih dipermudah dalam pembuatan program dikarenakan di dalam setiap project yang akan kita buat makan sudah disediakan sebuah Tomcat dan beberapa server lain. Sehingga dalam hal ini kita tidak perlu melakukan konfigurasi server untuk menjalankan project yang telah dibuat namun sebagai programmer cukup melakukan run project makan hasil project yang telah dikembangkan akan berjalan.

1. **Opinionated**

Java Spring Boot merupakan **opinionated** framework. Adapun opinionated framework merupakan framework yang sangat lengkap sehingga developer yang akan menggunakan framework tersebut menjadi mudah dan cepat, sebagai contoh sudah ada (Web Server, Controller, dan Data Access).

Kebalikannya Non Opinionated merupakan framework yang sangat fleksibel atau modular sehingga memudahkan developer melakukan kustomisasi pada framework tersebut, sebagai contoh hanya terdapat Web Server sedangkan controller dan data access developer perlu membuat sendiri sesuai architecture yang diinginkan

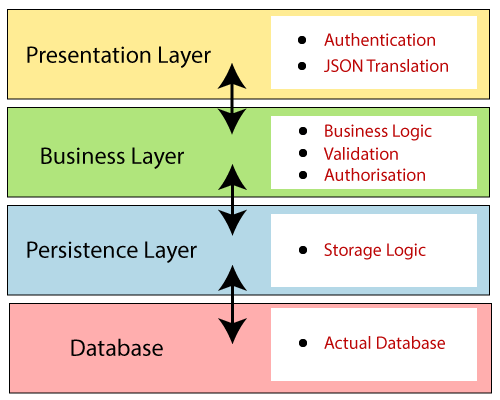
1. **Kelebihan Java Spring Boot**

Dengan menggunakan Java Spring Boot, selain dari project yang sudah servlet Spring Boot terdapat kelebihan lain menurut Nugraha (2020) yakni

* Sudah menyediakan fleksibilitas untuk mengatur Java Beans, Konfigurasi dan Database Transaction
* Mendukung batch processing dan mendukung REST endpoint (untuk aplikasi berbasis web service API).
* Dalam Spring Boot, semua sudah di auto configuration sehingga tidak perlu melakukan manual konfigurasi
* Sudah mendukung Annotation Based Konfigurasi menggunakan annotasi @SpringBootApplication
* Karena sudah terintegrasi dengan Maven, maka dapat dengan mudah mengatur plugin / dependency aplikasi.

1. **Spring Boot Architecture**

Spring Boot mempunya architecture dalam membangun sebuah Rest API yakni



1. **Presentation Layer**

Presentation Layer atau biasa disebut controller merupakan suatu layer yang digunakan untuk komunikasi ke client, sehingga jika client ada sebuah request maka akan memanggil layer controller atau presentation terlebih dahulu dan jika proses sudah selesai dan server perlu memberikan respons data biasanya berupa JSON maka layer yang bertanggung jawab adalah presentation atau controller juga.

1. **Business Layer**

Business Layer merupakan suatu layer yang berisi logika utama atau bisnis-bisnis logic yang merupakan core dari sebuah sistem server Rest API.

1. **Persistence Layer**

Persistence Layer atau biasa disebut Repository atau Data layer merupakan suatu layer yang khusus berinteraksi dengan database seperti melakukan get data, insert, update, ataupun delete dari database

1. **Database**

Database merupakan penyimpanan data utama dalam sebuah sistem server dapat bersifat SQL maupun No SQL.

## **Preparation Tools**

Dalam pengembangan sebuah aplikasi Spring Boot Rest API akan dibutuhkan beberapa tools yakni sebagai berikut

* **Git**

Berdasarkan website official git (https://git-scm.com, 2022), git adalah Git adalah sistem kontrol versi terdistribusi gratis dan open source yang dirancang untuk menangani manajemen project dengan cepat dan efisien.

* **Java**

Berdasarkan website official java (https://www.java.com, 2022), java merupakan bahasa pemrograman dan platform komputasi yang pertama kali dirilis oleh Sun Microsystems pada tahun 1995. Java telah berkembang yang dari awal sederhana dan sekarang menjadi bahasa pemrograman yang cukup banyak digunakan di seluruh dunia.

* **Maven**

Berdasarkan website official maven (https://maven.apache.org, 2022), maven adalah sebuah perangkat lunak guna memanajemen proyek berbasis konsep model objek proyek (POM) adapun Maven dapat mengelola pembuatan, pelaporan, dan dokumentasi proyek dari sebuah informasi pom file.

* **IDE (IntelliJ Community)**

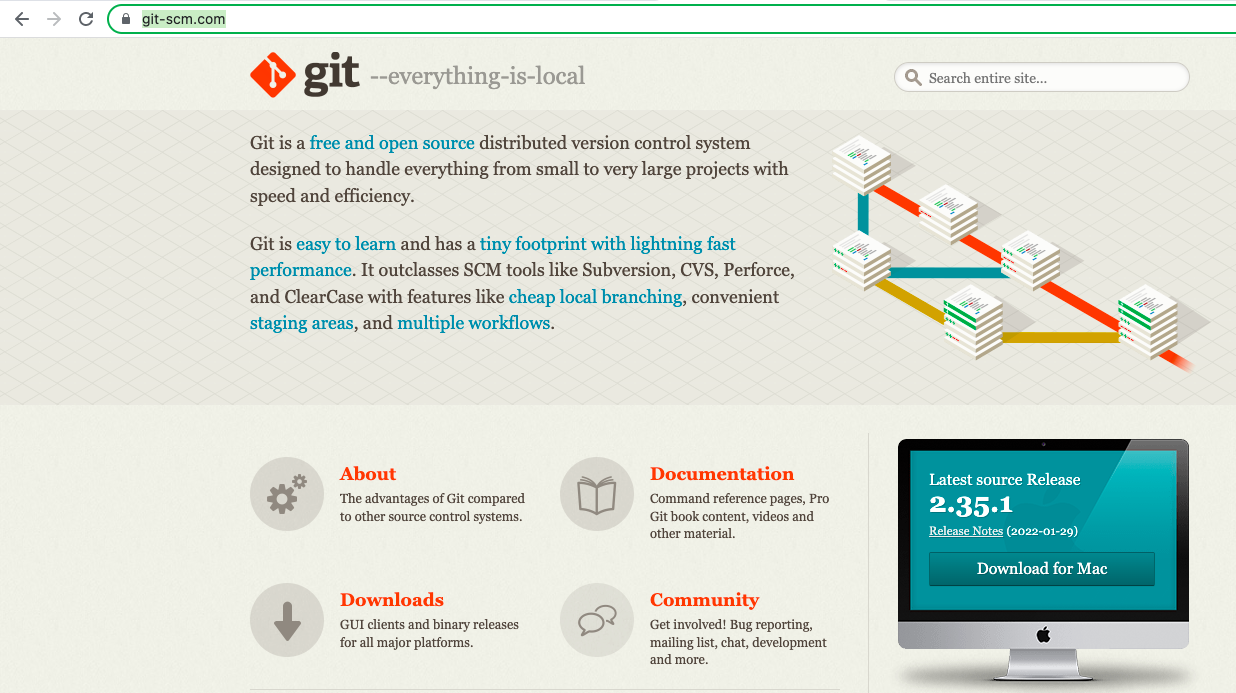
Berdasarkan wikipedia (2022) IDE (Integrated Development Environment) adalah program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak. Tujuan dari IDE adalah untuk menyediakan semua utilitas yang diperlukan dalam membangun perangkat lunak. Adapun dalam modul ini akan digunakan IDE IntelliJ Community dikarenakan kemudahan dan kehandalan dalam proses development Spring Project

Selanjutnya untuk detail langkah-langkah dalam instalasi Git, Java, Maven, dan IDE Tools adalah sebagai berikut

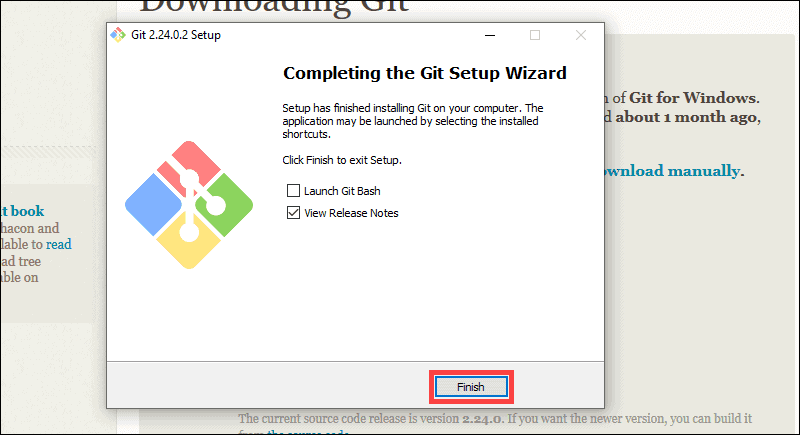
### **Git**

Langkah-langkah cara instalasi Git adalah sebagai berikut

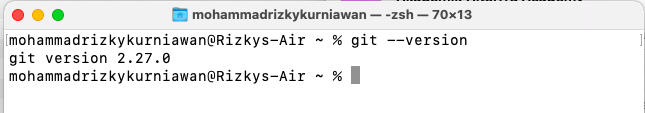
1. Buka halaman di url : <https://git-scm.com/> lalu pilih tombol Download for Windows atau Download for Mac



1. Setelah file setup git terdownload lalu lakukan instalasi seperti umumnya dengan klik tombol next hingga selesai



1. Selanjutnya untuk menguji apakah Git berhasil terinstall di dalam komputer atau laptop dengan buka command prompt atau terminal lalu ketikkan **git --version** maka nanti akan muncul versi dari git yang sudah terinstall di komputer masing-masing



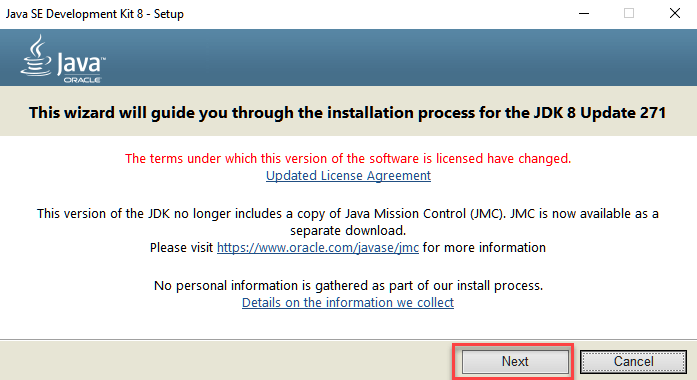
### **Java**

Langkah-langkah cara instalasi Java adalah sebagai berikut

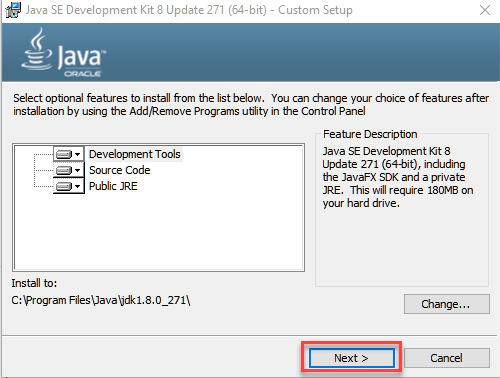
1. Buka halaman di url : <https://www.java.com/download/ie_manual.jsp> lalu Pilih tombol Agree and Start Free Download.



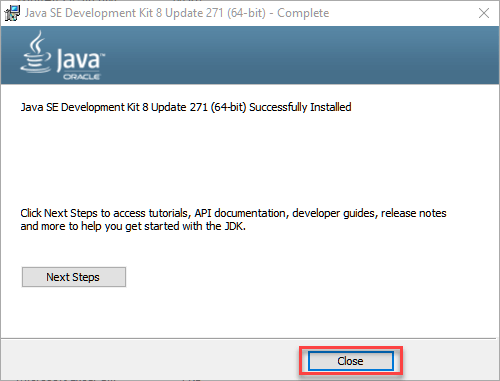
1. Setelah selesai di download, selanjutnya lakukan instalasi dengan buka file extensi .exe tersebut lalu nanti akan muncul dialog dan bisa dilakukan klik tombol next



1. Selanjutnya pilih lokasi tempat hasil instalasi java nantinya, secara default system akan mengarahkan ke dalam folder program file di drive C dan selanjutnya bisa klik tombol next.



1. Setelah proses instalasi selesai, silahkan tutup proses instalasi



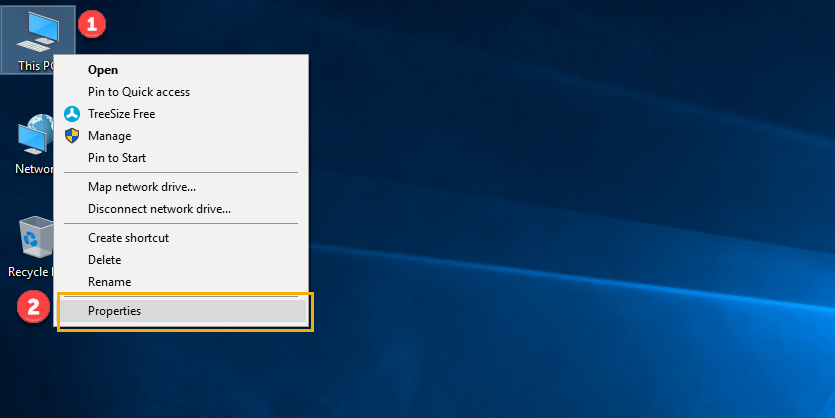
1. Selanjutnya konfigurasi Environment Variable Java (Path)



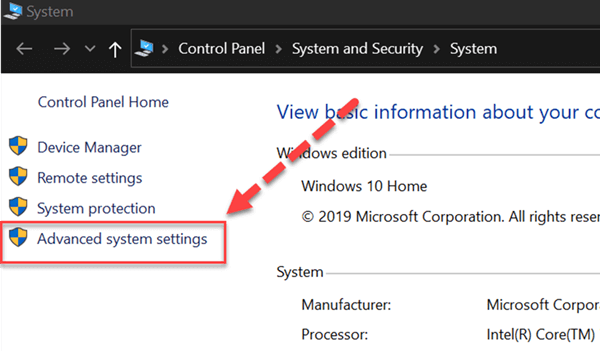
**Variabel PATH** memberikan lokasi executable seperti javac, java dll. Hal ini dimungkinkan untuk menjalankan program tanpa menentukan PATH tetapi harus ditentukan terlebih dahulu dan didaftarkan di dalam environment variable path dengan value seperti C:\Program Files\Java\jdk1.8.0\_271\bin \javac

Langkah-langkah yang diperlukan untuk konfigurasi Path Bin Java adalah sebagai berikut

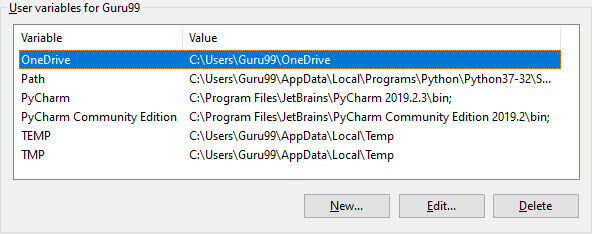
1. Buka properties di my computer dengan klik kanan pada my computer atau icon this PC dan pilih menu Properties



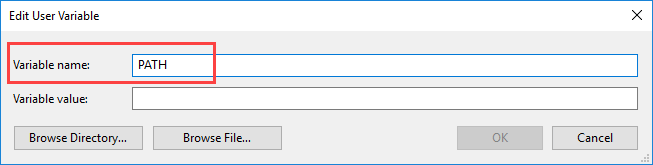
1. Selanjutnya click menu Advanced System Settings



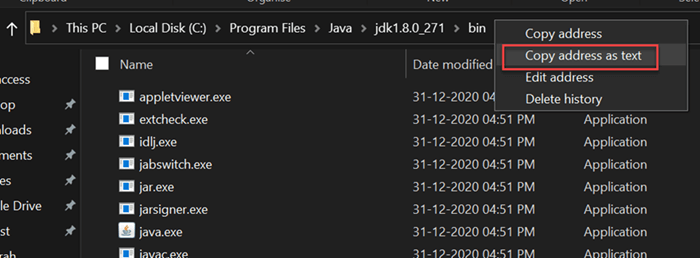
1. Lalu akan muncul dialan dan [ada bagian user variable cari variable dengan nama **Path,** jika tidak ada maka buat baru dengan pilih tombol **New** jika sudah ada selanjutnya tinggal ditambahkan dengan dipisahkan titik koma (;)



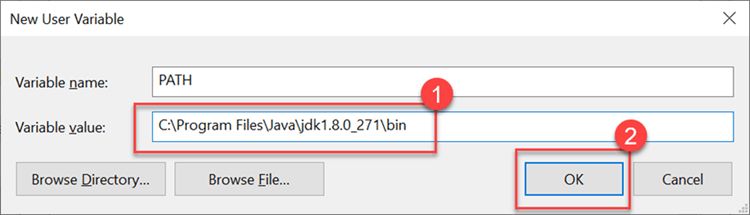
1. Ketika selesai pilih tombol New, selanjutnya masukkan variable name dengan nama baru yakni **PATH**



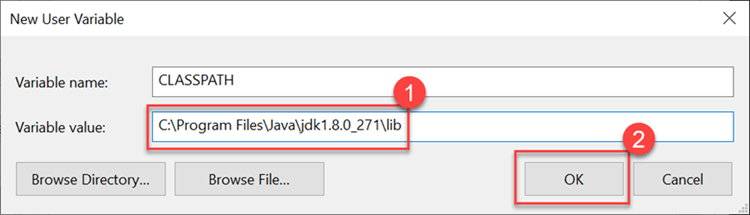
1. Isi Variable Value dengan alamat bin dari hasil instalasi java sebelumnya untuk memudahkan bukan folder Drive C -> Program Files -> Java -> jdk 1.8 -> bin, lalu klik kanan pada windows explorer di alamat folder bin dan pilih menu **Copy address as text**



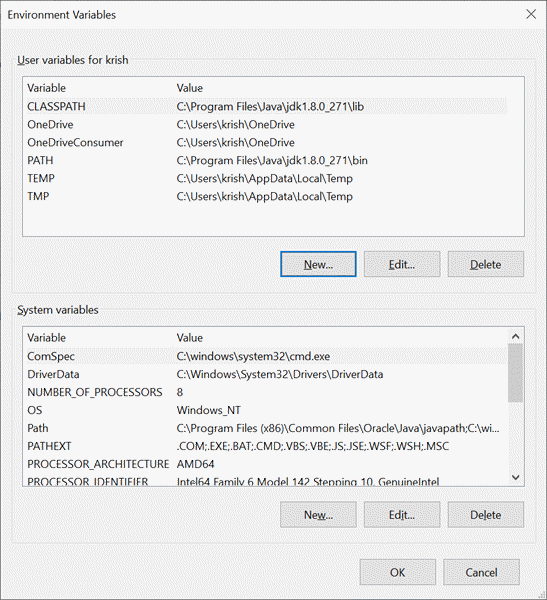
1. Masukkan sebagai value di dalam environment variable Path sebelumnya



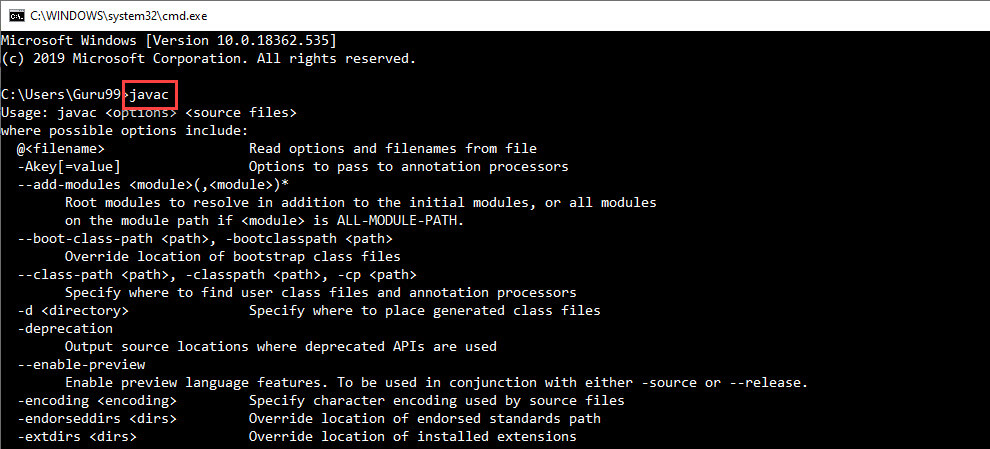
1. Di beberapa case tambahan tambahkan juga environment variable **CLASSPATH,** Jika sebelumnya mengambil folder bin untuk **Classpath** digunakan folder lib langkahnya sesuai dengan langkan sebelumnya

****

1. Setelah selesai maka akan tampil sebagai berikut



1. Selanjutnya untuk menguji apakah java yang sudah terinstall dengan baik maka dapat dilakukan dengan cara buka command prompt atau terminal lalu ketikkan **javac,** jika instalasi java berhasil akan muncul informasi terkait java yang terinstall

****

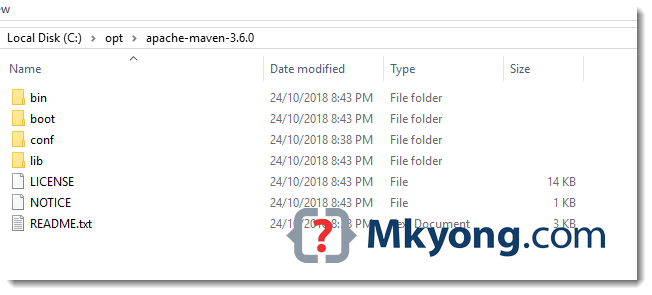
### **Maven (Optional)**

Langkah-langkah cara instalasi maven adalah sebagai berikut

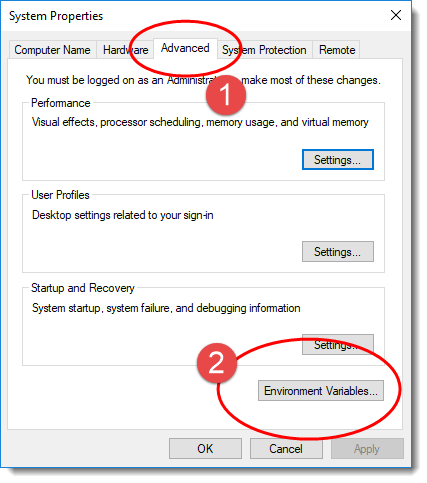
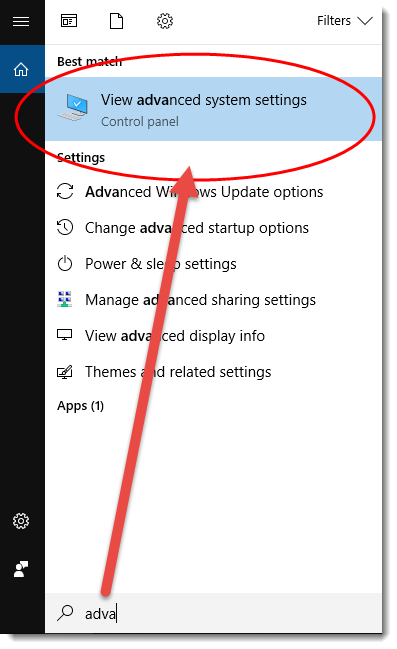
1. Download file maven di : <https://maven.apache.org/download.cgi>



1. Setelah berhasil di download letakkan extrak file binary zip tersebut lalu letakkan ke dalam sebuah folder misalkan di drive C:\apache-maven



1. Tambahkan MAVEN\_HOME variable ke dalam system environment variable di komputer anda dengan

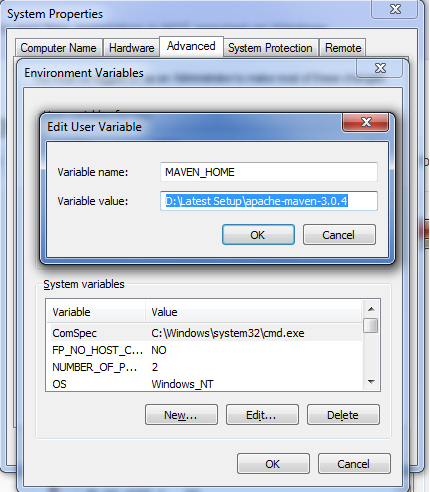




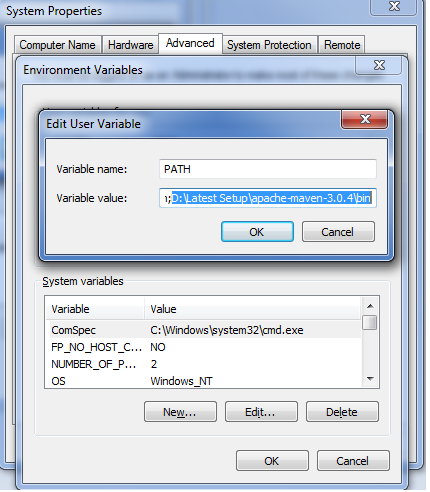
1. Di dalam dialog environment variable klik New dengan

**Variable Name :** MAVEN\_HOME

**Variable Value :** C:/apache-maven



1. Edit di dalam variable **path** lokasi bin dari maven yang sudah di extrak sebelumnya di C:/apache-maven/bin



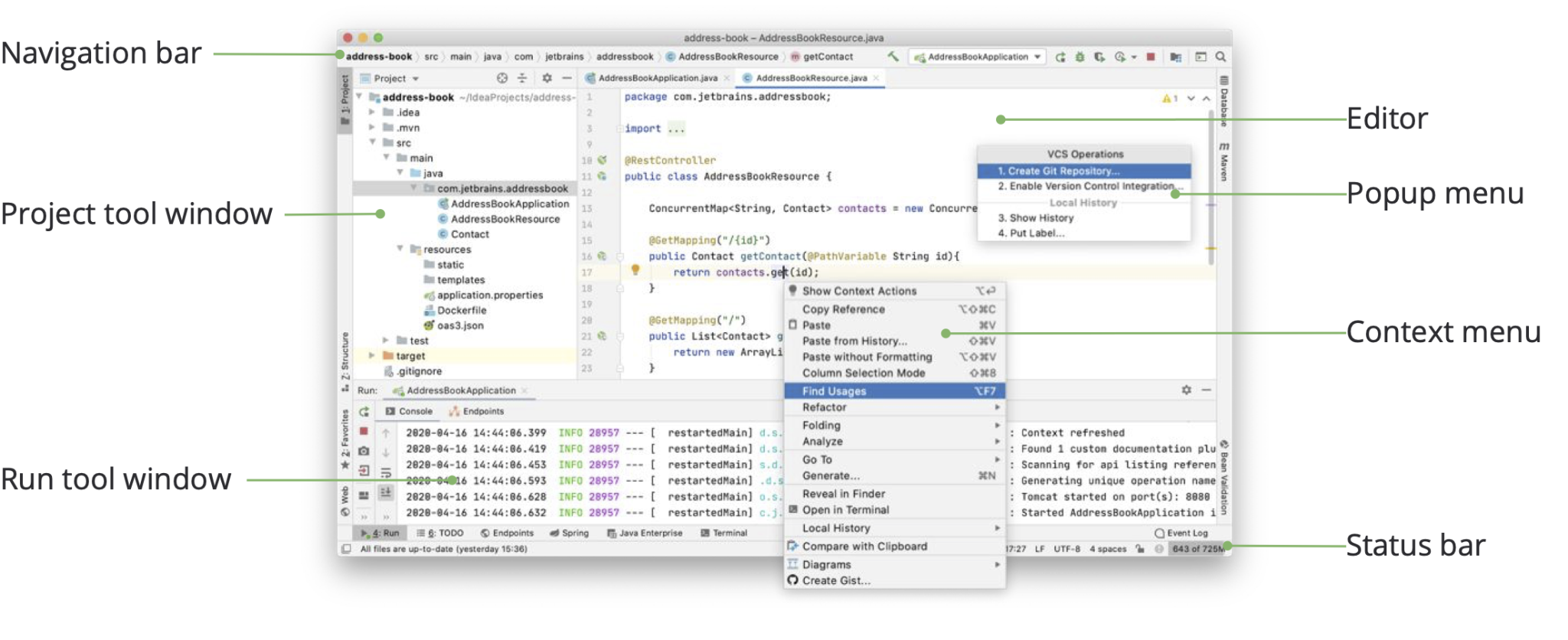
### **Integrated Development Environment (IDE) Tools Intellij Community**

Langkah-langkah cara instalasi IDE Intellij Community adalah sebagai berikut

1. Download file IDE Intellij Community di link berikut <https://www.jetbrains.com/idea/download/#section=mac> lalu klik tombol download untuk versi Community (Gratis)

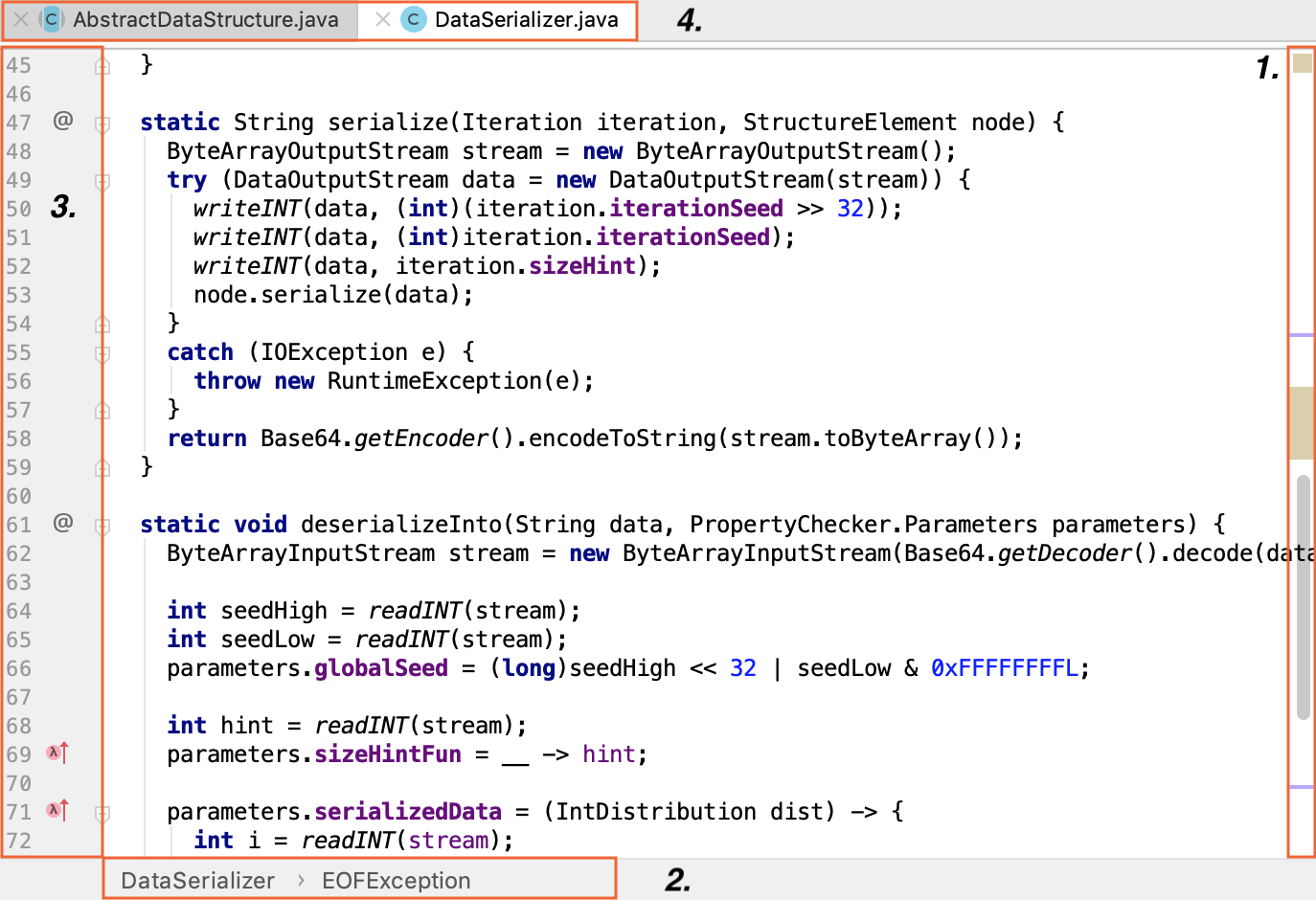


1. Lakukan proses instalasi dari file .exe tersebut, setelah selesai instalasi buka aplikasi Intellij maka akan tampil user interface sebagai berikut



Penjelasan masing-masing bagian secara lengkap sebagai berikut

1. **Editor**



Catatan :

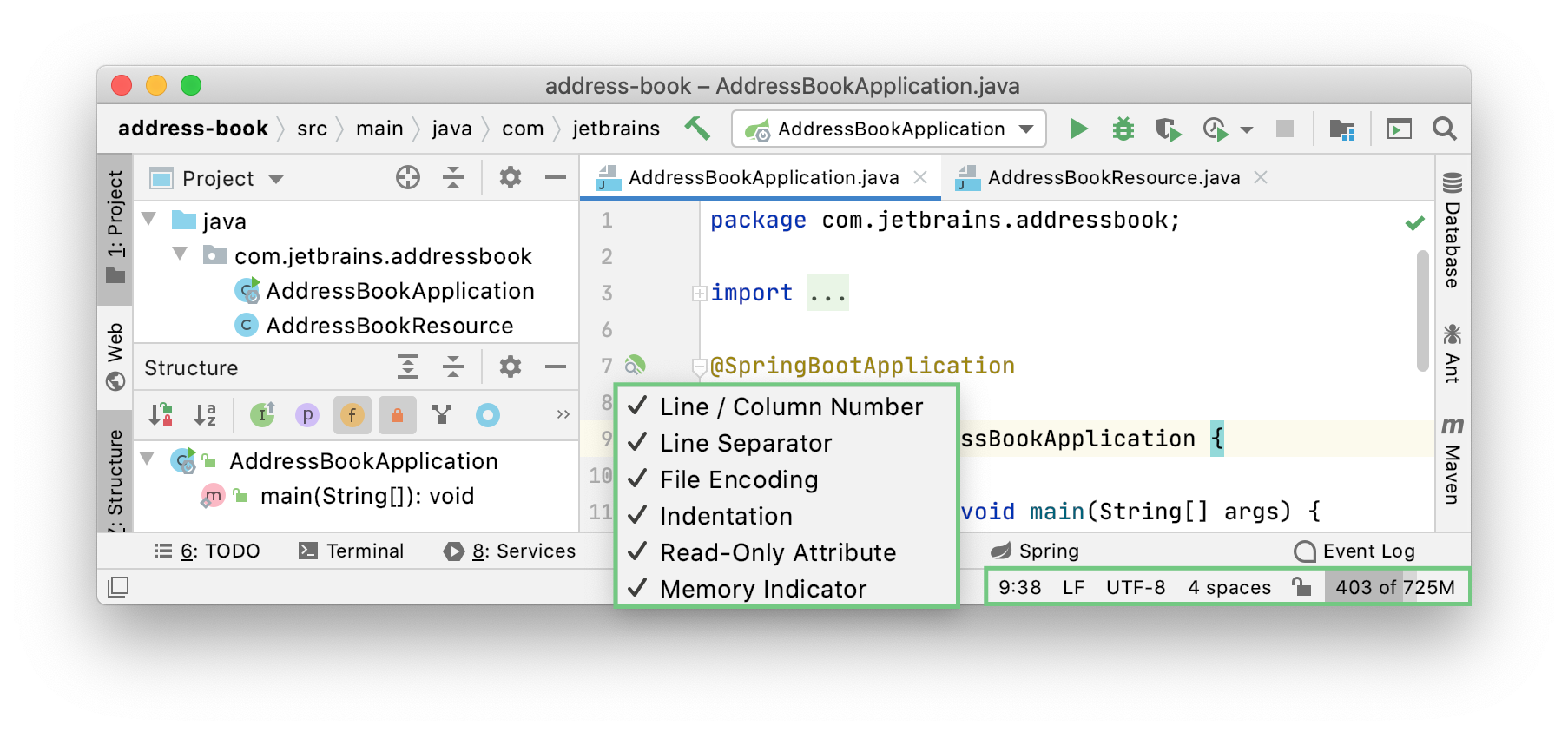
* 1 : Sebuah [**scrollbar**](https://www.jetbrains.com/help/idea/using-code-editor.html#scrollbar) yang dapat membantu menampilkan errors dan warnings di dalam file yang sedang dibuka
* 2 : Sebuah [**Breadcrumbs**](https://www.jetbrains.com/help/idea/navigating-through-the-source-code.html#editor_breadcrumbs) yang dapat membantu menavigasi ke dalam code dari file yang dibuka
* 3 : Sebuah **gutter** untuk menampilkan nomor baris dan anotasi
* 4 : Sebuah **Tabs** menampilkan nama file yang sedang dibuka.

1. **Navigation Bar**

Navigation Bar merupakan alternative cepat dalam melakukan Project View berupa navigasi structure project maupun editing files.

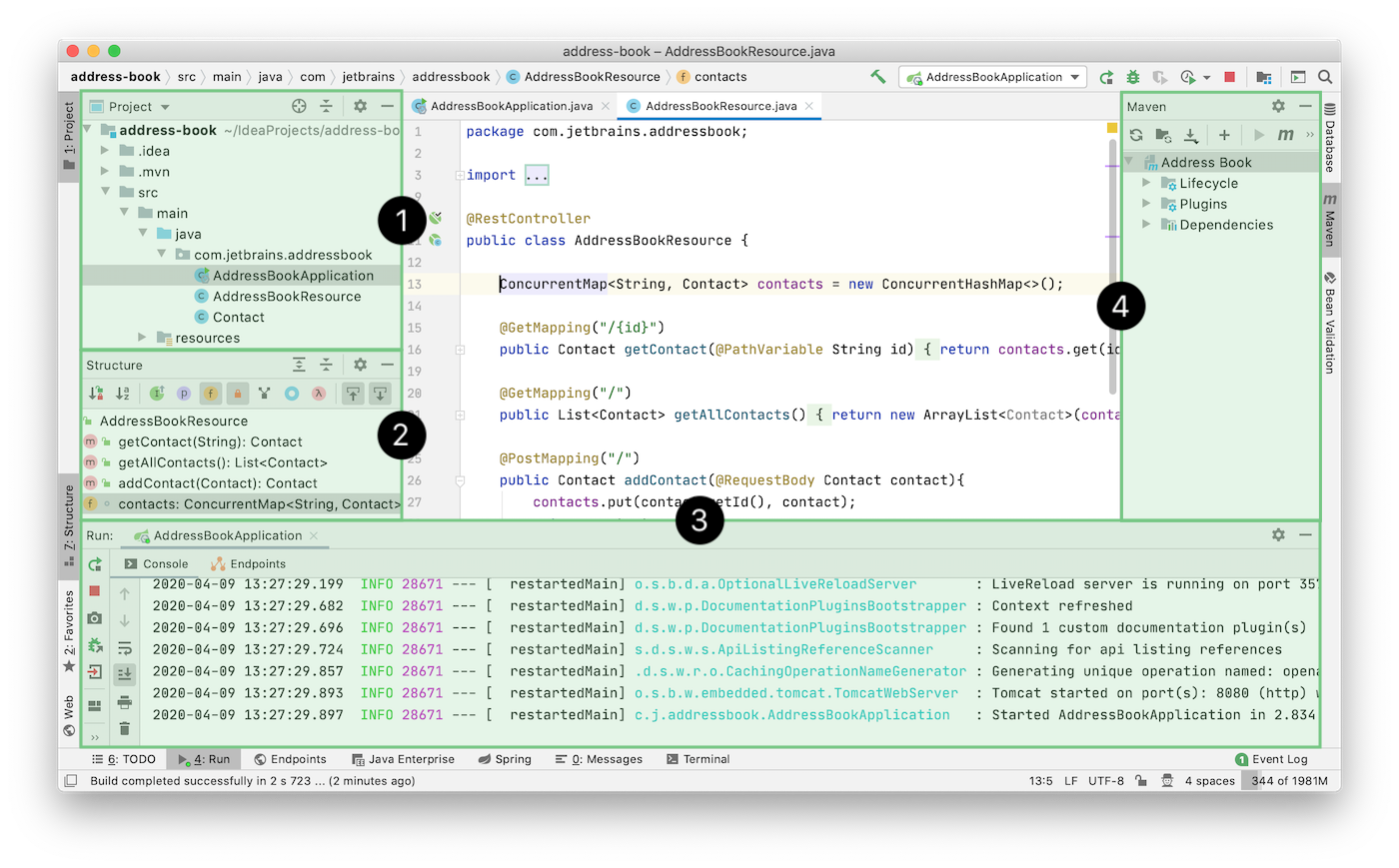
1. **Status Bar**

Pada bagian kiri di status bar dimana terletak dibawah dari main window menampilkan event messages terbaru serta deskripsi dari actions ketika melakukan hover menggunakan mouse pointer. Sedangkan bagian kanan dari status bar terdiri dari widgets yang dapat mengindikasikan status keseluruhan project and IDE serta menyediakan access ke berbagai settings.



1. **Tool Window**

Tool windows menyediakan akses berupa tugas ketika proses development yakni menampilkan struktur project, running and debugging aplikasi, integrasi dengan version control systems, code analysis, pencarian, navigation, and seterusnya.



Catatan :

* 1 : Project tool window
* 2 : Structure tool window
* 3 : Run tool window
* 4 : Maven tool window

1. **Context Menu**

Context menu dapat tampil ketika klik kanan pada sebuah element sehingga akan ditampilkan menu-menu action yang tersedia.

1. **Popup Menu**

Popup menus menyediakan cara cepat untuk melakukan actions yang berhubungan dengan context

1. **Main window**

The main window merupakan tempat jendala bekerja dengan satu project IntelliJ IDEA namun kita juga dapat membukan multiple project dalam multiple Window.

1. **IntelliJ IDEA keyboard shortcuts**

IntelliJ IDEA mempunyai keyboard shortcuts untuk melakukan perintah yang sering digunakan developer seperti perintah yang berhubungan dengan editing, navigation, refactoring, debugging, dan tasks lainnya.

| **Shortcut** | **Action** |
| --- | --- |
| Double Shift | [Search Everywhere](https://www.jetbrains.com/help/idea/searching-everywhere.html?keymap=primary_windows)  Quickly find any file, action, symbol, tool window, or setting in IntelliJ IDEA, in your project, and in the current Git repository. |
| Ctrl+Shift+A | [Find Action](https://www.jetbrains.com/help/idea/searching-everywhere.html?keymap=primary_windows#find_action)  Find a command and execute it, open a tool window, or search for a setting. |
| Alt+Enter | [Show Context Actions](https://www.jetbrains.com/help/idea/intention-actions.html?keymap=primary_windows)  Quick-fixes for highlighted errors and warnings, intention actions for improving and optimizing your code. |
| F2  Shift+F2 | [Navigate between code issues](https://www.jetbrains.com/help/idea/navigating-through-the-source-code.html?keymap=primary_windows#navigate-errors-warnings)  Jump to the next or previous highlighted error. |
| Ctrl+E | [View recent files](https://www.jetbrains.com/help/idea/navigating-through-the-source-code.html?keymap=primary_windows#recent_files)  Select a recently opened file from the list. |
| Ctrl+Shift+Enter | [Complete Current Statement](https://www.jetbrains.com/help/idea/working-with-source-code.html?keymap=primary_windows#editor_statement_select)  Insert any necessary trailing symbols and put the caret where you can start typing the next statement. |
| Ctrl+Alt+L | [Reformat Code](https://www.jetbrains.com/help/idea/reformat-and-rearrange-code.html?keymap=primary_windows#reformat_code)  Reformat the whole file or the selected fragment according to the current code style settings. |
| Ctrl+Alt+Shift+T | [Invoke refactoring](https://www.jetbrains.com/help/idea/refactoring-source-code.html?keymap=primary_windows)  Refactor the element under the caret, for example, safe delete, copy, move, rename, and so on. |
| Ctrl+W  Ctrl+Shift+W | [Extend or shrink selection](https://www.jetbrains.com/help/idea/working-with-source-code.html?keymap=primary_windows)  Increase or decrease the scope of selection according to specific code constructs. |
| Ctrl+/  Ctrl+Shift+/ | [Add/remove line or block comment](https://www.jetbrains.com/help/idea/working-with-source-code.html?keymap=primary_windows#editor_lines_code_blocks)  Comment out a line or block of code. |
| Ctrl+B | [Go To Declaration](https://www.jetbrains.com/help/idea/navigating-through-the-source-code.html?keymap=primary_windows#go_to_declaration)  Navigate to the initial declaration of the instantiated class, called method, or field. |
| Alt+F7 | [Find Usages](https://www.jetbrains.com/help/idea/find-highlight-usages.html?keymap=primary_windows#find-usages)  Show all places where a code element is used across your project. |
| Alt+1 | Focus the **Project** tool window |
| Escape | Focus the editor |

# 

# 

## **Rangkuman Modul I**

1. **API atau Application Programming Interface** adalah sebuah interface yang dapat menghubungkan aplikasi atau code satu dengan aplikasi atau kode lainnya
2. **Representational State Transfer (REST)** adalah desain arsitektur dari sebuah API dalam merancang sebuah aplikasi yang berbasis jaringan terdistribusi.
3. **Prinsip REST API** adalah Client-server, Stateless, Cache, Uniform Interface, Layered system, dan Code on demand.
4. **Jenis method HTTP** adalah GET (membaca), POST (menambah), PUT (mengubah), DELETE (menghapus)
5. **Response Code Rest API** yakni 2xx request berhasil, 4xx Data yang dikirimkan oleh client (request) mempunyai format yang salah atau kurang sesuai, 5xx: Terdapat kendala di dalam server internal.
6. **Bentuk Format Response** Rest API

{

"status" : true,

"message" : "Success",

"data": {

"id" : 1,

"nama" : "Buku kisah nabi-nabi",

"harga" : 50000,

"category" : {

"id" : 3,

"nama" : "buku"

}

}

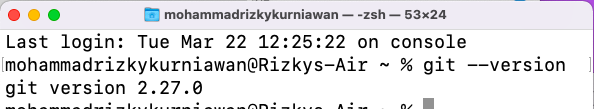
}

1. **Spring** merupakan salah satu framework yang dikembangkan menggunakan bahasa Java.
2. **Maven** merupakan build automation tools yang akan membantu manajemen project yang berbentuk file pom.xml.
3. **Spring Boot** merupakan salah satu jenis framework dari Spring. Di dalam Spring Boot, akan lebih dipermudah dalam pembuatan program dikarenakan di dalam setiap project yang akan kita buat makan sudah disediakan sebuah Tomcat dan beberapa server lain.
4. **Spring Boot Architecture** : Presentation Layer (Controller), Business Layer (Service), Persistence Layer (Repository)
5. P**reparation Tools**  : Git, Java, Maven, IntelliJ Community

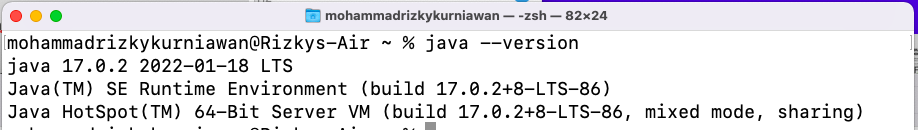
# 

## **Tugas Praktikum (Ketrampilan) Modul I**

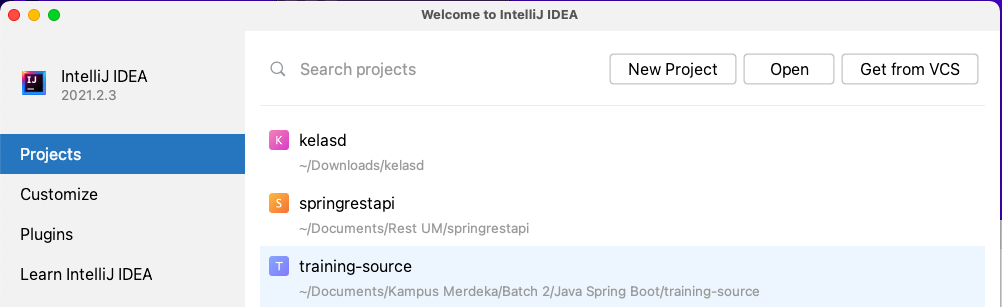
1. Lakukan Installasi Git dan berikan perintah git --version ke dalam terminal pastikan akan muncul version dari seperti berikut



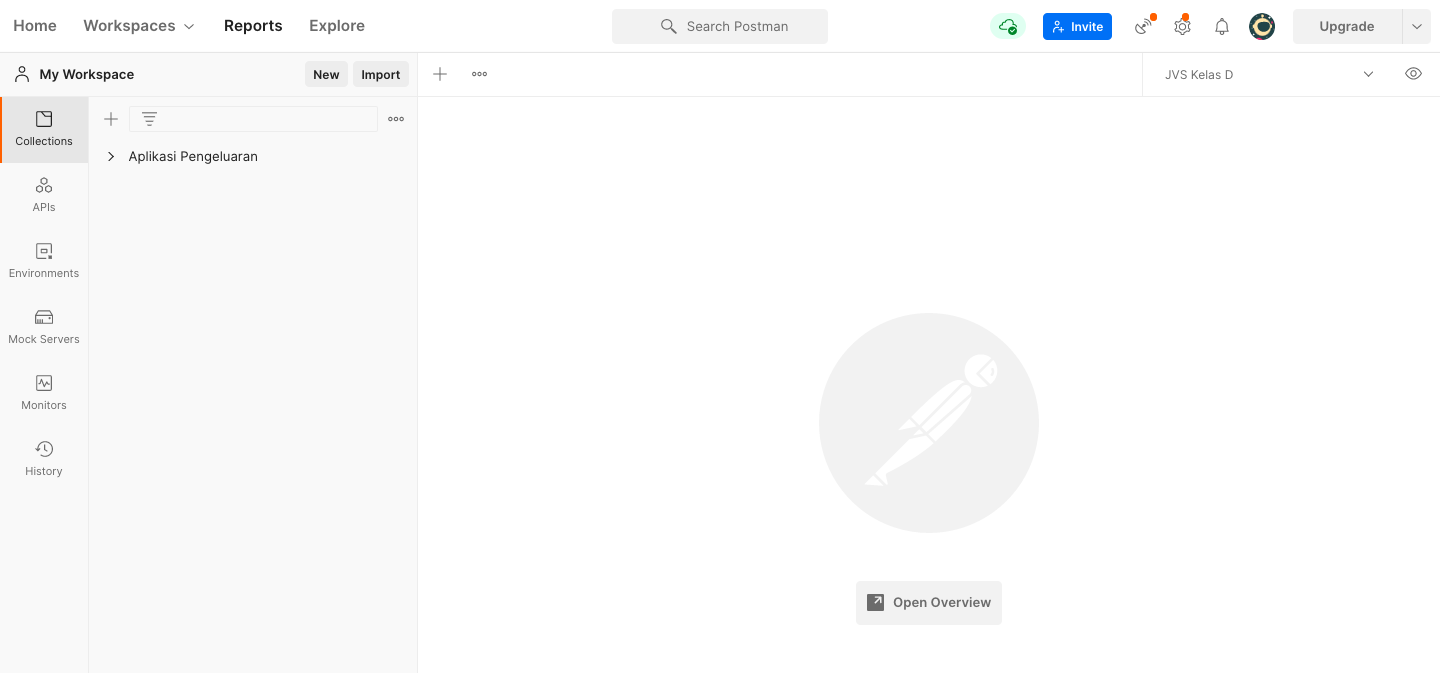
1. Lakukan Installasi Java JDK dan berikan perintah java --version ke dalam terminal pastikan akan muncul version dari Java JDK seperti berikut



1. Lakukan Instalasi IntelliJ Community dan pastikan sudah dapat terbuka welcome window IntelliJ



1. Lakukan Instalasi Postman dan pastikan sudah dapat terbuka welcome window postman



# 

## **Tugas Pilihan Ganda (Pengetahuan) Modul I**

1. Buka Link Berikut (Token : 111)

[**https://forms.gle/j5yULKEoDzW8x8FJ7**](https://forms.gle/j5yULKEoDzW8x8FJ7)

1. Isikan identitas dan Jawab Soal Pilihan Ganda dalam form tersebut dengan waktu 10 menit
2. Jumlah Soal terdapt 10 Soal