

MODUL MATA KULIAH

PEMROGRAMAN BERORIENTASI OBJEK

PG061 - 3 SKS



**FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR**

**JAKARTA
NOVEMBER 2019**

TIM PENYUSUN

M. Anif, M.Kom

Samsinar, S.Kom, M.Kom

Dani Anggoro, S.Kom, M.Kom



MODUL PERKULIAHAN #1 **Pengenalan PBO dengan Bahasa Pemrograman Java**

Capaian Pembelajaran	:	Mahasiswa Mampu: <ol style="list-style-type: none">1. Menjelaskan tentang Sejarah, Platforms, Keunggulan, Kerangka Program, Fase Pengembangan, Perangkat Keras dan Perangkat Lunak Untuk Kebutuhan Pengembangan Aplikasi dengan Bahasa Pemrograman Java,2. Serta Implementasi Kebudiluhuran Dalam Pengembangan Aplikasi Di Dunia Nyata.
Sub Pokok Bahasan	:	<ol style="list-style-type: none">1. Sejarah2. Platforms3. Keunggulan4. Kebutuhan Perangkat Lunak dan Perangkat Keras5. Kerangka Program6. Fase Pengembangan Program7. Kebudiluhuran

Daftar Pustaka	:	<ol style="list-style-type: none"> 1. Farrell. Joyce, An Object-Oriented Approach to Programming Logic and Design 3rd Edition, Course Technology, 2009 2. Thomas Wu. C, An Introduction to Object-Oriented Programming with Java™ 4th Edition, McGraw Hill, 2006 3. Deitel, Java How To Program, Deitel, Prentice Hall, 2008 4. Anif, M, Seri Aplikasi : Aplikasi Penjualan (Pemesanan Barang) dengan Java, Mitra Wacana Media, 2008
----------------	---	--

PRAKTIKUM 1

PENGENALAN PBO DENGAN BAHASA PEMROGRAMAN JAVA

1.1 Sejarah Bahasa Pemrograman Java

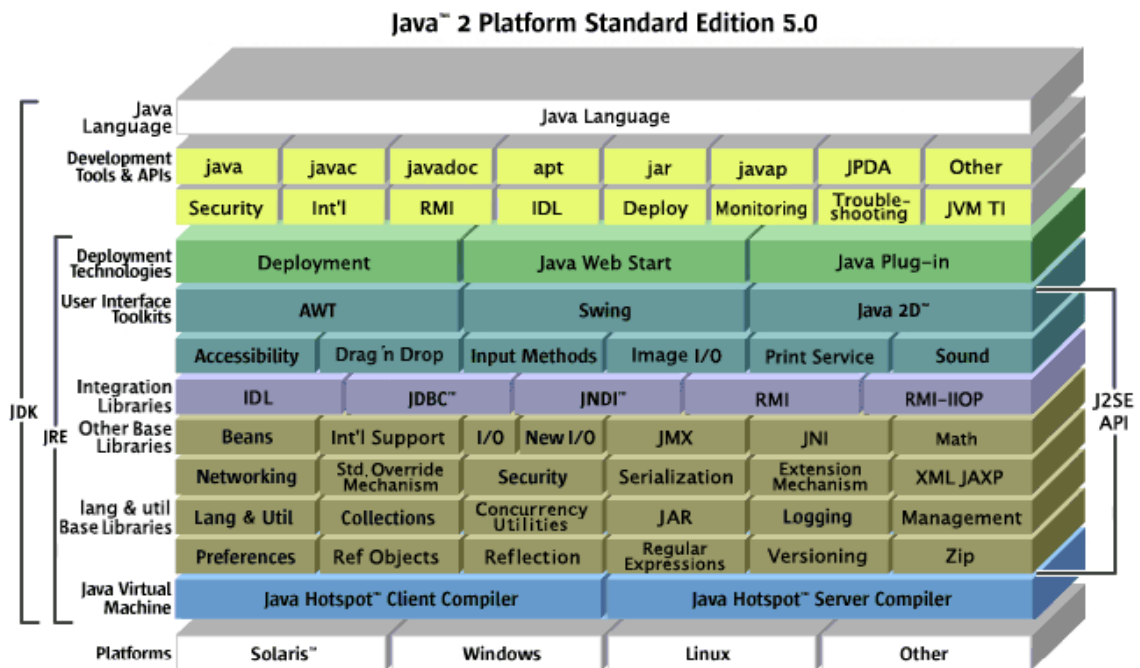
Originalnya Bahasa Pemrograman Java Di kembangkan Oleh Sun Microsystems yang di pelopori oleh James Gosling

1. Release pertama tahun 1995 untuk J2SE
2. Release pertama tahun 1999 untuk J2SE
3. Release pertama tahun 2001 untuk J2SE
4. Release terakhir Java Standard Edition yaitu Java SE 8. tentunya dengan kemampuan yang lebih baik, begitu juga dengan J2EE untuk Aplikasi Enterprise dan J2ME untuk Aplikasi Mobile.
5. Selanjutnya J2 ini di sebut dengan Java SE, Java EE, and Java ME.
6. Write Once, Run Anywhere (ditulis di satu kali dapat di jalankan di bermacam flatform).

1.2 Platform

Java adalah bahasa yang dapat dijalankan dimanapun dan di sembarang platform apapun, di beragam lingkungan: Internet, intranets, consumer electronic products, dan computer applications. Untuk beragam aplikasi yang dibuat dengan bahasa Java, Java dipaketkan dalam edisi-edisi berikut:

Java 2 Standar Edition (J2SE), J2SE menyediakan lingkungan pengembangan yang kaya fitur, stabil, aman, dan cross-platform. Edisi ini mendukung konektivitas basis data, rancangan user interface, masukkan/ keluaran (input/ output), dan pemrograman jaringan (network programming), dan termasuk sebagai paket-paket dasar bahasa Java.



Gambar 1. Java 2 Platform Standard Edition 5.0

Java didesain untuk dapat dijalankan pada lingkungan yang dinamis. Perubahan pada suatu *class* dengan menambahkan properties ataupun method dapat dilakukan tanpa mengganggu program yang menggunakan *class* tersebut



Gambar 2. Hirarki dan Portabilitas Teknologi Java

1.3 Keunggulan Bahasa Java

Java mudah dipelajari karena memiliki sintaksis yang menyerupai bahasa Inggris sehingga mudah dibaca, dipelajari dan dimengerti dengan cepat.

Java adalah bahasa pemrograman berorientasi objek. Prinsip dasar dari bahasa ini adalah segala aspek yang ada di dalam program Java dapat dipandang sebagai objek. Pematangan konsep pemrograman berorientasi objek seperti abstraction, encapsulation, polymorphism dan inheritance dipandang sangat perlu.

Java kaya akan API yang memiliki banyak sekali kelas-kelas yang sudah terdefinisi (Java predefined classes) yang dikelompokkan ke dalam paket-paket (packages). Anda dapat membuat kelas-kelas baru lebih cepat dengan memanfaatkan kelas-kelas yang ada di API Java.

Java didukung oleh perangkat pengembangan yang terintegrasi seperti Eclipse dan Netbeans. Selain menggunakan teks editor sederhana seperti Notepad dan Notepad++, Anda memiliki alternatif menggunakan perangkat gratis ini untuk mengembangkan program Java dengan lebih mudah dan lebih cepat.

Java didukung oleh komunitas yang siap membantu dan berbagi pengetahuan terkait bahasa pemrograman ini. Komunitas Java ini siap membantu programmer Java dari berbagai tingkatan, mulai dari pemula, tingkat lanjut sampai dengan yang ahli.

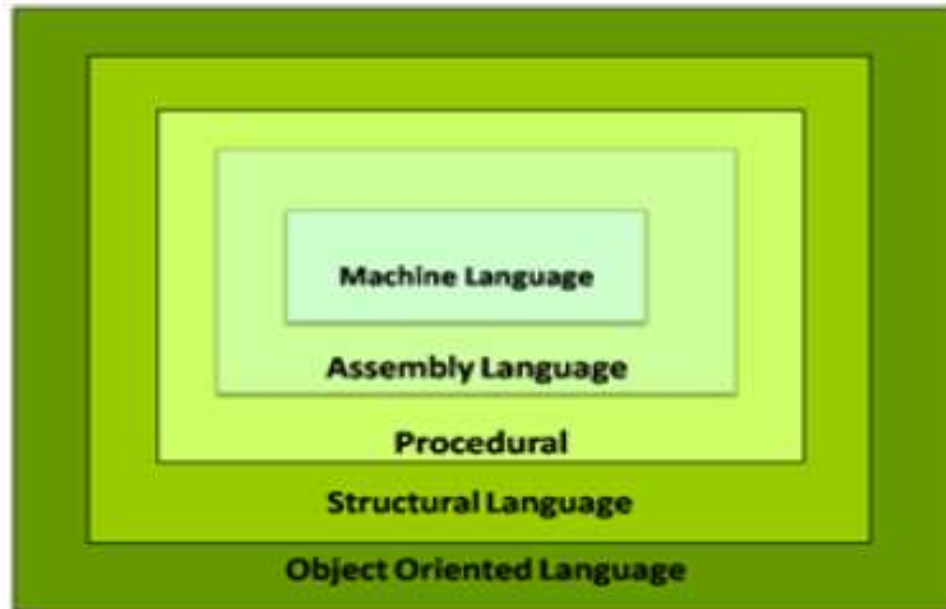
Java itu free, anda dapat mengunduh secara gratis versi standard edition (SE), menggunakan dan mengembangkannya tanpa dikenakan biaya.

Java memiliki dokumentasi yang luar biasa, Javadocs. Dokumentasi ini memberikan informasi yang sangat lengkap dan menyeluruh terkait dengan API Java.

Java tidak bergantung pada platform (platform-independent). Dengan Java Virtual Machine (JVM), Anda hanya perlu menulis program sekali dan dapat menjalankannya pada platform berbeda seperti sistem operasi Microsoft Windows, OS/2, Macintosh, UNIX dan IBM AS/400.

1.4 Paradigma Bahasa Pemrograman

Paradigma Bahasa pemrograman memberikan model untuk programmer dalam menulis listing program.



Gambar 3. Perbedaan Terstruktur dengan *Object Oriented*

Paradigma perbedaan dalam Bahasa pemrograman sebagai berikut:

1. Pemrograman tidak terstruktur atau **Programming Monolithic**

- Seluruh permasalahan ini diselesaikan sebagai blok tunggal.
- Semua data bersifat global dan tidak ada keamanan.
- Perintah melompat diperbolehkan jump dan banyak menggunakan perintah go to
- Cocok untuk permasalahan kecil.
- Sulit untuk melacak kesalahan program

Contoh Bahasa pemrograman yang termasuk dalam Programming Monolithic adalah: Assembly Language, BASIC.

2. Pemrograman Prosedural

- Masalah yang diberikan dibagi dalam beberapa sub masalah tergantung pada fungsinya.
- Masalah disebut prosedur atau Metode.

- c. Prosedur apapun dapat dipanggil pada setiap saat selama pelaksanaan program.
- d. Program ini memiliki variabel global dan lokal.

Fitur Pemrograman berorientasi Prosedur:

- a. Program besar yang terbagi dalam fungsi kecil atau Prosedur.
- b. Menggunakan Pendekatan pemrograman Top-Down.
- c. Data bergerak bebas dari satu fungsi ke yang lain.
- d. Sebagian besar fungsi berbagi data umum.
- e. Penekanan diberikan untuk algoritma

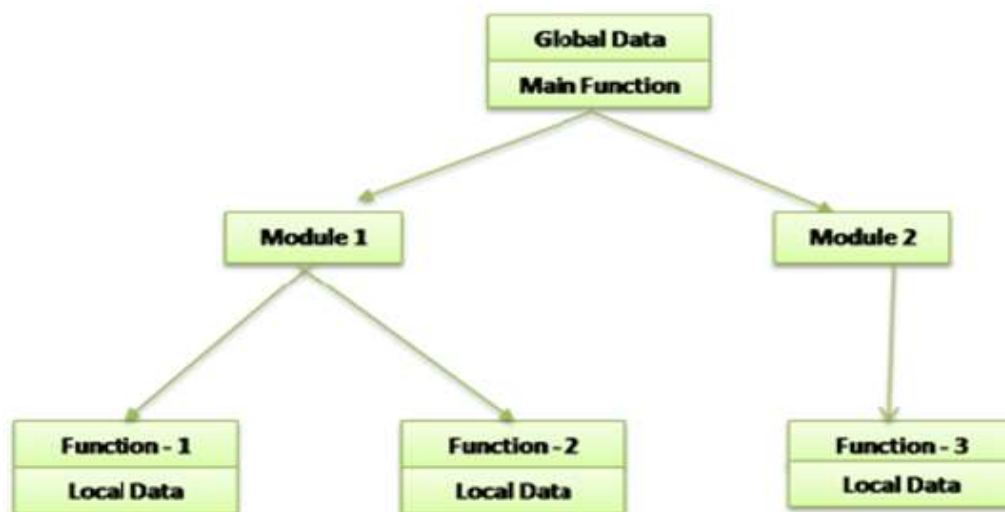
Kekurangan:

- a. Sangat sulit mengidentifikasi data yang digunakan oleh yang berfungsi.
- b. Sulit untuk melacak kesalahan program

3. Pemrograman Stuktural

- a. Kelebihan dan kekurangan jenis pemrograman ini yaitu:
- b. Program ini dibagi menjadi modul dan modul tersebut kemudian dibagi menjadi fungsi.
- c. Penggunaan Pernyataan go to dihapus atau dikurangi.
- d. Setiap modul dapat bekerja independen satu sama lain.

Berikut peng-gambarannya:



Gambar 4. Pemrograman Struktural

4. Pemrograman Berorientasi Objek

- a. Program ini dibagi menjadi jumlah unit kecil yang disebut Object. Data dan fungsi merupakan properti objek.
- b. Data dari objek hanya dapat diakses oleh fungsi yang terkait dengan objek tersebut.
- c. Fungsi satu objek dapat mengakses fungsi objek lain.

Fitur pemrograman berorientasi objek:

- a. Penekanan diberikan pada data daripada prosedur.
- b. Masalah dibagi menjadi obyek.
- c. Struktur data dirancang sedemikian rupa sehingga mereka mengatur objek.
- d. Data dan fungsi yang diikat bersama-sama.
- e. Penyembunyian data adalah mungkin.
- f. Data baru dan fungsi dapat dengan mudah dibuat.
- g. Obyek dapat berkomunikasi satu sama lain dengan menggunakan fungsi.
- h. Pendekatan bottom-up yang digunakan dalam membuat program

Tabel 1. Perbedaan Pemrograman Terstruktur dan Objek

Pemrograman Terstruktur	Pemrograman Berorientasi Obyek
<p>Pendekatan <i>top-down</i></p> <p>Fokus adalah pada algoritma dan kontrol aliran.</p> <p>Program dibagi menjadi beberapa sub modul atau fungsi atau prosedur.</p> <p>Fungsi yang independen satu sama lain.</p> <p>Tidak ada penerima yang ditunjuk dalam panggilan fungsi.</p> <p>Data dan fungsi sebagai dua entitas yang terpisah <i>Views</i>.</p> <p>Pemeliharaan mahal.</p>	<p>Pendekatan <i>bottom-up</i> yang diikuti.</p> <p>Fokus pada model obyek.</p> <p>Program ini diselenggarakan dengan memiliki sejumlah kelas dan objek.</p> <p>Setiap kelas berhubungan secara hirarkis.</p> <p>Ada penerima yang ditunjuk untuk setiap lewat pesan.</p> <p>Data dan fungsi sebagai satu kesatuan pandangan.</p> <p>Pemeliharaan relatif lebih murah.</p>
<p><i>Reuse</i> Software tidak mungkin.</p> <p>Fungsi panggilan digunakan.</p> <p>Fungsi abstraksi digunakan.</p> <p>Algoritma diberikan penting.</p> <p><i>Solution</i> adalah solusi spesifik-domain.</p> <p>Tidak ada enkapsulasi. Data dan fungsi yang terpisah</p>	<p>Membantu dalam penggunaan kembali perangkat lunak.</p> <p>Message passing digunakan.</p> <p>Data abstraction digunakan.</p> <p>Data diberikan penting.</p> <p><i>Solution</i> adalah spesifik masalah domain.</p> <p>Enkapsulasi paket kode dan data sama sekali. Data dan fungsi disatukan dalam satu kesatuan.</p>
<p>Hubungan antara programmer dan program ditekankan.</p> <p>Teknik data-driven digunakan.</p>	<p>Hubungan antara programmer dan pengguna ditekankan.</p> <p>Didorong oleh delegasi tanggung jawab.</p>

Dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Sedangkan untuk pemrograman terstruktur, menggunakan prosedur/tata cara yang teratur untuk mengoperasikan data struktur. Untuk tata nama, keduanya pun memiliki tatanan yang sama walaupun memiliki pengertian tersendiri.

Object oriented menggunakan "method" sedangkan terstruktur menggunakan "function". Bila di OOP sering didengar mengenai "objects" maka di terstruktur kita mengenalnya dengan "modules". Begitu pula halnya dengan "message" pada OO dan "argument" pada terstruktur. "Attribute" pada OO juga memiliki tatanan nama yang sepadan dengan "variabel" pada pemrograman terstruktur.

Pemrograman prosedural akan dikatakan lebih baik apabila dalam segala situasi melibatkan kompleksitas moderat atau yang memerlukan signifikan kemudahan maintainability. Manfaat yang dirasakan dalam penggunaan pemrograman prosedural adalah kemampuan kembali menggunakan kode yang sama tanpa menggunakan kode yang berbeda ataupun mengkopinya kembali. Dengan menggunakan "goto", memudahkan programmer melacak kumpulan data sehingga menghindarkan pemrograman terstruktur menjadi seperti spaghetti code.

Pemrograman berorientasikan objek dikatakan lebih baik apabila model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

1.5 Kebutuhan *Software* dan *Hardware*

Hardware:

1. Spesifikasi Minimum
2. Pentium 200-MHz
3. RAM 64 MB

Software:

1. Sistem Operasi Linux 7.1 or Windows xp/7/8
2. Java JDK 8
3. Microsoft Notepad atau text editor lainnya (JCreator).

1.6 Tata Cara Instalasi Java dan Editor

1. Instalasi JDK (Java Development Kit)

JDK atau Java Development Kit digunakan untuk mengkompilasi, debugging (memeriksa error), dan menjalankan program java.

Dalam JDK terdapat 2 hal yang perlu diketahui:

- a. *Compiler* yaitu yang bertugas untuk menerjemahkan kode program Java menjadi *bytecode*;
- b. Debugger yaitu yang bertugas untuk memeriksa error pada kode program Java.

Pada sistem operasi Windows, dapat menggunakan JDK dari Oracle sedangkan di Linux menggunakan openJDK.

Software JDK dapat di *download* atau diunduh pada situs <http://oracle.com> dengan versi manapun.

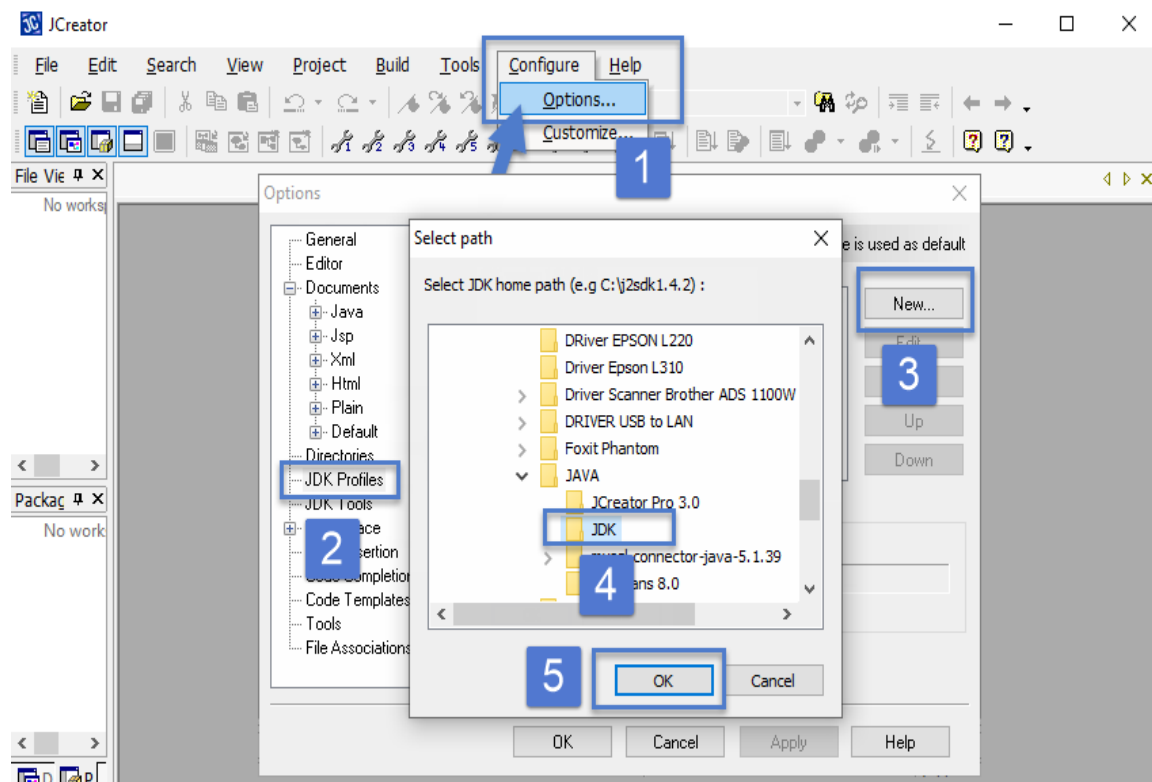
Contoh: **jdk-8u20-windows-i586**, kemudian lakukan dengan cara Double Click pada file tersebut, ikuti petunjuk dan tekan tombol next hingga selesai.

2. Editor JCreator

Download terlebih dahulu installer JCreator dengan versi JCreator Pro 3.0 atau JCreator 5.00 Pro, serta nomor serinya kemudian lakukan installasi dengan mengikuti petunjuk yang ada dan tekan tombol next hingga selesai.

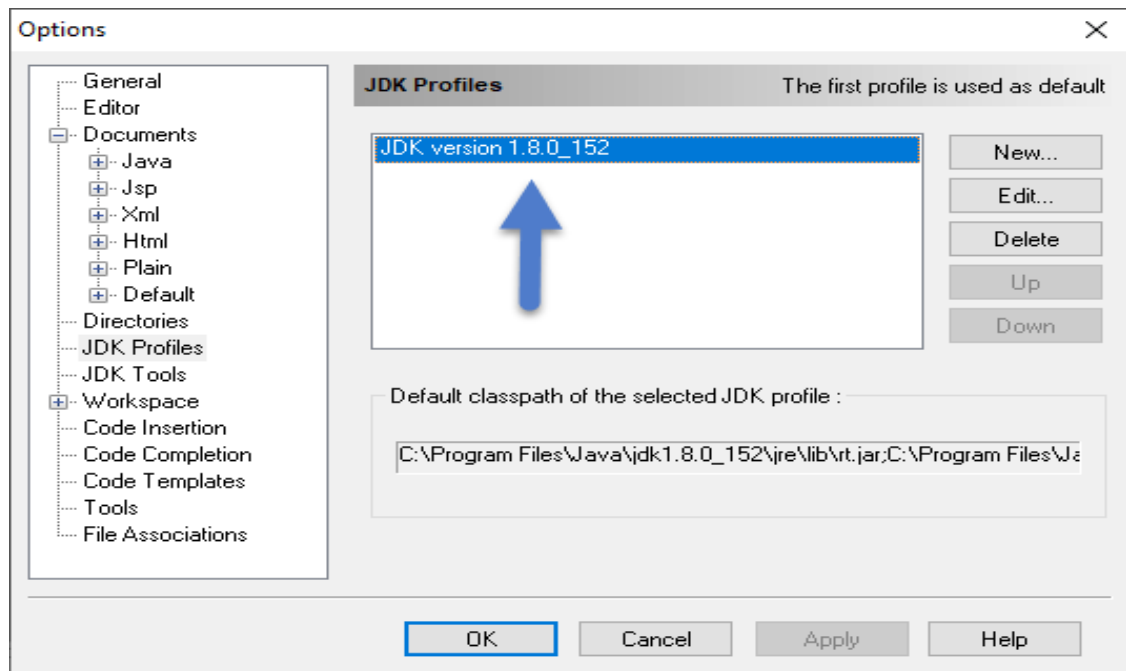
Setelah selesai kedua *software* terinstall, lakukan konfigurasi editor tersebut dengan Java, sehingga kedua *software* tersebut terhubung saat dilakukan kompilasi.

Dengan cara buka terlebih Aplikasi JCreator yang sudah terinstall, kemudian masuk kedalam menu → **Configure** dan Pilih → **Options**, maka akan tampil jendela Options kemudian masuk ke JDK Profiles. Jika JCreator baru terinstall maka harus menambahkan terlebih dahulu JDK melalui **Klik tombol New** dan akan tampil **Select path** yang berfungsi untuk mencari dimana letak file Installer JDK tersebut kemudian OK, yang terlihat pada gambar 3 berikut:



Gambar 5. Cara Konfigurasi Editor dengan Java

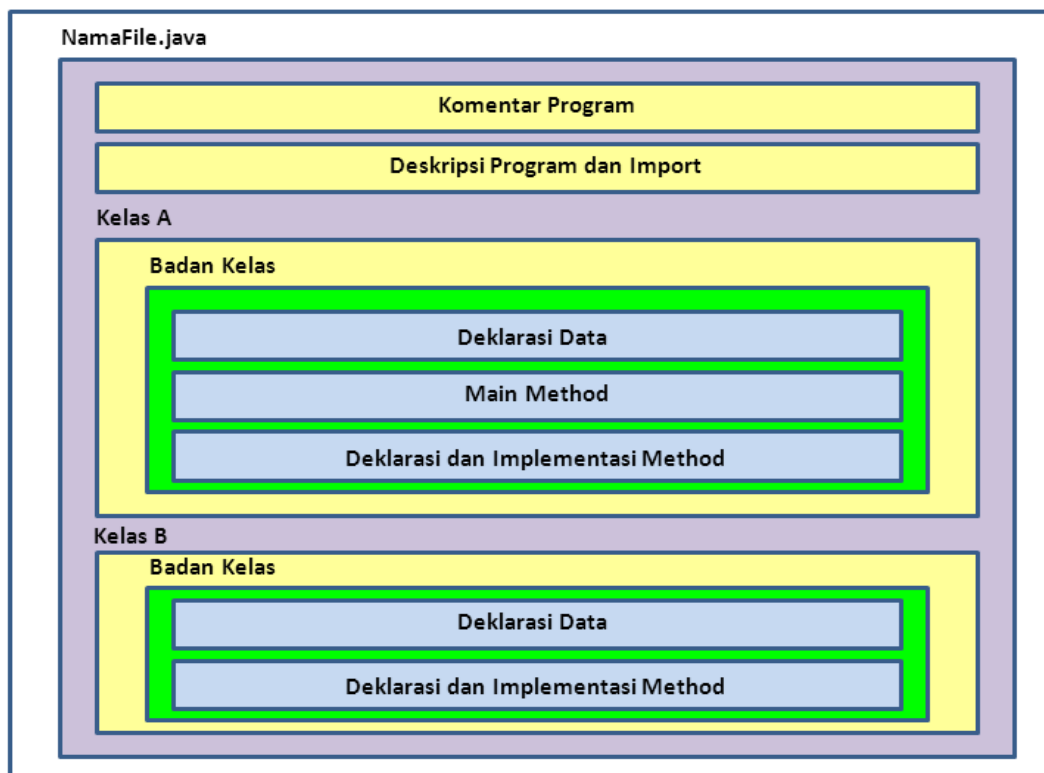
Tetapi jika sudah terinstall maka JDK nya akan tertera pada JDK Profile:



Gambar 6. JDK Sudah Terkonfigurasi

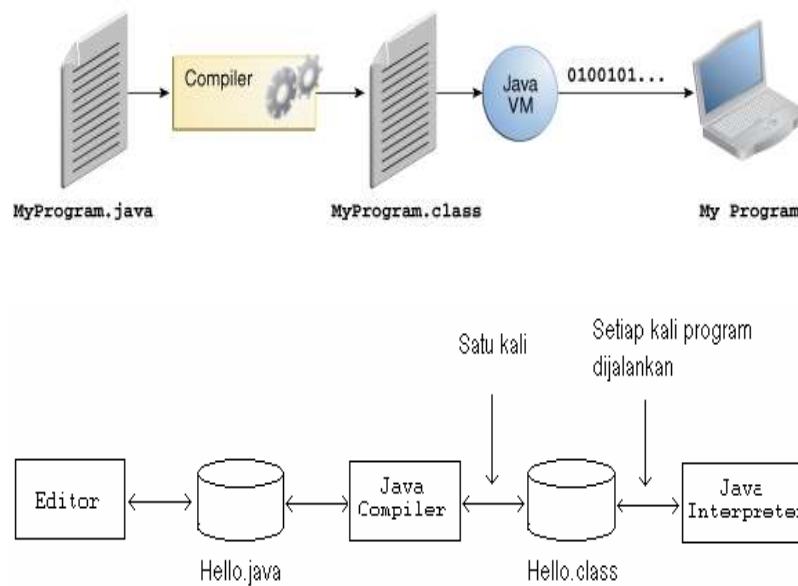
1.7 Kerangka Program

Berikut merupakan penggambaran kerangka program dalam Java:



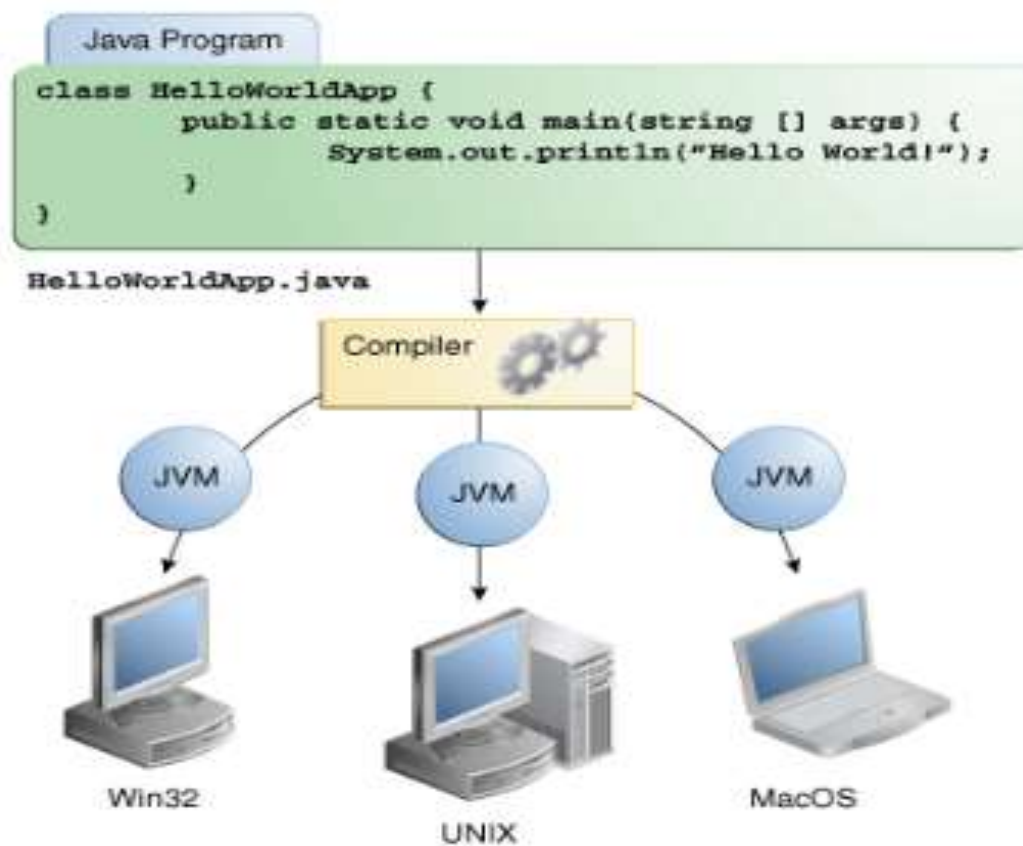
Gambar 7. Kerangka Program Java

1.8 Fase Pengembangan Program



Gambar 8. Fase Pengembangan Program


1.9 Contoh Program

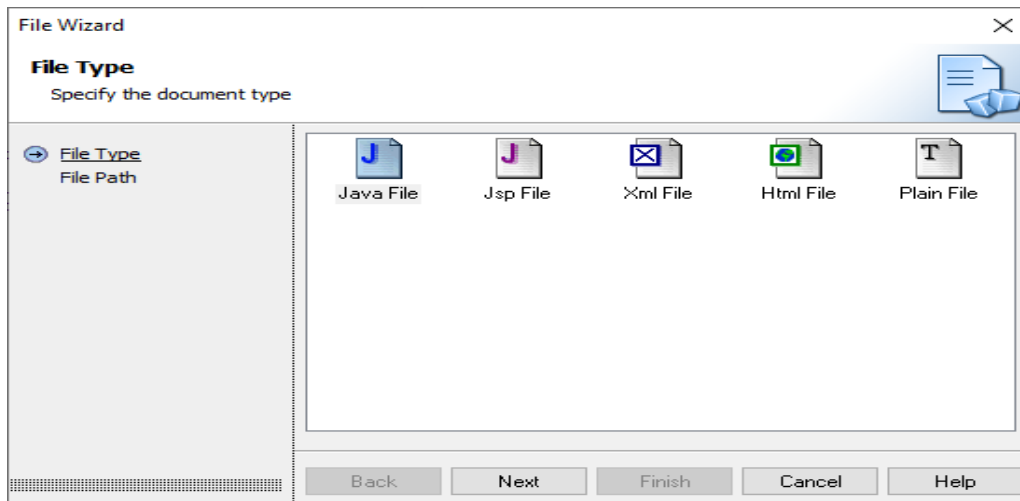


Gambar 9. Contoh Program Sederhana Java

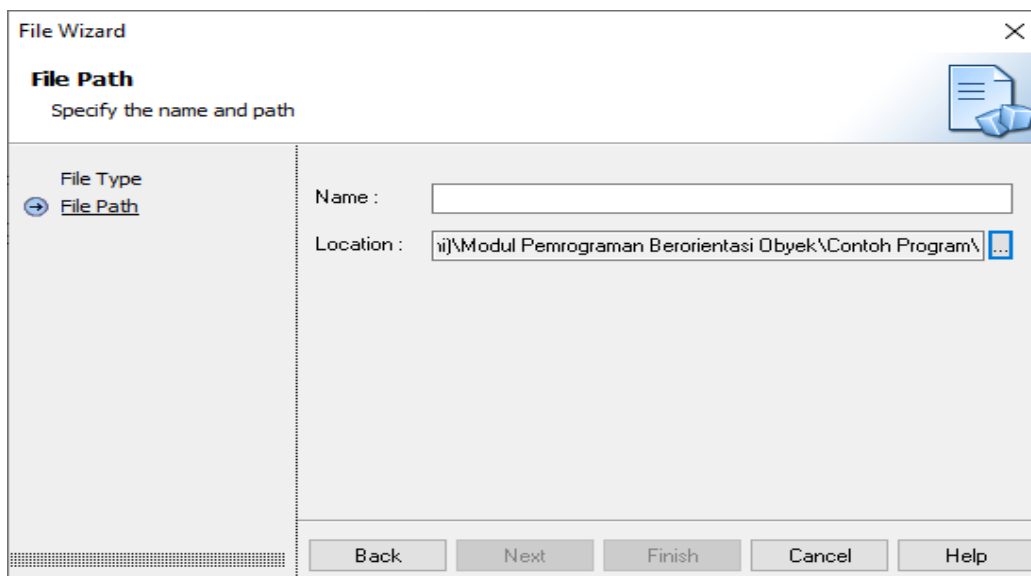
1.10 Latihan Mandiri: Praktikum

Langkah-langkah Praktikum

1. Buka Editor JCreator
2. Buatlah file baru dengan membuka menu File > New > File atau dengan Shortcut Ctrl + N atau dengan klik icon  pada ribbon menu, kemudian pilih > Java File dan Klik tombol Next



3. Buat Nama File sesuai dengan nama file Java masing-masing Contoh program, isikan pada bagian > Name, pilih lokasi tempat penyimpanan file pada pojok kanan Bagian Location kemudian Klik tombol Finish.



4. Lanjutkan dengan menuliskan program pada layar editor JCreator

Membuat Program untuk Mencetak 2 Baris Tulisan, seperti keluaran berikut:

Output Program :

Selamat Datang di Perkuliahan PBO
Di Universitas Budi Luhur

Program 1.1: tulisanCetak2Baris_01.java // di dalam method main //

```
1. public class tulisanCetak2Baris_01 {
2.     public static void main(String[] args) {
3.         System.out.println("Selamat Datang di Perkuliahan PBO");
4.         System.out.println("Di Universitas Budi Luhur");
5.     }
6. }
```

Program 1.2 : tulisanCetak2Baris_02.java // di dalam method main dg \n //

```
1. public class tulisanCetak2Baris_02 {
2.     public static void main(String[] args) {
3.         System.out.println("Selamat Datang di Perkuliahan PBO\nDi Universitas
        Budi Luhur");
4.     }
5. }
```

Program 1.3 : tulisanCetak2Baris_03.java // di dalam method konstruktor //

```
1. public class tulisanCetak2Baris_03 {
2.     public tulisanCetak2Baris_03() {
3.         System.out.println("Selamat Datang di Perkuliahan PBO");
4.         System.out.println("Di Universitas Budi Luhur");
5.     }
6.     public static void main(String[] args) {
        tulisanCetak2Baris_03 obj03 = new tulisanCetak2Baris_03();
7.     }
8. }
```

Program 1.4 : tulisanCetak2Baris_04.java // di dalam method cetak //

```
1. public class tulisanCetak2Baris_04 {
2.     public tulisanCetak2Baris_04() {
3.     }
4.     public void cetak() {
5.         System.out.println("Selamat Datang di Perkuliahan PBO");
6.         System.out.println("Di Universitas Budi Luhur");
7.     }
8.     public static void main(String[] args) {
9.         tulisanCetak2Baris_04 obj04 = new tulisanCetak2Baris_04();
```



```

10.      obj04.cetak();
11.  }
12. }

```

Program 5 : tulisanCetak2Baris_05.java // di dalam method cetak //

```

1.  public class tulisanCetak2Baris_05{
2.      public tulisanCetak2Baris_05(){
3.      }
4.      public void cetak(String a, String b){
5.          System.out.println(a);
6.          System.out.println(b);
7.      }
8.      public static void main(String[] args){
9.          tulisanCetak2Baris_05 obj05 = new tulisanCetak2Baris_05();
10.         obj04.cetak("Selamat Datang di Perkuliahan PBO", "Di Universitas Budi
        Luhur");
11.     }
12. }

```

1.11 Kebudiluhuran

<p>Sabar Mensyukuri</p> <p>Menahan atau menerima dengan ikhlas baik berupa kesenangan atau kesedihan dengan dasar bahwa segala sesuatu yang baik/buruk merupakan buah dari perbuatan yang telah dilakukan</p>	<p>Cinta Kasih</p> <p>Perasaan suka yang mendalam yang diwujudkan secara nyata dalam bentuk kepedulian terhadap sesama manusia, hewan, tumbuhan dan lingkungan atau alam sekitarnya.</p>	<p>Suka Menolong</p> <p>Suatu tindakan untuk meringankan beban atau penderitaan orang lain, baik materil maupun nonmateril yang menimbulkan rasa senang bagi orang yang ditolong maupun orang yang menolong</p>
<p>Jujur</p> <p>Sikap atau sifat seseorang yang menyatakan apa sesuatu sesungguhnya dan apa adanya, tidak ditambahi atau dikurangi sesuai dengan fakta dan obyektif sehingga dapat dipercaya semua ucapannya</p>	<p>Tanggung Jawab</p> <p>Merupakan kesadaran manusia akan tingkah laku atau perbuatan baik yang disengaja maupun yang tidak disengaja, sebagai perwujudan kesadaran akan kewajiban.</p>	<p>Rendah Hati</p> <p>Sifat pribadi seseorang yang dapat memposisikan sama antara dirinya dengan orang lain, merasa tidak lebih pintar, baik, mahir serta lebih tinggi atau mulia, juga dapat menghargai orang lain dengan tulus. Mencerminkan sifat yang berlawanan dengan kesombongan.</p>
<p>Toleransi</p> <p>Suatu sikap atau perilaku seseorang menerima pihak lain dan menghargai perbedaan atau tindakan orang lain. Melarang adanya diskriminasi terhadap kelompok-kelompok yang berbeda atau tidak dapat diterima oleh mayoritas dalam suatu masyarakat.</p>	<p>Kerjasama</p> <p>Melakukan kegiatan dengan orang lain, dengan cara menyatukan kekuatan untuk mencapai satu tujuan yang diinginkan bersama</p>	<p>Sopan Santun</p> <p>Sikap, perbuatan atau tingkah laku seseorang yang baik dalam pergaulan antar manusia yang beradab sesuai dengan tata krama atau adat istiadat setempat.</p>

Gambar 10. Kebudiluhuran

1.12 Kesimpulan

Pada Pertemuan ini dapat disimpulkan bahwa mahasiswa dapat menjelaskan hal-hal sbb:

1. Menjelaskan Sejarah, Platform, Keunggulan, emuan Program, Fase Pengembangan Program Aplikasi dengan Bahasa Pemrograman Java
2. Menjelaskan Kebutuhan Perangkat Keras Dan Perangkat Lunak Untuk Kebutuhan Pengembangan Aplikasi.
3. Menjelaskan Peran Kebudiluhuran dalam Implementasi Java di Dunia Nyata.