ALGORITMA DAN PEMROGRAMAN

dengan JAVA

Prof. Dr. Widodo Budiharto

2019

### Bab 1

### Jawa untuk Pengembangan Aplikasi Bisnis

### Tujuan Instruksional Umum

* Setelah memelajari Bab ini, mahasiswa mampu menjelaskan fitur-fitur serta versi yang ada pada bahasa pemrograman Java.

**Tujuan Instruksional Khusus:**

* Mahasiswa mampu menjelaskan sejarah bahasa pemrograman Java
* Mahasiswa mampu menjelaskan fitur-fitur yang ada pada bahasa Java
* Mahasiswa dapat membuat dan menjelaskan struktur program Aplikasi Java
* Mahasiswa dapat menjelaskan pentingnya pengembangan aplikasi bisnis dengan Java
* Mahasiswa dapat menjelaskan Java J2EE

**Sekilas Bahasa Pemrograman Java**

1. **Bahasa Jawa untuk Aplikasi Bisnis**

Aplikasi bisnis merupakan aplikasi yang dibuat untuk memudahkan pekerjaan manusia, misalnya di industri retail, Sistem Informasi dan *e-Commerce*. Pengembangan aplikasi bisnis *(Bussiness Application Development*) yang handal tentunya membutuhkan peranti lunak/*software* yang handal. Salah satu *software* yang dikenal dan dipercaya para pengembang hingga saat ini adalah Java. Java dipilih karena dapat diunduh secara free, handal terhadap virus, tersedia kelas-kelas yang memudahkan di dalam pengembangan aplikasi database berbasis weh dan enterprise serta mendukung pengembangan aplikasi *mobile* yang tidak bergantung pada platform sistem operasi dan mikroprosesor tertentu.

Pada tahun 1995, Sun Microsystem meluncurkan peranti lunak pemrograman bernama Java versi 1.0, sebuah bahasa yang mendukung penuh pemrograman berorientasi objek (OOP) untuk membangun aplikasi berbasis windows maupun web yang robust. Kehadiran bahasa tersebut diawali oleh pembuatnya bernama James Gosling, Mike Sheridan dan Patrick Naughton pada Juni 1991. Diberi nama Java, karena pembuatnya terinspirasi dari nikmatnya kopi Java yang membuat peminumnya merasa lebih dinamis. Bahasa Java ini meskipun mempumyai konsep pemrograman yang mirip dengan bahasa pendahulunya, tetapi mempunyai fitur-fitur yang lebih menjanjikan, seperti penggabungan fitur memori manajemen dan garbage collection dari Smalltalk dan sintak dari C/C++ menjadi suatu bahasa java yang mudah untuk dipelajari. Jika anda ingin menjadi developer handal dan profesional, maka penguasaan pemrograman Java terutama J2EE merupakan pilihan yang tepat.

1. **Versi Bahasa Pemrograman Java**

Terdapat 4 platform bahasa Java, yaitu:

* **Java Standard Edition (Java SE)**

Java SE API menyediakan fungsionalitas inti dari bahasa pemrograman Java. Ia mendefinisikan segala sesuatu dari bahasa pemrograman Java tingkat tinggi yang digunakan untuk aplikasi jaringan komputer, akses keamanan, database, antarmuka pengembangan pengguna grafis (GUI), dan *parsing* XML. Selain API inti, platform Java SE terdiri dari virtual machine, perangkat pengembangan, teknologi penyebaran, dan library lain dan toolkit umum yang digunakan dalam aplikasi teknologi Java.

* **Java Platform, Enterprise Edition (Java EE)**

Platform Java EE dibangun di atas platform Java SE. Platform Java EE menyediakan API dan lingkungan runtime untuk mengembangkan dan menjalankan aplikasi skala besar, multi-tier, scalable, aplikasi jaringan yang handal, dan aman.

* **Java Platform, Micro Edition (Java ME)**

Platform Java ME menyediakan API dan small-footprint virtual machine untuk menjalankan aplikasi bahasa pemrograman Java pada perangkat kecil, seperti ponsel. Aplikasi Java ME sering merupakan klien dari layanan platform Java EE.

* **Java FX**

Java FX adalah platform untuk membuat aplikasi internet yang kaya menggunakan API user-interface yang ringan. Aplikasi Java FX menggunakan *hardware-accelerated* grafis dan mesin media untuk mengambil keuntungan dari kinerja klien yang lebih tinggi dan modern serta API tingkat tinggi (*high-level APIs*) untuk menghubungkan ke sumber jaringan data. Aplikasi Java FX biasanya merupakan klien dari layanan platform Java EE.

1. **Kompilasi program di Java**

Ketika sebuah program Java dikompilasi, kode program tersebut menjadi apa yang disebut sebagai *java bytecode*, ia akan mentranslasi ke kode mesin atau instruksi prosesor yang spesifik pada prosesor. Pada Lingkungan pengembangan java, ada 2 bagian yaitu *java compiler* dan *java interpreter*. *Compiler* membangkitkan *bytecode*, disamping itu ada kode mesin dan interpreter yang mengeksekusi program Java. Kekurangan menggunakan bytecode ialah kecepatan eksekusi. Untuk menulis sebuah program Java, dibutuhkan editor, compiler Java dan Java Runtime Environment.

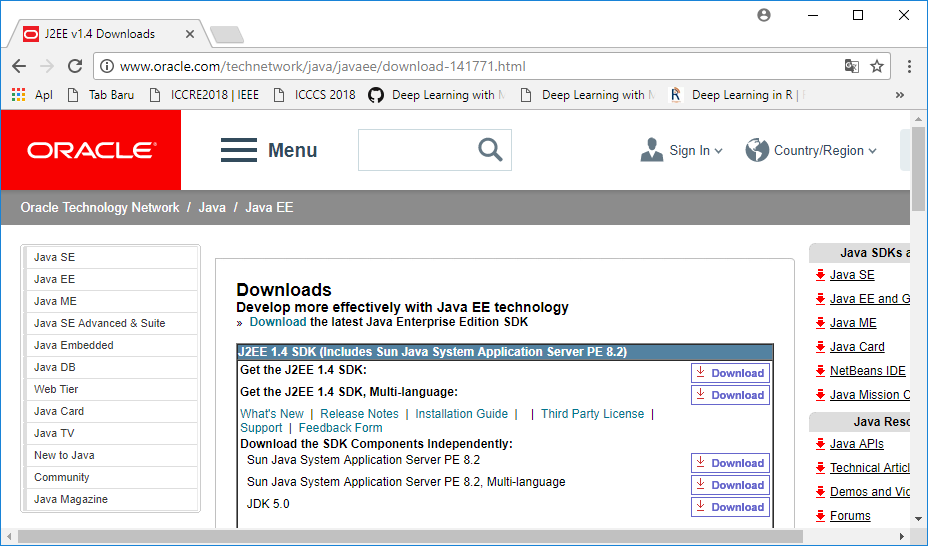
Pertimbangan mengapa Java banyak digunakan yaitu tidak hanya sekedar bahasa pemrograman, tetapi juga merupakan platform. Banyak bahasa program yang tidak memiliki fitur seperti multi-threading, GUI-building, dan kemampuan dalam networking. Dimana anda harus menggunakan perangkat keras atau operating-system dependent API (Application Program Interface) yang akan mengikat (binding) program. Sedangkan Java platform merupakan tekhnologi pertama yang berintegrasi penuh untuk dapat menggunakan thread, socket, GUI *component* dan banyak lagi.

Sewaktu Anda menginstall java, secara otomatis java akan me*load* java platform ke dalam sistem Anda. Java platform tersebut termasuk **Java Virtual Machine (JVM)** dan termasuk juga *package* yang akan mendukung compiler, interpreter dan *run-time environment* pada saat Java program di kompile maupun pada saat running. Package bisa dianalogikan sebagai standard library dari function atau class dalam bahasa program lain.

Pada 1998, Sun mengupdate spesifikasi Java dan memperkenalkan Java 1.2 beserta library terbaru yang membuat java bukan hanya sebuah bahasa tetapi juga menjadikan ia sebuah platform bernama Java 2 Standard Edition (J2SE). Dengan J2SE, Sun berusaha memperbaiki kekurangannya dengan membundle berbagai macam library menjadi 1 unit. J2SE menyediakan library untuk dukungan GUI, jaringan, akses database dan lainnya. J2SE hanya mampu membangun aplikasi stand alone, namun tidak dapat membangun dan mendistribusikan aplikasi berskala enterprise /perusahaan. Gambar di bawah menampilkan platform Java SE (JDK) 9 dan J2EE terbaru yang dapat diunduh secara gratis pada situs <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

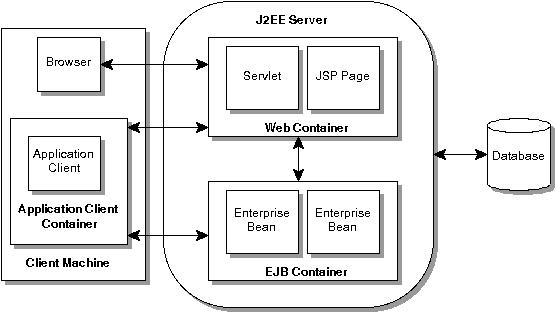


***Gambar 1.1 Tampilan Web pengunduhan Java***



***Gambar 1.2 Tampilan Web pengunduhan Jawa J2EE 1.4 SDK***

J2EE diperkenalkan pada tahun 1998, yang mendefinisikan sebuah arsitektur multi tier untuk *Enterprise Informasion System (EIS*). Platform J2EE menentukan komponen aplikasi logis di dalam sebuah system dan mendefinisikan aturan pada proses pengembangan. Gambar di bawah ini menampilkan arsitektur dari J2EE dimana Business logic dalam model *reusable component* sehingga lebih mudah di dalam membuat aplikasi. Secara umum web server menyimpan dokumen web yang dapat kita akses melalui browser di komputer kita, database server dan server aplikasi *(application server)* seperti gambar 1.3 di bawah:



***Gambar 1.3 Arsitektur J2EE standar***

Program Java dapat dijalankan pada berbagai tipe komputer dan Sistem Operasi dari program kecil sampai program berskala besar yang berjalan di superkomputer. Program di download lewat jaringan dengan HTML dan dijalankan oleh Web Browser IE, Modzilla dan lain-lain. Lingkungan yang berbeda juga membutuhkan program yang berbeda pula. Ada 4 macam program Java yang bisa dibuat:

* **Applications**, merupakan program yang dapat berjalan di *standalone* komputer, dari mulai program besar dan rumit sampai dengan program simple. Bisa berjalan lewat comand line atau GUI. Juga apakah aplikasi menggunakan interface untuk koneksi ke jaringan, database atau aplikasi lain, tergantung dari set instruksi API yang digunakannya. Jika aplikasi yang dibuat mempunyai tampilan grafis windows yang menarik, disebut sebagai **Java swing** karena menggunakan paket swing**.**
* **Applet,** merupakan program java yang bisa dijalankan lewat Web browser dengan menggunakan bahasa HTML. Sewaktu HTML menjalankan class yang terdapat Java Applet, maka Web browser akan menggunakan JVM untuk menjalankan applet tersebut.
* **Servlet,** seperti applet, merupakan program yang men-*generate* isi dari Web page. Tetapi tidak seperti applet, servlet berjalan dalam Java-enabled Web server dan men-generate dokumen HTML yang kemudian dikirimkan ke browser client untuk ditampilkan. Contoh sewaktu anda mengisikan form isian yang dibuat dengan Java.
* **JSP (Jawa Server Pages)**, aplikasi ini ialah aplikasi web yang umumnya diproses penampilannya di client, umumnya memanggil suatu kelas yang disebut bean yang diproses di web server.



Aplikasi yang berjalan di standalone PC



Applet, servlet dan JSP berjalan lewat jaringan dan Web Browser

***Gambar 1.4 Berbagai macam tipe aplikasi java***

### Keunggulan Java

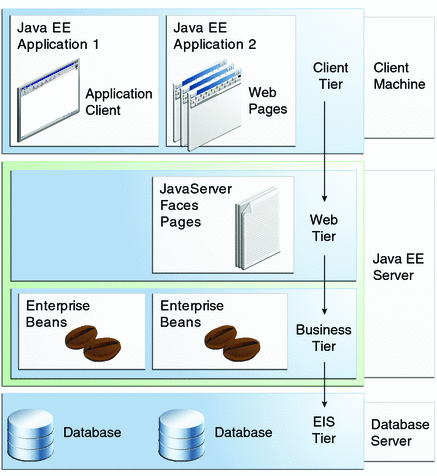
Dengan menggunakan Java programming language, Sun Microsystem membangun sebuah bahasa program yang bisa dibilang powerfull untuk kita gunakan, karena mempunyai keunggulan-keunggulan sebagai berikut:

* **Simple.** Dengan tersedia banyak kelas, maka membuat aplikasi berbasis Java menjadi lebih simple/mudah.
* **Secure**. Menggunakan Browser yang kompatibel dengan java, seseorang dapat dengan mudah mendownload applet java tanpa harus takut virus atau *malicious intent*. Java menggunakan proteksi ini dengan model java execution environment.
* **Object Oriented**. Pemrograman java mendukung pemrograman berorientasi object OOP (mendukung inheritance, polymorphism dan encapsulation).
* **Interpreted.** Java platform compiler menterjemahkan Java source code menjadi bytecode. Java interpreter akan memproses bytecode, bukan source code sewaktu program sedang running.
* **Distributed.** Java sangat powerfull untuk membangun aplikasi terdistribusi seperti cluster dan Grid Computing.
* **Platform Independent,** Program java yang Anda jalankan di Windows dapat anda jalankan /kompile di sistem operasi linux atau hardware yang berbeda.
* **Java sangat Powerful.** Para designer Java program membuat bahasa program Java dan API-nya untuk bisa dengan mudah dipelajari dan digunakan. Dibandingkan dengan bahasa yang lain seperti C dan C++. Sintax dari bahasa Java berdasarkan pada sintaks dari bahasa C.
* **Network-savvy.** Yang merupakan keunggulan lain dari java yaitu fitur penyedia layanan untuk networking dan mendistribusikan objek dalam suatu environment. Package yang disebut *java.net* yang membantu dalam pemrograman jaringan. Dengan demikian jika kita ingin membuat program yang menggunakan network akan lebih mudah.

**J2EE** menggunakan model aplikasi multi tier terdistribusi untuk aplikasi Perusahaan. *Application logic* dibagi menjadi komponen-komponen sesuai dengan fungsinya, dan komponen-komponen aplikasi lainnya yang membuat J2EE terinstall di komputer yang berbeda bergantung pada tier di lingkungan J2EE . Gambar 1.4menampilkan *two multitiered J2EE applications* yang terbagi menjadi

* Komponen client-tier yang berjalan di komputer klien.
* Komponen Web-tier yang berjalan di J2EE server.
* Komponen Business-tier yang berjalan di J2EE server.
* *Enterprise information system* (EIS)-tier software yang berjalan di EIS server.

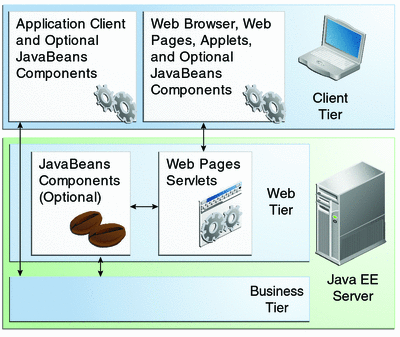
Meskipun aplikasi J2EE dapat terdiri dari 3 atau 4 tier, aplikasi multitier J2EE secara umum dianggap sebagai 3 tier karena mereka terdistribusi melalui 3 lokasi yang berbeda yaitu komputer klien, computer J2EE Server dan komputer database atau legacy di backend.



***Gambar 1.5 Aplikasi multi tier berbasis Java***

**Komponen –Komponen Web**

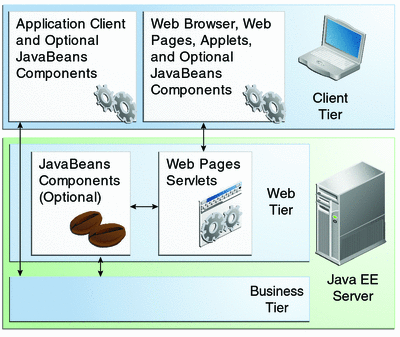
Komponen Web J2EE dapat berupa servlet atau JSP. Servlet ialah kelas–kelas bahasa pemrograman java yang secara dinamis memproses permintaan *(request)* dan membangkitkan respon. JSP ialah dokumen berbasis teks yang dieksekusi sebagai servlet tetapi mengizinkan pendekatan lebih alami untuk membuat isi statis.



**Gambar 1.6 J2EE Server dan Container**

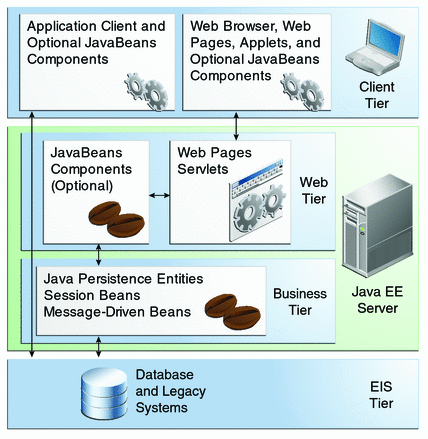
# Komponen Bisnis

Kode bisnis, dimana logic yang mengatasi atau menangani kebutuhan domain bisnis teretentu seperti perbankan, retail atau keuangan, ditangai oleh Enterprise Beans yang berjalan di tier bisnis. Gambar di bawah menampilkan bagaimana enterprise bean menerima data dari klien, memproses nya dan mengirim ke sistem informasi enterprise untuk media penyimpan.Enterprise bean juga mengambil dari dari media penyimpan, memprosesnya dan mengirim ke program klien.



***Gambar 1.7 Web Tier dan aplikasi J2EE***

Sedangkan gambar di bawah ini menampilkan tier business dan EIS pada platform J2EE



**Gambar 1.8 Tier Business dan EIS**

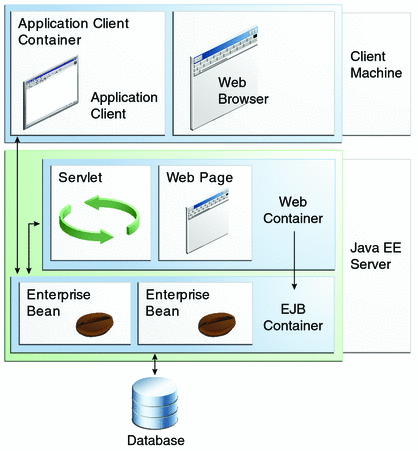
Ada 3 jenis Enterprise Java Bean yaitu *session bean*, *entity bean* dan *message driven bean* yang dapat Anda pelajari pada materi EJB.

# Kontainer J2EE

Umumnya, aplikasi multitier thin client cukup sulit karena melibatkan banyak intrik kode untuk menangani manajemen transaksi dan state *(state management),* pooling sumber daya dan lainnya. Arsitektur J2EE yang platform independent berbasiskan komponen membuat aplikasi J2EE mudah untuk dibuat karena logic bisnis diorganisasikan ke dalam komponen yang dapat digunakan ulang *(reusable)* . J2EE server juga menyediakan layanan dalam bentuk sebuah kontainer untuksetiap tipe. Karena Anda tidak harus membangun service ini sendiri, sehingga Anda hanya perlu berkonsentrasi mengatasi masalah bisnis/pemrograman Anda saja.

# Layanan Kontainer

Container ialah antarmuka diantara sebuah komponen dan platform low level spesifik yang mendukung komponen. Sebelum sebuah web, enterpise bean atau komponen klien dapat dieksekusi, ia harus dirakit ke aplikasi J2EE dan disebarkan ke kontainernya. Proses perakitan melibatkan seting kontainer untuk tiap komponen di aplikasi J2EE itu sendiri. Seting kontainer akan menghasilkan J2EE security model yang memungkinkan adan mengkonfigurasi sebuah komponen web atau enterprise bean sehingga sumber daya sistem diakses hanya oleh user yang punya authorisasi*.*Kontainer juga mengatur layanan yang tidak dapat dikonfigurasi seperti enterprise bean dan daur hidup servlet, koneksi database, data persistence dan akses ke platform J2EE API*(Application Programming Interface*).  Proses penyebaran *(deployment)* menginstall komponen-komponen aplikasi J2EE ke container J2EE seperti gambar berikut.



***Gambar 1.9 J2EE Server dan kontainer***

1. **Server J2EE**

Bagian runtime dari sebuah produk J2EE. J2EE server menyediakan kontainer Web dan EJB(Enterpise JavaBeans)

###### Enterprise JavaBeans (EJB) container

Mengatur eksekusi dari enterprise bean untuk aplikasi j2ee. Enterpise bean dan kontainernya berjalan di server j2ee.

###### 

###### Web container

Mengatur eksekusi dari komponen JSP dan servlet untuk aplikasi j2ee.

# Application client container

Mengatur eksekusi komponen aplikasi klien .Aplikasi klien dan kontainernya berjalan di klien.

# Applet container

Mengatur eksekusi dari applet. Terdiri dari sebuah web broser dan plug in java yang berjalan di klien secara bersama.

**Pemaketan Aplikasi**

Aplikasi J2EE dalam bentuk file ***Enterprise Archive*** (EAR) . File EAR ialah file Java Archive (JAR) standar dengan ekstensi.ear . File EAR berisi module J2EE. Menggunakan file EAR dan modul memungkinkan untuk mreakit berbagai aplikasi j2ee yang berbeda menggunakan komponen yang sama.

ModulJ2EE terdiri dari 1 atau lebih komponen J2EE untuk kontainer yang sama dan 1 komponen deployment descriptor .Sebuah descriptor penyebaran ialah berupa dokumen XML dengan ekstensi*.xml* yang menjelaskan sebuah seting penyebaran.

4 tipe dari modul ialah :

* Modul Enterprise Javabeans berisi file-file clas untuk enterprise beans dan EJC Deployment descriptor, yang dipaket dalam file JAR dengan ekstention .jar.
* Modul web berisi file JSP, file-file class untuk servlet, file GIF dan html dan sebuah web deployment descriptor yang dipaket sebagai file JAR dengan ekstention .war(web archive).
* Modul adapter resource berisi seluruh interface java, kelas, native libraries, resource descriptor dan dokumen lainnya dengan ekstetion .rar(Resource adapter archive)
* Modul klien aplikasi berisi file kelas dan client deployment descriptor yang dipaket sebagai file .jar.

### Layanan Standar J2EE

J2EE 7 mempunyai layanan standar yang lebih lengkap yaitu :

* *HyperText transfer protocol/HyperText Transfer Protocol Secure Sockets* ***(HTTP/HTTPS)*** yang didukung oleh Server J2EE
* ***Java Transaction API (JTA) 1.3***, yang menyediakan antarmuka transaksi untuk menempelkan sistem pemrosesan transaksi
* Remote Method Invocation to *Internet Inter-ORB Protocol* (RMI-IIOP), komponen EJB menggunakan layanan ini untuk komunikasi.
* **Java Database Connectivity (JDBC)**, JDBC menyediakan antarmuka untuk mengeksekusi pernyataan SQL .
* ***Java Message Service (JMS)***  sebuah layanan pesan asinkron yang memungkinkan user mengirim atau menerima pesan
* ***JavaMail***, memungkinkan pengiriman dan penerimaan email melalui transportasi pesan dan penyimpan pesan.
* ***Java Naming and Directory Interface (JNDI)***, digunakanuntuk mengakses direktori seperti LightWeight Direcotry Access Protocol (LDAP).
* ***JavaBeans Activation Framework (JAF)****,* Javamail menggunakan JAF untuk menangani berbagai macam*Multipurpose Internet Mail Extentions* (MIME) yang mungkin beradea di dalam pesan email.
* ***Web Services***, mendukung web services yang disediakan mealui *Simple Object Access Protocol (SOAP) for attachment, API for Java* (SAAJ) 1.1 untuk pesan SOAP, *Java API for XML registries* (JAXR) 1.0 untuk akses ke *Universal Description, Discovery and Integration* (UDDI) dan Java API untuk XML based RPC (JAX-RPC) 1.0 untuk menentukan bagaimana klien dapat menggunakan Web Services.
* Java API for RESTful Web Services (JAX-RS)
* Managed Beans
* Contexts and Dependency Injection for the Java EE Platform (JSR 299), informally known as CDI
* Dependency Injection for Java (JSR 330)
* Bean Validation (JSR 303)
* Java Authentication Service Provider Interface for Containers (JASPIC)

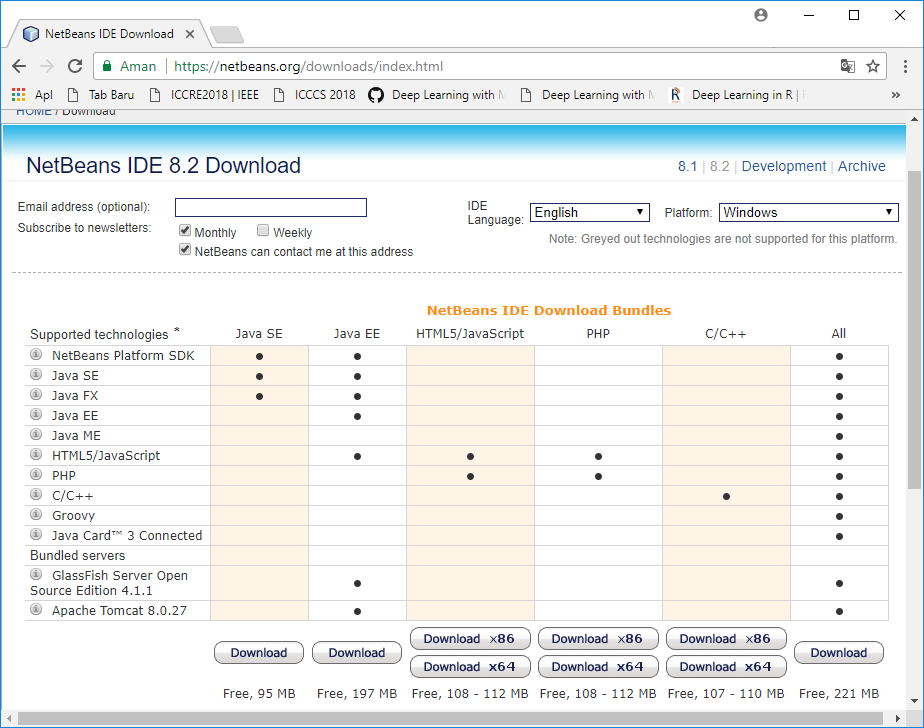
## Mengompilasi Program Java, JSP dan Servlet

Untuk memelajari Java melalui buku ini, Anda harus menginstal program java versi J2SE JDK7, J2EE 6 dan web server Apache Tomcat sebagai web server percobaan yang sudah terinstal pada saat instalasi editor Java NetBeans. J2SE version 8 digunakan untuk membangun aplikasi level mudah dan menengah, misalnya untuk membuat applet, program console sederhana, algoritma, dokumen web dan lain lain. Sedangkan J2EE untuk membangun aplikasi Web berskala besar / Enterprise menggunakan Servlet dan JSP, beserta fitur-fitur advanced lainnya seperti Enterprise Jawa Beans (EJB).

**Instalasi Java J2SE, J2EE dan NetBeeans 8.2**

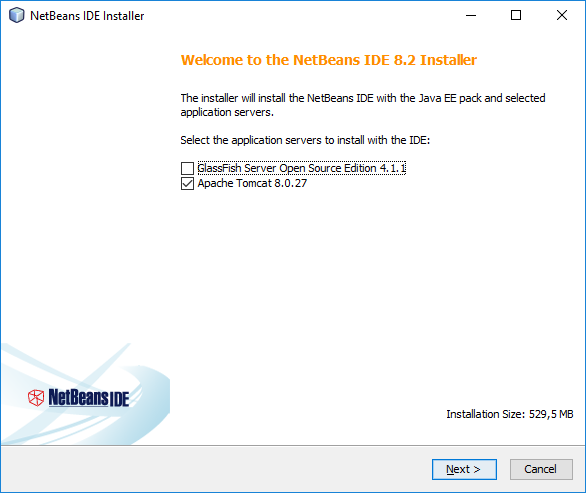
Berikut langkah menginstaslasi Program Java dan NetBeans sebagai editor java dan membuat program java sederhana dan cara menjalankannya di windows, untuk di linux Anda ikuti petunjuk konfigurasi program yang tidak jauh beda.

1. Unduhlah NetBeans IDE 8.2 di <https://netbeans.org/downloads/index.html> yang mendukung J2SE dan J2EE sebesar 197M



***Gambar 1.10 Mengunduh program NetBeans***

1. Unduh dan install J2SE version 9 di **C:\J2SE**. Lalu jalankan program instalasi NetBeans, buat folder instalasi misal **C:\NetBeans** lalu klik next hingga finish

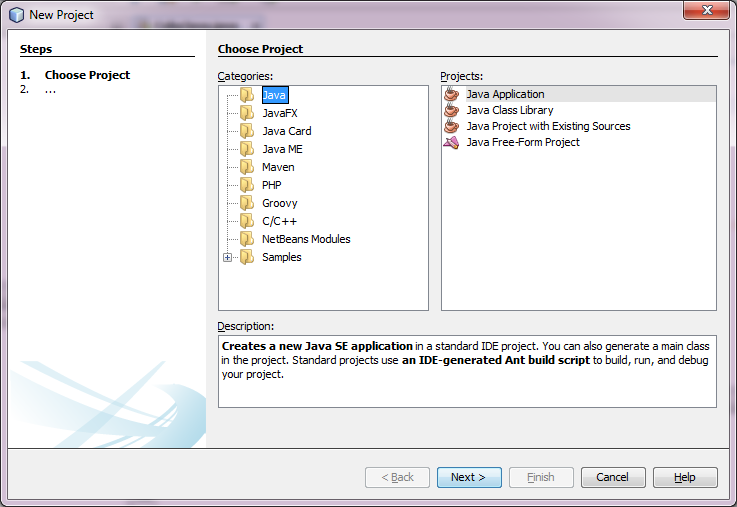


***Gambar 1.11 Instalasi Program NetBeans***

**Membuat program Java Application**

Jika Anda masih awam dengan Java, langkah paling tepat ialah membuat program aplikasi Java sederhana. Berikut langkah-langkahnya :

1. Buka NetBeans, lalu pilih menu **File | NewProject** serta pilih Projects **Java Application**, lalu klik next



***Gambar 1.12 Memilih proyek Java Application***

1. Beri nama program SelamatDatang (SelamatDatang.java) dengan Project Name J2SE, sehingga hasilnya berada di D:\selamatdatang\SelamatDatang.java. Oleh karena itu pada program SelamatDatang.java, tercipta package selamatdatang. Modifikasi kode yang tercipta sebagai berikut dengan menambahkan kode yang dicetak lebih tebal, kode tesebut akan menampilkan pesan **“Selamat Belajar Panduan Lengkap Pengembangan Aplikasi Bisnis dari Prof. Widodo”.**

/\*

\* SelamatDatang.java

\*

\* Created on 1 January, 2019

\*/

package selamatdatang; //karena berada di dalam folder selamatdatang

public class SelamatDatang {

public static void main(String[] args)

{

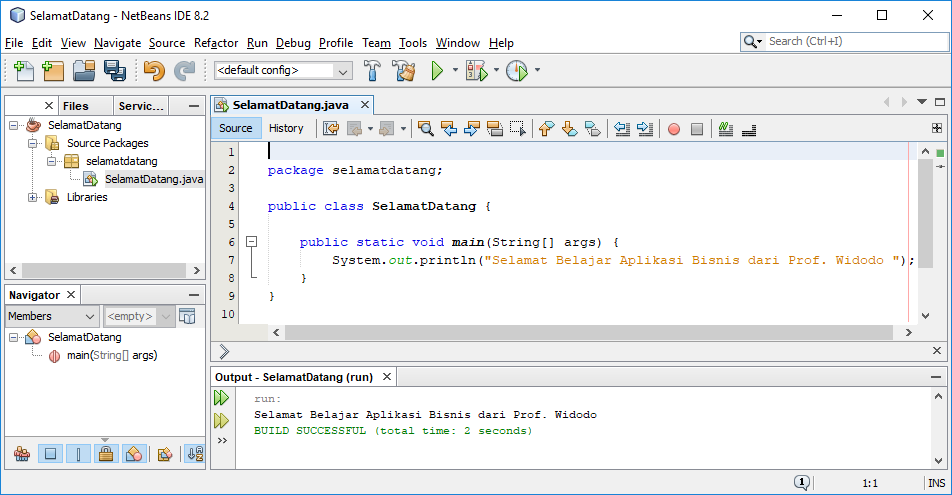
**System.out.println ("Selamat Belajar Aplikasi Bisnis dari Prof. Widodo ");**

}

}

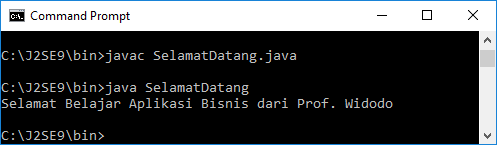
Program diatas mempunyai sebuah kelas bernama SelamatDatang, dimana pada method utama main() akan menampilkan pesan **“Selamat Belajar Aplikasi Bisnis dari Prof. Widodo”.** Method main() ialah fungsi yang pertama kali dipanggil setiap proram dijalankan. Karena fungsi main dinginkan tidak mengembalikan harga, maka digunakan tipe void, static agar dapat dipanggil langsung tanpa perlu membuat objek serta bersifat public agar dapat diakses/digunakan oleh semua kelas yang membutuhkan.Fungsi main juga membutuhkan 1 buah array bernama args bertipe data String.Perlu diketahui, di java String adalah suatu kelas, yang mempunyai kelebihan dibandingkan pointer atau tipe data karakter di C/C++. Simbol kurung [] yang menandakan args adalah suatu array boleh diletakkan di depan atau dibelakang args.

Perlu diketahui juga, bahwa untuk kelas, setiap awal karakter dari setiap kata harus huruf besar. Program di atas membutuhkan kelas System dengan method println() untuk menampilkan pesan ke monitor dan pastikan setiap akhir baris kode diberikan titik koma (;). Setiap Anda membuat kurung buka ( { ) pastikan ada segera tutup dengan kurung tutup ( } ) untuk menghindari kesalahan penulisan kurung jika program cukup kompleks.



***Gambar 1.18 Program yang sukses dijalankan***

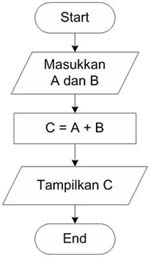
1. Anda sebenarnya juga dapat menulis program Java di editor standar seperti Notepad dan melakukan kompilasi program .java melalui DOS prompt dengan cara memberikan perintah **javac SelamatDatang.java** pada console untuk mengkompilasi menjadi **SelamatDatang.class.** Jalankan dengan memberikan perintah **java SelamatDatang** (tidak boleh ditulis nama kelasnya), tampilan berikut jika program tersimpan di folder **c:\j2se9\bin**. Perlu diperhatikan baris program package selamatdatang dari contoh kode di atas harus dihapus dalam hal ini.



##### Gambar 1.19 Tampilan hasil kompilasi program

**Algoritma**

Algoritma adalah urutan langkah-langkah logis penyeselaian masalah yang disusun secara sistematis dan logis yang dapat digambarkan menggunakan flowchart atau pseudocode. Pseudocode adalah cara penulisan algoritma yang hampir menyerupai Bahasa Pemrograman, namun Pseudocode ditulis lebih sederhana dengan menggunakan bahasa baku (natural English) dan mudah dipahami. Contoh menggunakan flowchart jika kita ingin mencari jumlah dari 2 buah bilangan:



***Gambar 1.20 Contoh flowchart untuk penjumlahan bilangan***

Contoh pseudocode:

begin

input a, b,c

set a to 2

set b to 3

c = a+b

print c

end

**LATIHAN :**

1. Sebutkan kelebihan Java dibanding bahasa pemrograman lainnya
2. Jelaskan fitur-fitur yang dimiliki oleh J2EE 1.4
3. Jelaskan definisi dari algoritma
4. Jelaskan perbedaan antara versi J2SE, J2EE dan J2ME
5. Jelaskan istilah berikut ini :

-Domain -Web server

-Application Server -IP statis

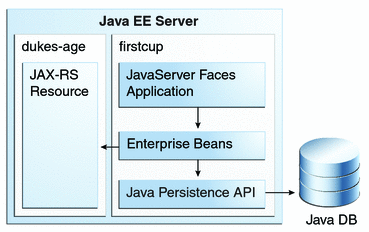
-Interopability -IP dinamis

-EJB -Internet

-RMI -Intranet

-Upload -Ekstranet

1. Buatlah contoh aplikasi bisnis di J2EE dwengan arsitektur sebagai berikut:



***Gambar 1.21 Demo aplikasi bisnis J2EE***

1. Buatlah aplikasi sederhana Bubble Sort dan Merge Sort.
2. Buat aplikasi java sederhana yang dapat menampilkan pesan mengugnakan println() sebagai berikut :

**##################################################**

**\* NAMA : MUHAMMAD YUSUF UMAIR \***

**\* JURUSAN : TEKNIK INFORMATIKA \***

**\* MK : WEB PROGRAMMING \***

**##################################################**

1. Buat aplikasi java yang mampu menampilkan menerima input harga dan jumlah barang, lalu total perkalian harga dan jumlah barang tersebut ditampilkan.

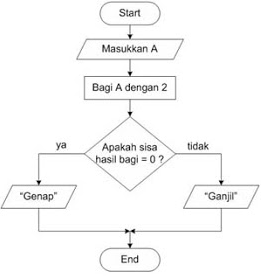
Contoh:

**Masukkan jumlah barang: 5**

**Masukkan harga: 5000**

**Total: 25000**

1. Buatlah program Java menggunakan flowchart berikut:

****

**Gamabr 1.22 contoh soal pemilihan kondisi**

1. Buatlah pseudocode dari program Java berikut:

**ContohIfElse.java**

public class ContohIfElse {

public static void main(String[] args) {

int nilai = 10;

String pesan="", jumlah="";

if (nilai>2) {

jumlah = "Lebih besar dari 2";

if(nilai%2 == 0){

pesan = "Bilangan Genap";

}else{

pesan = "Bilangan Ganjil";

}

System.out.println(jumlah+" dan "+pesan);

}else{

jumlah = "Lebih kecil dari 2";

System.out.println(jumlah);

}

}

}

### Bab 2

### Pemrograman OOP

### Tujuan Instruksional Umum

* Setelah memelajari Bab ini, mahasiswa mampu menjelaskan dasar pemrograman berorientasi obyek/ Object Oriented Programming (OOP)

**Tujuan Instruksional Khusus:**

* Mahasiswa mampu menjelaskan class di Java
* Mahasiswa mampu membuat obyek dari kelas
* Mahasiswa mampu membuat program OOP termasuk inheritance, encapsulation dan polimorphism
* Mahasiswa mampu menerapkan OOP pada pemrograman Windows menggunakan Swing
* Mahasiwa mampu membuat kelas untuk melakukan koneksi ke databsae MySQL

**Pemrograman Berorientasi Obyek (OOP)**

Pemrograman OOP memungkinkan kita membangun peranti lunak skala besar dan GUI secara efektif. Inti dari pemrograman OOP di java ialah adanya **kelas(class)** pada program. Kelas merupakan template, blueprint yang mendefinisikan kolom variabel dan method yang tersedia. Class merupakan sebuah tipe data yang mengenkapsulasi data. Implementasi dari variabel dan method dari tipe data ini disebut sebagai obyek.

*OOP It is a technique to analyze given problems and convert it through objects to depict our solutions. An object represents an entity in the real world that obviously can be identified.*

Variabel di dalam kelas disebut variabel instance dan fungsi pada kelas disebut method kelas. Variabel dan method di dalam kelas disebut sebagai anggota kelas. Rumus standar pemrograman OOP adalah:

1. Buat obyek dari suatu kelas

Nama\_Kelas obj= new Nama\_Kelas();

Contoh: Supermarket p = new Supermarket();

1. Set properti

Obj.nama\_properti/variabel=value

Contoh: p.color=”Yellow”;p.tinggi=5;

1. Panggil method

Obj.nama\_method();

Contoh: p.cetakData();

Untuk mengatur akses terhadap variabel dan method, terdapat istilah acess modifier di Java. Atribut dan method dapat memiliki salah satu sifat access modifier berikut:

• **Private**, hanya dapat digunakan di kelasnya, tidak dapat dipanggil dari luar kelas.  
• **Protected,** hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya  
•**Public,** dapat dipanggil oleh kelas manapun.

Lebih lengkapnya dapat di lihat di tabel di bawah:

Tabel 2.1 Access modifier di Java

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Access Levels** | | | | |
| **Modifier** | **Class** | **Package** | **Subclass** | **World** |
| public | Y | Y | Y | Y |
| protected | Y | Y | Y | N |
| *no modifier* | Y | Y | N | N |
| private | Y | N | N | N |

Berikut contoh penulisan kelas yang umum bernama Lingkaran, yang memiliki variabel dan method dengan access modifier tertentu:

**Lingkaran.java:**

package selamatdatang;

class Lingkaran{ // Simpan sebagai Lingkaran.java

// variabel instance Private

private double radius;

private String color;

// Construktor overloaded

public Lingkaran() { // Konstruktor ke 1

radius = 1.0;

color = "red";

}

public Lingkaran(double r, String inputcolor) { // Konstruktor ke 2

radius = r;

color = "Blue";

}

public Lingkaran(double r) { // Konstruktor ke 3

radius = r;

color = "Green";

}

// method public

public double getRadius() {

return radius;

}

public String getColor() {

return color;

}

public double getArea() {

return radius \* radius \* Math.PI;

}

}

Untuk mencoba menggunakan kelas di atas, gunakan kelas utama sebagai berikut:

**TesLingkaran.java:**

package selamatdatang;

import selamatdatang.Lingkaran;

public class TesLingkaran { // simpan sebagai "TesLingkaran.java"

public static void main(String[] args) { // Program entry point

// Deklarasi dan konstruksi instance c1

Lingkaran c1 = new Lingkaran(2.0, "blue"); // Menggunakan konstruktor ke 3

System.out.println("Radius adalah: " + c1.getRadius());

System.out.println("Warna: " + c1.getColor());

System.out.printf("Area: %.2f%n", c1.getArea());

// Deklarasi dan konstruksi instance c2

Lingkaran c2 = new Lingkaran(3.0);

System.out.println("Radius: " + c2.getRadius());

System.out.println("Warna: " + c2.getColor());

System.out.printf("Area: %.2f%n", c2.getArea());

}

}

Hasilnya:

Radius adalah: 2.0

Warna: Blue

Area: 12.57

Radius: 3.0

Warna: Green

Area: 28.27

Terlihat pada program di atas, pemrograman untuk kasus komplek menjadi lebih singkat jika menggunakan OOP. Contoh penerapan lain penggunaan OOP dalam aplikasi bisnis sebagai berikut:

**Supermarket.java:**

package selamatdatang;

import java.util.Scanner;

public class Supermarket{

private double harga, jumlah, total;

Scanner user\_input = new Scanner( System.in );

void inputHarga(){

System.out.println ("Masukkan harga");

harga=Double.parseDouble(user\_input.next());

}

void inputJumlah(){

System.out.println ("Masukkan jumlah");

jumlah=Double.parseDouble(user\_input.next());

}

void displayTotal() {

//Hitung total

total=harga\*jumlah;

System.out.println ("Total adalah: " + total);

}

public static void main(String []args){

System.out.println ("SUPERMARKET SEJAHTERA");

Supermarket p=new Supermarket();//buat obyek

p.inputHarga();

p.inputJumlah();

p.displayTotal();

}

}

Hasilnya:

SUPERMARKET SEJAHTERA

Masukkan harga

5000

Masukkan jumlah

4

Total adalah: 20000.0

**Inheritance**

Inheritance atau turunan di Java inheritance merupakan mekanisme dimana sebuah obyek mewarisi dan memiliki properti dan method dari parent atau kelas atasnya. Kelas atas disebut superclass, sedangkan kelas turunan disebut subclass. Inheritance merepresentasikan **IS-A relationship**, juga dikenal sebagai *parent-child* relationship. Untuk menandakan turunan, digunakan keword **extends** sebagai berikut:

public class Animal{}  
public class Mammal extends Animal{}  
public class Reptile extends Animal{}  
public class Dog extends Animal{}

Berikut contoh penerapan inheritance:

**Programmer.java:**

package selamatdatang;

class Employee{

float salary=9000000;

}

class Programmer extends Employee{

int bonus=10000;

public static void main(String args[]){

Programmer p=new Programmer();

System.out.println("Gaji Programmer Rp.:"+p.salary);

System.out.println("Bonus Programmer:"+p.bonus);

}

}

Hasilnya:

Gaji Programmer Rp.:9000000.0

Bonus Programmer:10000

**Enkapsulasi**

Enkapsulasi merupakan mekanisme membungkus*(wrapping)* data dan method dalam sebuah unit yang dikenal juga dengan ***data hiding***. Keuntungan dari enkapsulasi ialah kolom atau field dari kelas dapat dibuat read-only atau write-only serta kelas memilik kontrol total dari apa yang disimpan. Berikut contoh enkapsulasi dengan menggunakan private dan public pada variabel dan method.

**EncapTest.java:**

public class EncapTest {

private String name;

private String idNum;

private int age;

public int getAge() {

return age;

}

public String getName() {

return name;

}

public String getIdNum() {

return idNum;

}

public void setAge( int newAge) {

age = newAge;

}

public void setName(String newName) {

name = newName;

}

public void setIdNum( String newId) {

idNum = newId;

}

public class RunEncap {

public static void main(String args[]) {

EncapTest encap = new EncapTest();

encap.setName("Anggita ");

encap.setAge(20);

encap.setIdNum("D1986");

System.out.print("Nama : " + encap.getName() + " Umur : " + encap.getAge());

}

}

Hasilnya: Nama : Anggita Umur : 20

**Polimorphism**

Polymorphism merupakan kemampuan obyek untuk dapat berwujud berbagai bentuk, berikut contohnya:

**Animal.java:**

package selamatdatang;

public class Animal{

public void sound(){

System.out.println("Animal is making a sound");

}

}

**Cat.java:**

package selamatdatang;

public class Cat extends Animal{

@Override

public void sound(){

System.out.println("Meow meowww");

}

public static void main(String args[]){

Animal obj = new Cat();

obj.sound();

}

}

**Horse.java:**

class Horse extends Animal{

@Override

public void sound(){

System.out.println("Neigh");

}

public static void main(String args[]){

Animal obj = new Horse();

obj.sound();

}

}

**Abstraction**

Sebuah kelas berisi kata kunci abstract pada deklarasinya dikenal sebagai kelas abstract. Kelas abstract mirip dengan interfaces dimana kita tidak bisa membuat instance. Jika sebuah kelas menyertakan method abstract, maka kelas tefsebut harus dideklarasikan abstract juga sebagai berikut:

public abstract class GraphicObject {

// deklarasi field-field

// deklarasi method non abstrak

abstract void draw();

}

Berikut contoh penerapan kelas abstract:

**Employee.java:**

public abstract class Employee {

private String name;

private String address;

private int number;

public Employee(String name, String address, int number) {

System.out.println("Constructing an Employee");

this.name = name;

this.address = address;

this.number = number;

}

public double computePay() {

System.out.println("Inside Employee computePay");

return 0.0;

}

public void mailCheck() {

System.out.println("Mailing a check to " + this.name + " " + this.address);

}

public String toString() {

return name + " " + address + " " + number;

}

public String getName() {

return name;

}

public String getAddress() {

return address;

}

public void setAddress(String newAddress) {

address = newAddress;

}

public int getNumber() {

return number;

}

}

**AbstractDemo.java:**

public class AbstractDemo {

public static void main(String [] args) {

/\* Following is not allowed and would raise error \*/

Employee e = new Employee("George W.", "Houston, TX", 43);

System.out.println("\n Call mailCheck using Employee reference--");

e.mailCheck();

}

}

**Penerapan OOP pada Pemrograman Windows**

Pada Java, pemrograman Windows dapat menggunakan paket AWT(Abstract Windowing Toolkit) dan Swing. Namun, untuk grafik terkini dan hasil yang bagus, saat ini umumnya menggunakan paket Swing. Untuk membuat aplikasi windows, dibutuhkan Frame dan Panel yang dapat dibuat menggunakan kelas JFrame dan JPanel. Sedangkan untuk mengatur komponen seperti button, combobox, textbox dan lainnya di posisi yang diinginkan, dikenal dengan layout manager. Layout manager di Java antara lain flowlayout, grid layout, gridbag layout dan border layout. Berikut contoh membuat 1 form windows sederhana:

**Form.java:**

package selamatdatang;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JTextField;

import javax.swing.JFrame;

import java.awt.FlowLayout;

public class Form extends JFrame{

public Form(){

//To set FlowLayout with left alignment

//By specifying the horizontal gap 1

//and vertical gap 2

setLayout(new FlowLayout(FlowLayout.LEFT, 10, 20));

//To add label and text field into the frame

add(new JLabel("First Name : "));

add(new JTextField(8));

add(new JLabel("Middle Name : "));

add(new JTextField(5));

add(new JLabel("Last Name : "));

add(new JTextField(8));

}

public static void main(String[] args) {

Form frame = new Form();

frame.setTitle("Show Flow Layout");

frame.setSize(200, 200);

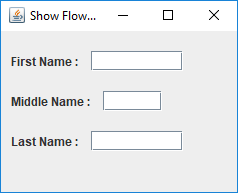
frame.setLocationRelativeTo(null);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setVisible(true);

}

}



Gambar 2.1 Hasil Program aplikasi Windows dengan Swing.

Berikut contohnya:

**EntriData.java:**

package selamatdatang;

//Demo Program Swing

//Kuliah Business Application Development 2018

import java.awt.GridLayout;

import javax.swing.\*;

class EntriData {

public static void main(String[] args) {

JTextField NAMA = new JTextField("");

String[] JURUSAN = {"Sistem informasi", "Computer Science", "DKV"};

JComboBox combo = new JComboBox(JURUSAN);

JRadioButton s1 = new JRadioButton("S1");

JRadioButton s2 = new JRadioButton("S2");

JPanel panel = new JPanel(new GridLayout(0, 1));

panel.add(new JLabel("NAMA:"));

panel.add(NAMA);

panel.add(new JLabel("JURUSAN"));

panel.add(combo);

panel.add(new JLabel("JENJANG:"));

panel.add(s1);

panel.add(s2);

ButtonGroup bG = new ButtonGroup();

bG.add(s1);

bG.add(s2);

int result = JOptionPane.showConfirmDialog(null, panel, "ENTRY DATA",

JOptionPane.OK\_CANCEL\_OPTION, JOptionPane.PLAIN\_MESSAGE);

if (result == JOptionPane.OK\_OPTION) {

JOptionPane.showMessageDialog(null, "Hasil: " + NAMA.getText()+"

Jurusan:"+combo.getSelectedItem() ,"DATA",

JOptionPane.INFORMATION\_MESSAGE);

System.out.println(combo.getSelectedItem()+ " " + NAMA.getText()

+ " " + s1.getText()+ s2.getText());

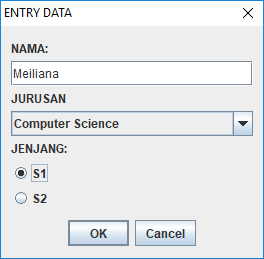
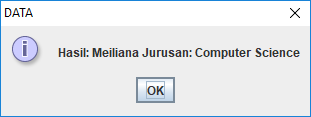
} else {

}

}

}

Jika dijalankan, programnya akan menampilkan sebuah frame sebagai berikut:

Gambar 2.2 Hasil Program dengan kontrol textbox dan lainnya

**Latihan:**

1. Jelaskan definisi dari kelas dan method.
2. Berikan definisi dari encapsulatiion, inheritance dan polimorphism
3. Jelaskan perbedaan antara AWT dan Swing.
4. Buatlah program OOP untuk kasus Supermarket yang menerima input harga dan kuantitas barang, jika pembelian barang 3 ke atas, diberikan diskon 10%. Berikut contohnya:

**PenerapanDiskon.java:**

package selamatdatang;

class Diskon{

protected double harga, qty, disc,total;

void setDiskon(){

if (qty>3) { disc=0.1;

total=(qty\*harga)-((qty\*harga)\*disc); //disc 10%)

System.out.println ("Anda dapat disc 10%. Total yang harus dibayar= " + total);

}

else {

total=(qty\*harga);

System.out.println ("Total yang harus dibayar= " + total);

}

}

}

public class PenerapanDiskon extends Diskon {

public static void main(String [] args) {

Diskon r=new Diskon();

r.harga=1000;

r.qty=5; System.out.println ("SUPERMARKET MAPAN");

r.setDiskon();

}

}

1. Sebutkan mengenai FlowLayout, Grid Layout dan Border Layout.
2. Buatlah program Entri data Mahasiswa menggunakan Swing dan menggunakan Grid Layout
3. Buatlah kelas koneksi di java untuk koneksi database MySQL

Contoh:

create database Kampus;

use Kampus;

create table emp(id **int**(10),name varchar(40),age **int**(3));

**Koneksi.java:**

import java.sql.\*;

class Koneksi{

public static void main(String args[]){

try{

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection(

"jdbc:mysql://localhost:3306/Kampus","root","root");

//Nama database Kampus, root merupakan username dan password

Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery("select \* from emp");

while(rs.next())

System.out.println(rs.getInt(1)+"  "+rs.getString(2)+"  "+rs.getString(3));

con.close();

}catch(Exception e){ System.out.println(e);}

}

}