**BAB II**

**LANDASAN TEORI**

* 1. **Penjadwalan**

Pengertian jadwal menurut kamus besar bahasa Indonesia adalah pembagian waktu berdasarkan rencana pengaturan urutan kerja, daftar atau tabel kegiatan atau rencana kegiatan dengan pembagian waktu pelaksanaan yang terperinci. Sedangkan pengertian penjadwalan adalah proses, cara, perbuatan menjadwalkan atau memasukkan ke dalam jadwal.

Menurut Chambers (1995:22) menyatakan bahwa jadwal didefinisikan sebagai sesuatu yang menjelaskan di mana dan kapan orang-orang dan sumber daya berada pada suatu waktu..

Kebanyakan orang terbiasa dengan jadwal pelajaran yang disajikan sebagai tabel hari dalam seminggu dan jangka waktu. Dapat dilihat bahwa setiap hari dibagi ke dalam jangka waktu. Setiap jangka waktu memiliki daftar mata kuliah yang sedang diajarkan, oleh siapa dan di mana. Jadwal dapat dinyatakan dalam sejumlah cara yang berbeda, masing-masing mahasiswa harus memiliki jadwal sendiri tergantung pada mata pelajaran, begitu juga masing-masing guru dan ruang, semua ini adalah perspektif yang berbeda pada jadwal yang sama.

Penjadwalan dapat didefinisikan sebagai pengaturan pengalokasian sumber daya dalam jangka waktu tertentu untuk melakukan serangkaian tugas yang telah dirancangkan.

Penjadwalan adalah aturan atau proses pengorganisasian, pemilihan dan penentuan waktu, penggunaan tempat atau sumber – sumber untuk mengerjakan semua aktifitas yang diperlukan yang memenuhi kendala aktiitas dan sumber daya.

Penjadwalan proses belajar mengajar merupakan pengaturan perencanaan belajar mengajar yang meliputi mata pelajaran, guru, waktu dan tempat pada sekolah.

* 1. **Komite Olahraga Nasional Indonesia (KONI)**

Pada tahun 1938, Di tengah keresahan terhadap diskriminasi penggunaan fasilitas olahraga, para pemuda Indonesia mendirikan Ikatan Sport Indonesia (ISI). Berbentuk federasi, ISI beranggotakan PSSI (Perserikatan Sepak Bola Indonesia), Pelti (Perserikatan Lawn Tenis Indonesia), dan Perserikatan Bola Keranjang Seluruh Indonesia (PBKSI).

Tahun 1946, PORI (Persatuan Olahraga Republik Indonesia) sebagai badan olahraga bersifat nasional dan KORI (Komite Olimpiade Republik Indonesia) dibentuk oleh para Pemimpin olahraga eks pengurus GELORA

Tahun 1948, PONI diselenggarakan di Surakarta (9-9-1948). PORI dan KORI membentuk delegasi untuk menghadiri Olympic Games XIV, London, namun gagal karenasituasi politik.

Tahun 1949, Kongres PORI III, Induk Organisasi mendapat hak otonomi, PORI sebagai badan koordinator.

Tahun 1950, PORI diubah jadi Persatuan Olahraga Indonesia (PORI) KORI diubah jadi KOI (Komite Olimpiade Indonesia).

Tahun 1951, Indonesia ikut serta dalam Asian Games I di New Delhi. Dalam persiapan tim ke Asian Games terjadi adanya tumpang tindih pelaksanaan tugas antara PORI dan KOI. Pada Kongres PORI – KOI bertepatan dengan PON II di Jakarta, dicapai kesepakatan bahwa demi efisiensi PORI melebur ke KOI. Ketua tetap Sri Sultan Hamengku Buwono IX.

Tahun 1952, KOI mendapat pengakuan IOC dan untuk pertama kali Indonesia ikut serta pada Olympic Games XV Helsinki.

Tahun 1959, DAGI (Dewan Asian Games Indonesia) dibentuk pemerintah. Tugas DAGI mempersiapkan penyelenggaraan Asian Games IV 1962. KOI sebagai badan pembantu DAGI dalam hubungan internasional.

Tahun 1961, KOGOR (Komando Gerakan Olahraga) dibentuk pemerintah. Tugas KOGOR mempersiapkan pembentukan tim nasional Indonesia. Induk-induk organisasi olahraga sebagai pelaksana teknis cabang olahraga yang bersangkutan. KOGOR dibentuk ditiap daerah tingkat I, bertugas menggerakkan olahraga membina bibit menunjang pembinaan olahraga nasional. Demokrasi terpimpin dengan pengerahan segenap “fund and forces” semaksimal mungkin.

Tahun 1962, DEPORA (Departemen Olahraga) dibentuk, dipimpin oleh Menteri Maladi. Asian Games IV diselenggarakan di Jakarta (24/8 - 4/9 – 1962).

Tugas DEPORA :

* Mengelola pembibitan, pembinaan SDM (penataran pembina,pelatih,dsb)
* Penelitian dan pengembangan.
* Dukungan anggaran serta pembangunan dan pengembangan prasarana sarana olahraga.

Tahun 1963, Februari, KOI diskors oleh IOC karena tidak mengundang Israel dan Taiwan dalam Asian Games IV. Juni, skorsing dicabut oleh IOC. GANEFO I diselenggarakan di Jakarta (10 – 22/11-1963).

Tahun 1964, DORI (Dewan Olahraga Republik Indonesia) dibentuk pemerintah. Semua organisasi, KOGOR, KOI, Induk organisasi olahraga dimasukkan dalam DORI.

Tugas DORI :

* Menetapkan kebijakan umum olahraga.
* Membina dan mengawasi seluruh kegiatan olahraga

Tahun 1965, Sekretariat Bersama Induk-induk Organisasi Cabang Olahraga dibentuk (25 Desember 1965), mengusulkan mengganti DORI menjadi Komite Olahraga Nasional Indonesia yang mandiri dan bebas dari pengaruh politik. Presidium Sekretariat Bersama (Sek Ber). Brig.Jen. Jonosewojo (PELTI). Kombes Pol. Tjoek Soejono (PABBSI). Drs. Ferry Sonneville (PBSI). Kol. Saelan (PSSI)

1966- SK Presiden (Soekarno) No. 143 A dan 156 A tahun 1966 mengukuhkan dibentuknya KONI (“lama”) sebagai pengganti DORI. Badan baru ini tidak dapat berfungsi karena tidak didukung oleh Induk Organisasi Olahraga berkenaan situasi politik pada masa itu. Kabinet Ampera dibentuk oleh Presiden Soeharto. Depora dibubarkan dan dibentuk Direktorat Jenderal Olahraga dibawah Kementerian Pendidikan dan Kebudayaan.

KONI (“baru”) dibentuk oleh Induk Organisasi Olahraga (31 Desember 1966) dengan Ketua Sri Sultan Hamengbuwono IX.

Tahun 1967, KONI dikukuhkan dengan SK Presiden (Soeharto) No. 57 Tahun 1967.

Status KONI :

* KONI adalah badan mandiri dan non pemerintah, artinya kegiatan olahraga kembali kepada masyarakat.
* KONI sebagai mitra membantu pemerintah dibidang olahraga.
* KONI tidak dikendalikan kelompok kekuasaan dan bebas dari kepentingan politik.

Tahun 1978, Dengan alasan efisiensi KONI – KOI menjadi satu, pengurusnya sama namun fungsinya berbeda. KONI melakukan pembinaan di dalam negeri, KOI melakukan kegiatan dalam hubungan luar negeri.

Tahun 1981, MUSORNAS IV KONI mengesahkan AD/ART dengan menetapkan KONI ibarat sekeping mata uang dua sisi yang kedalam menjalankan tugasnya sebagai KONI dan keluar berstatus sebagai KOI. Kondisi tersebut berlangsung sampai keberadaan KONI saat ini.

Tahun 2005, Pemerintah menerbitkan Undang-undang Nomor 3 Tahun 2005 Tentang Sistem Keolahragaan Nasional dan memecah KONI menjadi KON dan KOI. KON melakukan pembinaan dalam negeri dan penyelenggaraan Pekan Olahraga Nasional, KOI melakukan kegiatan pengiriman atlet keluar negeri dan penyelenggara pekan olahraga internasional di Indonesia.

Tahun 2007, Pemerintah menerbitkan Peraturan Pemerintah Nomor 16, 17, dan 18 Tahun 2007 sebagai peraturan pelaksanaan UU No.3 Tahun 2005. KONI menyelenggarakan Musornas Luar Biasa (Musornaslub) antara lain mengesahkan anggaran dasar KONI dan KOI serta Rita Subowo sebagai Ketua Umum KONI dan KOI masa bakti 2007-2011.

Tahun 2010, Rakor di Surabaya, seluruh peserta KONI Provinsi merekomendasikan pembentukan Pokja Amandemen UU No. 3 Tahun 2005 tentang SKN dan penyatuan KONI dan KOI.

Tahun 2011, Musornas KONI di NTB, Tono Suratman terpilih sebagai Ketua Umum KONI Pusat.

Tahun 2012, RAT KONI memutuskan perlu adanya penyempurnaan AD/ART KONI. Dibentuk Pokja dari unsur KONI Pusat, KONI DKI Jaya, Kalbar, Sultra, PB Perbakin, dan PB IKASI. Pada Desember 2012, Rembuk Olahraga Nasional KONI di Balikpapan merekomendasikan perlunya pemerintah menetapkan aturan untuk KONI yang implementatif sebagai satu-satunya wadah organisasi dalam mencapai prestasi Nasional menuju prestasi Internasional.

* 1. **Pengertian Atlet**

Masyarakat yang sering melakukan olahraga dapat disebut sebagai olahragawan, namun belum tentu bisa disebut sebagai atlet. Atlet sendiri berasal dari bahasa Yunani, dari kata Athlos yang berarti kontes. Atlet adalah orang yang ikut serta dalam. Jadi seseorang bisa dikatakan seorang atlet jika orang tersebut telah ikut suatu kompetisi olah raga yang kompetitif dalam suatu turnamen.

Atlet harus mempunyai kemampuan fisik yang lebih tinggi dari rata-rata masyarakat umum, tak terkecuali kemampuan dalam kebugaran jasmaninya. Hal tersebut sangat dibutuhkan untuk menunjang performa ataupun kinerja atlet itu sendiri. Performa atlet merupakan salah satu penentu kemenangan pada sebuah pertandingan. Atlet dapat menunjang kebugaran jasmaninya dengan latihan. Djoko Pekik Irianto, dkk (2009: 1), menyatakan bahwa latihan merupakan langkah penyempurnaan berolahraga melalui pendekatan ilmiah, khususnya prinsip-prinsip pendidikan, secara teratur dan terencana sehingga mempertinggi kemampuan dan kesiapan olahragawan. Latihan juga merupakan suatu program pengembangan atlet untuk bertanding, berupa peningkatan keterampilan dan kapasitas energi.

* 1. **Artificial Bee Colony (ABC)**
     1. **Pengertian Algoritma ABC**

Algoritma Artificial Bee Colony merupakan algoritma optimasi yang diusulkan oleh Dervis Karaboga pada tahun 2005, yang modelnya diambil dari tingkah laku cerdas lebah madu dalam mencari sumber makanannya. Dengan konsep sederhana yang memiliki ruang besar untuk pengembangan dan teorinya yang cukup mudah untuk dimengerti dan diimplementasikan, algoritma ABC semakin mendapat perhatian dari kalangan peneliti dan telah beberapa kali digunakan dalam beberapa kasus optimasi seperti Job Shop Scheduling dan Travelling Salesman Problem. Namun seperti algoritma yang berbasiskan local search umumnya, penggunaan algoritma ABC erat hubungannya dengan konvergensi yang prematur, stagnasi, dan waktu eksekusi yang lama. Selain itu, terdapat juga kemungkinan bahwa solusi terbaik yang didapat tidak ideal atau tidak sesuai dengan harapan.

Algoritma Artificial Bee Colony merupakan satu dari sekian banyak algoritma optimasi yang diadaptasikan dari konsep Swarm Intelligence (SI) yang

mendapat perhatian dari banyak kalangan peneliti. Algoritma ini pertama diusulkan oleh Dervis Karaboga pada tahun 2005. Seperti diimplikasikan oleh namanya, algoritma ABC merupakan sebuah algoritma yang memodelkan kecerdasan kolektif lebah madu dalam mencari sumber makanan, yang terdiri dari 3 komponen esensial, yakni:

1. Food Sources (sumber makanan): nilai atau kualitas dari suatu sumber makanan ditentukan oleh jaraknya dengan sarang lebah, banyaknya jumlah makanan, dan kemudahan dalam mengambil makanan tersebut.
2. Employed Foragers: merupakan lebah-lebah yang bertugas menyimpan informasi dari sumber makanan yang ditemukan. Unemployed Foragers, merupakan lebah-lebah yang bertugas mencari sumber-sumber makanan yang dapat dieksploitasi. Terdapat 2 jenis dari Unemployed Foragers :
3. Scouts
4. Onlookers
   * 1. **Operator Algoritma ABC**

Algoritma ABC dimulai dengan sebuah populasi yang terdiri dari kumpulan agen lebah. Berdasarkan tugasnya, agen lebah terbagi menjadi tiga jenis, yaitu :

1. Scout Bees

Scout Bees merupakan agen lebah yang bertugas mencari posisi sumber

makanan di lingkungan sekitar sarang secara acak. Informasi posisi sumber makanan yang telah ditemukan akan diteruskan ke agen lebah berikutnya, Employed Bees.

1. Employed Bees

Employed Bees merupakan agen lebah yang berhubungan langsung dengan sumber makanan yang sebelumnya ditemukan oleh Scout Bees. Tugas dari Employed Bees adalah menyimpan informasi yang berhubungan dengan tiap-tiap sumber makanan, baik berupa informasi tentang jarak dan arah dari sarang, informasi tingkat profitabilitas atau kekayaan dari sebuah sumber makanan, maupun nilai kepantasan informasi sumber makanan tersebut untuk disebarluaskan. Oleh karena itu, jumlah dari Employed Bees harus ekuivalen dengan jumlah sumber makanan yang ditemukan. Dalam membagikan informasi sumber makanannya, Employed Bees melakukan tarian yang bernama Waggle Dance di Dancing Room yang bertempat pada pusat dari sarang lebah dengan Onlooker Bees sebagai yang menonton dan memilih sumber makanan yang akan dieksploitasi.

* + 1. **Onlooker Bees**

Onlooker Bees merupakan agen lebah yang bertugas untuk memilih dan mengeksploitasi sumber makanan yang informasinya disimpan oleh Employed Bees.

* 1. **Pemrograman Java**
     1. **Teknologi Java**

Java adalah suatu teknologi di dunia software komputer, yang merupakan suatu bahasa pemrograman, dan sekaligus suatu platform. Sebagai bahasa pemrograman, Java dikenal sebagai bahasa pemrograman tingkat tinggi. Java mudah dipelajari, terutama bagi programmer yang telah mengenal C/C++. Java merupakan bahasa pemrograman berorientasi objek yang merupakan paradigma pemrograman masa depan. Sebagai bahasa pemrograman Java dirancang menjadi handal dan aman. Java juga dirancang agar dapat dijalankan di semua platform, dan juga dirancang untuk menghasilkan aplikasi-aplikasi dengan performansi yang terbaik, seperti aplikasi database Oracle 8i/9i yang core-nya dibangun menggunakan bahasa pemrograman Java. Sedangkan Java bersifat neutral architecture, karena Java Compiler yang digunakan untuk mengkompilasi kode program Java dirancang untuk menghasilkan kode yang netral terhadap semua arsitektur perangkat keras.

Sebagai sebuah platform, Java terdiri atas dua bagian utama, yaitu:

1. Java Virtual Machine (JVM)
2. Java Application Programming Interface (Java API)

Sun membagi arsitektur Java membagi tiga bagian, yaitu:

1. J2EE (Java 2 Enterprise Edition)
2. J2SE (Java 2 Second Edition)
3. J2ME (Java 2 Micro Edition)
   * 1. **Java Development Kit (JDK)**

Java Development Kit (JDK) merupakan perangkat lunak yang digunakan untuk menajemen dan membangun berbagai aplikasi Java. JDK merupakan superset dari JRE, berisikan segala sesuatu yang ada di JRE ditambahkan compiler dan debugger yang diperlukan untuk mengembangkan applet dan aplikasi.

Bahasa pemrograman Java menyediakan library-library standar yang telah di-compile dan dapat langsung digunakan dalam implementasi pembuatan sebuah aplikasi. Pada library, terdapat berbagai macam class yang dapat digunakan dan telah dikelompokkan ke dalam package.

* + 1. **GUI (Graphical User Interface)**

Class dapat digambarkan sebagai cetakan untuk membuat object dengan karakteristik sesuai yang dimiliki oleh class tersebut. Class menyediakan kemampuan bagi object-object yang dibangun dari class tersebut. Class dapat dipandang sebagai gambaran umum dari sebuah object, sedangkan bentuk aktual object dalam Java disebut instance. Analoginya, andaikan Class sebagai gambar rancangan sebuah gedung, maka instance adalah gedung yang telah dibangun berdasarkan rancangan tersebut. Jadi sebenarnya tidak ada perbedaan antara Object dan Instance, hanya saja object merupakan definisi umum sedangkan instance merupakan sebutan bagi object yang nyata terlihat. Keduanya mengacu pada benda yang sama.

Dalam Java, class dikelompokkan berdasarkan fungsi umumnya, misalkan untuk membuat program berbasis GUI (Graphical User Interface), atau fungsi lainnya. Kelompok class ini disebut class library. Pemrograman tidak perlu repot membuat class library, karena development kit yang ada telah membuat library sedemikian rupa, sehingga mudah digunakan. Misalnya library java.awt untuk memprograman GUI. Dengan adanya program berbasis grafis ini, mengolah database akan menjadi lebih mudah, disamping tampilannya lebih menarik. Beberapa bagian dasar GUI adalah:

1. JLabel

JLabel namaLabel = new JLabel(“label”)

Sama seperti namanya, class ini akan membuat suatu label pada sebuah window.

1. JTextField

JTextField namaText = new JTextField(jumlah\_karakter\_max);\

JTextField adalah suatu class yang akan memudahkan dalam membentuk isian text pada suatu frame.

1. JButton

JButton namaButton = new JButton(“label\_pada\_button”);

JButton adalah suatu class yang akan membantu membentuk objek tombol.

1. JFrame

JFrame sebenarnya adalah suatu class yang disediakan untuk memudahkan dalam membuat suatu window.

1. JInternalFrame

Sama halnya dengan JFrame, JInternalFrame akan membantu dalam membuat window. Bedanya JInternalFrame akan berada didalam suatu window lainnya, jadi tidak dapat berdiri sendiri seperti JFrame.

1. JMenu

JMenu akan membuat objek menu bar pada sebuah frame.

1. ActionListener

ActionListener bukan merupakan suatu objek, melainkan sebuah class yang akan menampung aksi-aksi dari sebuah objek (yang banyak dipakai disini adalah tombol). Aksi-aksi itu akan ditampung dalam sebuah metode bernama ActionPerformed. Jadi bila menggunakan ActionListener, maka harus menyertakan metode ActionPerformed pada class tersebut.

1. Container

Container.setLayout(new GridLayout(jumlah\_baris, jumlah\_kolom));

Container akan membantu dalam mengatur penempatan objek seperti text, label,ataupun tombol. Ada banyak cara yang disediakan untuk mengatur penempatan objek. GridLayout akan membagi sebuah window menjadi beberapa baris dan kolom sesuai yang masukkan pada saat setLayout. Dan penempatan objeknya dimulai dari kiri atas lalu kekanan.

Empat hal utama untuk membangun program berbasis GUI menggunakan swing, yaitu :

1. Menciptakan dan konfigurasi component Untuk menciptakan component GUI sama seperti membuat object lain dalam Java, dengan menjalankan constructor. Setiap component dapat dilihat karakteristiknya dalam dokumentasi Java.
2. Menambahkan component pada container Setiap component dalam Java harus diletakkan didalam container. Container dalam Java terdapat di java.awt.Container. Container yang sering digunakan yaitu JFrame dan JDialog. Untuk program berbasis web, dapat menggunakan container Applet dalam class java.applet.Applet. Ketiga container tersebut merupakan top level- container, yaitu container dasar yang menjadi tempat dasar bagi component dan container yang lain. Selain tiga di atas masih ada beberapa container lain seperti JPanel.
3. Mengatur tata letak component Setelah meletakkan component dalam container, maka component dapat diatur ukuran dan tata letaknya. Selain itu dapat juga diatur warna dan bentuknya. Untuk mengatur layout, object yang digunakan yaitu LayoutManager.
4. Menangani kejadian (event-handling) yang dihasilkan oleh component Component seringkali mengalami kejadian seperti button diklik, frame digeser, dan lain sebagainya. Kejadian merupakan hasil interaksi user dan program. Kejadian ini lebih lanjut disebut dengan event. Agar program dapat merespon event, maka perlu adanya event-listener.
   * 1. **Bahasa Pemrograman Java**

Java adalah bahasa pemrograman yang dapat dijalankan diberbagai computer termasuk telepon genggam. Dikembangkan oleh Sun Microsystems dan dirilis tahun 1995. Java berbeda dengan JavaScript. JavaScript adalah bahasa scripting yang digunakan oleh web browser.

* + 1. Kelebihan Pemrograman Java

Java memiliki beberapa kelebihan dibandingkan dengan bahasa pemrograman lainnya.

Adapun kelebihan tersebut di antaranya:

1. Sederhana dan Ampuh

Java dirancang untuk mudah dipelajari, terutama bagi programmer yang telah mengenal C/C++ akan mudah sekali untuk berpindah ke Java. Pemakai dapat belajar membuat program dengan Java secara cepat jika telah memahami konsep dasar pemrograman berorientasi objek. Java member programmer kemampuan untuk menuangkan semua ide, karena bahasa pemrograman ini bukan merupakan scripting language (bahasa naskah) yang menghilangkan kemampuan programmer untuk berinovasi, tetapi dengan cara berorientasi objek yang mudah dan jelas.

1. Aman

Java dirancang sebagai bahasa pemrograman yang handal dan aman. Aplikasi-aplikasi yang dibangun dengan bahasa Java sangat handal dengan manajemen memori yang bagus. Aplikasi Java juga dikenal sangat secure, yaitu kasus- kasus seperti buffer everflow yang umumnya menjadi lubang keamanan aplikasi-aplikasi berbasis C/C++ tidak terjadi di Java, karena pengaturan keamanannya yang bagus.

1. Berorientasi-Objek

Java merupakan bahasa pemrograman berorientasi objek dan bukan turunan langsung dari bahasa pemrograman manapun, juga sama sekali tida kompetibel dengan semuanya. Java memiliki keseimbangan, menyediakan mekanisme peng-class-an sederhana, dengan model antar muka dinamik yang intuitif hanya jika diperlukan.

1. Kokoh

Kesalahan sering terjadi pada saat kompilasi karena Java merupakan bahasa pemrograman yang sensitif dalam hal deklarasi dan tipe data. Oleh karena hal tersebut, Java memiliki batasan agar dapat menemukan kesalahan lebih cepat saat mengembangkan program yaitu dengan langsung memeriksa program saat ditulis, dan sekali lagi ketika program di jalankan. Hal ini akan lebih menghemat waktu jika dibandingkan dengan keharusan menjalankan program terlebih dahulu dan memeriksa semua bagian program untuk melihat ketidakcocokan dinamis selama program berjalan.

1. Interaktif

Java memiliki kemampuan yang memungkinkan program melakukan beberapa hal pada saat bersamaan, tanpa harus kesulitan menangani proses yang akan terjadi selanjutnya. Jalinan program-program Java yang mudah digunakan memungkinkan programmer untuk memikirkan pembuatan perilaku khusus, tanpa harus mengintegrasikan perilaku tersebut dengan model pemrograman global yang mengatur perulangan kejadian.

1. Netral Terhadap Berbagai Arsitektur

Java memiliki kemampuan yang memungkinkan program melakukan beberapa hal pada saat bersamaan, tanpa harus kesulitan menangani proses yang akan terjadi selanjutnya. Jalinan program-program Java yang mudah digunakan memungkinkan programmer untuk memikirkan pembuatan perilaku khusus, tanpa harus mengintegrasikan perilaku tersebut dengan model pemrograman global yang mengatur perulangan kejadian.

1. Terinterpretasi dan Berkinerja-Tinggi

Java dirancang untuk tetap berkinerja baik pada CPU yang tidak terlalu kuat. Walaupun Java merupakan bahasa terinterpretasi, kode-kode Java telah dirancang dengan hati-hati sehingga mudah diterjemahkan ke dalam bahasa asli suatu mesin untuk menghasilkan kinerja yang tinggi. Java dilengkapi keajaiban lintas-platform yang luar biasa dengan kompilasi ke dalam representasi langsung yang disebut kode-byte Java (Java byte-code), yang dapat diterjemahkan oleh sistem manapun yang memilki program Java didalamnya. Sebagai sebuah platform.

1. Multiplatform

Kelebihan utama dari Java ialah dapat dijalankan di beberapa platform/ system operasi komputer, sesuai dengan prinsip tulis sekali, jalankan di mana sajaa. Dengan kelebihan ini pemrogram cukup menulis sebuah program Java dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin/bytecode) sekali lalu hasilnya dapat dijalankan di atas beberapa platform tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis Java dikerjakan diatas operating system Linux tetapi dijalankan dengan baik di atas Microsoft Windows. Platform yang didukung sampai saat ini adalah Microsoft Windows, Linux, Mac OS dan Sun Solaris. Penyebabnya adalah setiap system operasi menggunakan programnya sendiri-sendiri (yang dapat diunduh dari situs Java) untuk menginterpretasikan bytecode tersebut.

1. Perpustakaan Kelas yang Lengkap

Java terkenal dengan kelengkapan library/perpustakaan (kumpulan program-program yang disertakan dalam pemrograman Java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya. Kelengkapan perpustakaan ini ditambah dengan keberadaan komunitas Java yang besar yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi.

1. Bergaya C++

Dalam bahasa pemrograman manapun selalu ada fitur komentar. Java sendiri memiliki sintaks seperti bahasa pemrograman C++, dengan tambahan yang sangat istimewa yaitu komentar yang nantinya digunakan untuk menciptakan dokumentasi secara otomatis oleh suatu aplikasi yang disebut javadoc sehingga menarik banyak pemrogram C++ untuk pindah ke Java. Saat ini pengguna Java sangat banyak, sebagian besar adalah pemrogram C++ yang pindah ke Java. Universitas-universitas di Amerika Serikat juga mulai berpindah dengan mengajarkan Java kepada murid-murid yang baru karena lebih mudah dipahami oleh murid dan dapat berguna juga bagi mereka yang bukan mengambil jurusan komputer.

1. Pengumpulan sampah otomatis

Memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa C++ yang dipakai secara luas). Serupa dengan C, java mengalokasikan operator baru untuk mengalokasikan memori pada heap untuk objek baru.

* + 1. **Kekurangan Pemrograman Java**

Adapun kekurangan pemrograman Java, yaitu:

1. Tulis sekali, perbaiki di mana saja

Masih ada beberapa hal yang tidak kompatibel antara platform satu dengan platform lain. Untuk J2SE, misalnya SWT-AWT bridge yang sampai sekarang tidak berfungsi pada Mac OS X.

1. Mudah didekompilasi

Dekompilasi adalah proses membalikkan dari kode jadi menjadi kode sumber. Ini dimungkinkan karena kode jadi Java merupakan bytecode yang menyimpan banyak atribut bahasa tingkat tinggi, seperti nama-nama kelas, metode, dan tipe data. Hal yang sama juga terjadi pada Microsoft .NET Platform. Dengan demikian, algoritma yang digunakan program akan lebih sulit disembunyikan dan mudah dibajak/direverse-engineer.

1. Penggunaan memori yang banyak

Penggunaan memori untuk program berbasis Java jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnya seperti C/C++ dan Pascal (lebih spesifik lagi, Delphi dan Object Pascal). Biasanya ini bukan merupakan masalah bagi pihak yang menggunakan teknologi terbaru (karena trend memori terpasang makin murah), tetapi menjadi masalah bagi mereka yang masih harus berkutat dengan mesin komputer berumur lebih dari 4 tahun.

* 1. **Unified Modelling Language**

Mengenai pemahaman alat bantu dan perancangan berorientasi objek, maka penulis membangun sistem informasi yang berorientasi objek yang merupakan suatu perancangan yang berbeda dengan berorientasi data. Namun secara konteks perancangan ini digunakan untuk membangun sistem informasi sesuai kebutuhan dari pemakai ( user ) sistem informasi.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan syntax / semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML syntax mendefinisikan bagaimana bentuk - bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD ( Object - Oriented Design ), Jim Rumbaugh OMT ( Object Modeling Technique ), dan Ivar Jacobson OOSE ( Object - Oriented Software Engineering ).

UML sendiri menyediakan alat – alat bantu berupa diagram visual yang mnggambarkan berbagai aspek yang ada didalam sistem. Adapun alat – alat bantu diagram yang sediakan didalam UML yaitu :

1. use case diagram
2. class diagram
3. activity diagram
4. sequence diagram

Berikut merupakan penjelasan diagram – diagram UML :

1. Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara actor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, mengcreate sebuah daftar belanja, dan sebagainya. Seorang/ sebuah actor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Use case diagram dapat sangat membantu bila kita sedang menyusun requirement sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang test case untuk semua feature yang ada pada sistem.

Sebuah use case dapat menginclude fungsionalitas use case lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa use case yang diinclude akan dipanggil setiap kali use case yang menginclude dieksekusi secara normal. Sebuah use case dapat diinclude oleh lebih dari satu use case lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang common. Sebuah use case juga dapat mengextend use case lain dengan behaviournya sendiri. Sementara hubungan generalisasi antar use case menunjukkan bahwa use case yang satu merupakan spesialisasi dari yang lain.

1. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan ( atribut / properti ) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut ( metoda / fungsi ).

Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain - lain. Class memiliki tiga area pokok :

1. Nama ( stereotype )
2. Atribut
3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

1. Private, tidak dapat dipanggil dari luar class yang bersangkutan.
2. Protected, hanya dapat dipanggil oleh class yang bersangkutan dan anak - anak yang mewarisinya.
3. Public, dapat dipanggil oleh siapa saja.

Class dapat merupakan implementasi dari sebuah interface, yaitu class abstrak yang hanya memiliki metoda. Interface tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah class.

Dengan demikian interface mendukung resolusi metoda pada saat run time. Sesuai dengan perkembangan class model, class dapat dikelompokkan menjadi package. Kita juga dapat membuat diagram yang terdiri atas package.

Hubungan Antar Class

1. Asosiasi, yaitu hubungan statis antar class. Umumnya menggambarkan class yang memiliki atribut berupa class lain, atau class yang harus mengetahui eksistensi class lain. Panah navigability menunjukkan arah query antar class.
2. Agregasi, yaitu hubungan yang menyatakan bagian ( “terdiri atas..”).
3. Pewarisan, yaitu hubungan hirarkis antar class. Class dapat diturunkan dari class lain dan mewarisi semua atribut dan metoda class asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari class yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
4. Hubungan dinamis, yaitu rangkaian pesan ( message ) yang dipassing dari satu class kepada class lain. Hubungan dinamis dapat digambarkan dengan menggunakan sequence diagram.
5. Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam system yang sedang dirancang, bagaimana masing - masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi ditrigger oleh selesainya state sebelumnya ( internal processing ). Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem ( dan interaksi antar subsistem ) secara eksak, tetapi lebih menggambarkan proses - proses dan jalur – jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu use case atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Sama seperti state, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. Decision digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses - proses paralel ( fork dan join ) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. Activity diagram dapat dibagi menjadi beberapa object swimlane untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

1. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem ( termasuk pengguna, display, dan sebagainya ) berupa message yang digambarkan terhadap waktu. Sequence diagram terdiri atar dimensi vertical ( waktu ) dan dimensi horizontal ( objek - objek yang terkait ).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah - langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang mentrigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Masing - masing objek, termasuk aktor, memiliki lifeline vertikal. Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya.

Pada fase desain berikutnya, message akan dipetakan menjadi operasi / metoda dari class. Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message. Untuk objek - objek yang memiliki sifat khusus, standar UML mendefinisikan icon khusus untuk objek boundary, controller dan persistent entity.