Bab 2
Landasan Teori

Pada bab kedua ini, memberikan pembahasan tentang teori-teori yang berhubungan dengan perancangan sistem informasi pengadaan barang *department procurement* PT. Sinar terang logamjaya (Stallion). Hal-hal yang dijelaskan yaitu:

2.1 Konsep Dasar Sistem

Pengertian dan definisi sistem pada berbagai bidang berbeda-beda, tetapimeskipun istilah sistem yang digunakan bervariasi, semua sistem pada bidang-bidang tersebut mempunyai beberapa persyaratan umum, yaitu sistem harus mempunyai elemen, lingkungan, interaksi antar elemen, interaksi antara elemen dengan lingkungannya, dan yang terpenting adalah sistem harus mempunyai tujuan yang akan dicapai.

Berdasarkan persyaratan ini, sistem dapat didefinisikan sebagai seperangkat elemen yang digabungkan satu dengan lainnya untuk suatu tujuan bersama.Kumpulan elemen terdiri dari manusia, mesin, prosedur, dokumen, data atau elemen lain yang terorganisir dari elemen-elemen tersebut. Elemen sistem disamping berhubungan satu sama lain, juga berhubungan dengan lingkungannya untuk mencapai tujuan yang telah ditentukan sebelumnya.

Terdapat beberapa definisi sistem yaitu :

"Sebuah sistem terdiri dari bagian-bagian saling berkaitan yang beroperasi bersama untuk mencapai beberapa sasaran atau maksud". (Gordon B. Davis : 1984)

"Sistem adalah himpunan dari unsur-unsur yang saling berkaitan sehingga membentuk suatu kesatuan yang utuh dan terpadu". (Ramond Mcleod (2001).

2.2 Karakteristik Sistem

1. Komponen Sistem

   Komponen sistem atau elemen sistem dapat berupa :

   a. Elemen-elemen yang lebih kecil yang disebut sub sistem, misalkan sistem komputer terdiri dari sub sistem perangkat keras, perangkat lunak dan manusia.
   b. Elemen-elemen yang lebih besar yang disebut supra sistem. Misalkan bila perangkat keras adalah sistem yang memiliki sub sistem CPU, perangkat I/O dan memori, maka supra sistem perangkat keras adalah sistem komputer.

2. Batas sistem

   Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batas suatu sistem menunjukkan ruang lingkup dari sistem tersebut.

3. Lingkungan luar sistem

   Lingkungan dari sistem adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. lingkungan luar yang mengutungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedang lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung

   Penghubung merupakan media perantara antar subsistem. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya. Output dari satu subsistem akan menjadi input untuk subsistem yang lainnya dengan melalui penghubung. Dengan penghubung satu subsistem dapat berinteraksi dengan subsistem yang lainnya membentuk satu kesatuan.

5. Masukkan

   Masukan adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa maintenance input dan sinyal input. Maintenance input adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Sinyal input adalah energi yang diproses untuk didapatkan keluaran.

6. Keluaran

   Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supra sistem.

7. Pengolah

   Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan merubah masukan menjadi keluaran. Suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan-bahan yang lain menjadi keluaran berupa barang jadi.

8. Sasaran atau tujuan

   Suatu sistem pasti mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

2.3 Klasifikasi Sistem

1. Sistem abstrak dan sistem fisik. Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Misalnya sistem teologia, yaitu sistem yang berupa pemikiran- pemikiran hubungan antara manusia dengan tuhan. Sistem fisik merupan sistem yang ada secara fisik misalnya sistem komputer, sistem akuntansi dan sistem produksi.
2. Sistem alamiah (*natural system*) dan sistem buatan manusia (*human made system*). Sistem alamiah adalah sistem yang terjadi melalui proses alam, misalnya sistem perputaran bumi. Sistem buatan manusia adalah sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan

interaksi antara manusia dengan mesin disebut dengan *human machine system* atau ada yang menyebut dengan *man machine system*, karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

3. Sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*). Sistem tertentu beroperasi tertentu dengan tingkah laku yang sudah dapat diprediksi. Inteaksi diantara bagian- bagiannya dapat dideteksi dengan pasti, sehingga keluaran dari sistem dapat diramalkan. Sistem komputer adalah contoh dari sistem tertentu yang tingkah lakunya dapat dipastikan berdasarkan program-program yang dijalankan. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.
4. Sistem tertutup (*closed system*) dan sistem terbuka (*open system*). Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak diluarnya. Secara teoritis sistem tertutup ada, tetapi kenyataan tidak ada sistem yang benar-benar tertutup yang ada hanyalah *relatively closed system* (secara relatif tertutup, tidak benar-benar tertutup), sedang sistem terbuka adalah sistem yang berhubungan dan terpengaruh oleh lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lainnya, karena sistem sifatnya terbuka dan tepengaruh oleh lingkungan luarnya, maka suatu sistem harus mempunyai sistem pengendalian yang baik. Sistem-sistem yang baik harus dirancang sedemikian rupa, sehingga secara relatif tertutup karena sistem tertutup akan bekerja secara otomatis dan terbuka untuk pengaruh yang baik saja.

2.4 Konsep Dasar Informasi

2.4.1 Pengertian Informasi

Menurut Raymond Mcleod, :

"Informasi adalah data yang telah diolah menjadi bentuk yang memiliki arti bagi si penerima dan bermanfaat bagi pengambilan keputusan saat ini atau mendatang".

Secara umum informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian- kejadian yang nyata yang digunakan untuk pengambilan keputusan.

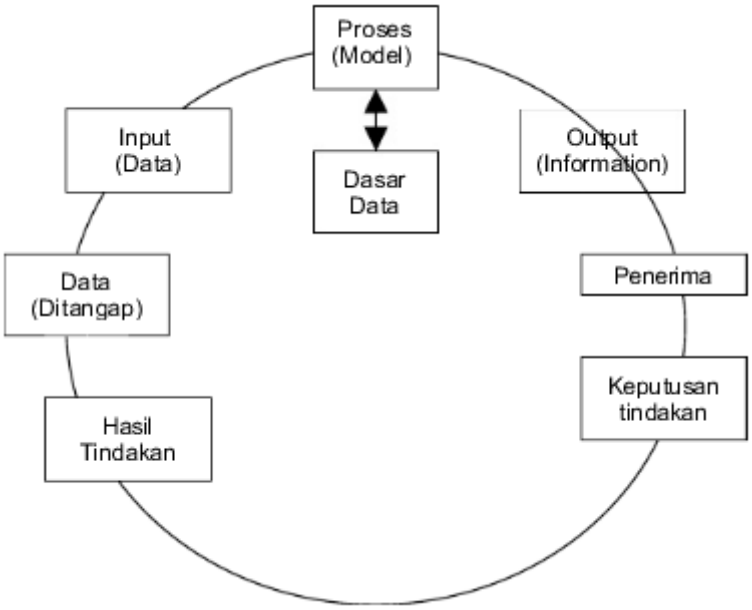
Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian adalah sesuatu yang terjadi pada saat tertentu. Di dalam dunia bisnis, kejadian-kejadian yang sering terjadi adalah transaksi perubahan dari suatu nilai yang disebut transaksi. Kesatuan nyata adalah berupa suatu obyek nyata seperti tempat, benda dan orang yang betul-betul ada dan terjadi.

Data merupakan bentuk yang masih mentah, belum dapat bercerita banyak sehingga perlu diolah lebih lanjut. Data diolah melalui suatu metode untuk menghasilkan informasi. Data dapat berbentuk simbol-simbol semacam huruf, angka, bentuk suara, sinyak, gambar, dsb.

Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data

tersebut akan ditangkap sabagai input, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus. Siklus informasi ini dapat digambarkan sebagai berikut :





Gambar 2.1 Siklus Informasi

2.4.2 Fungsi Informasi

Informasi mempunya beberapa fungsi diantaranya adalah :

1. Untuk meningkatkan pengetahuan bagi si pemakai.
2. Untuk mengurangi ketidakpastian dalam proses pengambilan keputusan pemakai.
3. Menggambarkan keadaan yang sebenarnya dari sesuatu hal.

2.4.3 Kualitas Informasi

Informasi yang berkualitas harus akurat, tepat pada waktunya dan relevan.

1. Akurat

   Berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan terjadi gangguan yang dapat merubah atau merusak informasi tersebut.

2. Tepat Waktu

   Berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan di dalam pengambilan keputusan. Bila

pengambilan keputusan terlambat, maka dapat berakibat fatal bagi organisasi. Saat ini mahalnya nilai informasi disebabkan harus cepatnya informasi itu didapat sehingga diperlukan teknologi-teknologi mutakhir untuk mendapatkan, mengolah dan mengirimkannya.

3. Relevan

   Berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang berbeda- beda.

2.5 Konsep Dasar Sistem Informasi

2.5.1 Pengertian Sistem Informasi

Menurut Mc Leod :

"Sistem Informasi merupakan sistem yang mempunyai kemampuan untuk mengumpulkan informasi dari semua sumber dan menggunakan berbagai media untuk menampilkan informasi".

Sistem informasi dapat didefinisikan sebagai suatu sistem di dalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan.

2.5.2 Komponen Sistem Informasi

Komponen sistem informasi terdiri dari 5 komponen, kelima komponen sistem informasi tersebut dapat diklasifikasikan sebagai hardware dan software yang berfungsi sebagai mesin, people dan procedures yang merupakan manusia dan tatacara menggunakan mesin, data yang merupakan jembatan penhubung antara manusia dan mesin agar terjadi suatu proses pengolahan data.

1. Hardware-hardware dalam komponen sistem informasi dapat di golongkan menjadi beberapa bagian, yaitu:
    a. Peralatan penyimpanan data Disk merupakan perangkat yang paling sering digunakan sebagai peralatan penyimpanan. Disk diorganisasikan berupa silinder-silinder dengan tiap permukaan terdapat head yang ditumpuk secara vertikal. Head terdiri dari beberapa track. Dan treck terbagi menjadi sektor- sektor.
    b. Peralatan input. Peralatan input merupakan alat yang digunakan untuk menerima input.
    c. Peralatan output

Peralah output merupakan suatu alat keluaran/tampilan suatu data setelah mengalami proses. Output yang dihasilkan dari pengolahan data digolongkan kedalam 4 macam bentuk yaitu:

1) Tulisan: terdiri dari huruf, kata, angka, karakter khusus, dan simbol-simbol lainnya.
2) Image: didalam suatu bentuk grafik atau gambar.
3) Bentuk yang dapat dibaca oleh mesin dalam bentuk simbol yang hanya dibaca dan dimengerti oleh komputer.
4) Suara: dalam bentuk musik atau omongan.
    a. Peralatan komunikasi data

       Komunikasi adalah suatu bagian dari ilmu komunikasi yang mengkhususkan diri pada penyampaian informasi yang berupa teks dan gambar.

2. Software

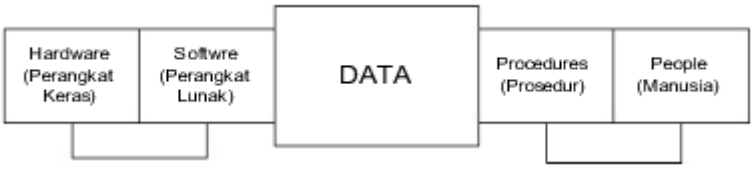
   Software merupakan kumpulan dari perintah /fungsi yang ditulis dengan aturan tertentu untuk memerintahkan komputer melaksanakan tugas tertentu.

3. Data

Data merupakan komponen dari informasi yang akan diproses lebih lanjut untuk menghasilkan informasi. Data merupakan jembatan penghubung antara manusia dan mesin agar terjadi suatu proses pengolahan data. Himpunan data akan mempunyai sifat yang unik, yaitu:

a. Saling berkaitan (interrelated), data-data tersebut akan saing berkaitan/terintegrasi dan tersimpan secara terorganisir didalam suatu media penyimpanan.
b. Kebersamaan (Shared), daya yang terintegrasi tersebut dapat diakses oleh berbagai macam pengguna/orang tetapi hanya satu yang dapat merubahnya yaitu Database Administrator (DBA).
c. Terkendali (controlled), data yang terintegrasi hanya dapat diubah oleh Database Administrator (DBA).

4. ProsedurDokumentasi prosedur/proses sistem, buku penuntun
   operasional dan teknis. Prosedur menghubungkan berbagai perintah dan aturan yang akan menentukan rancangan dan penggunaan sistem informasi.
5. Manusia
   Manusia adalah mereka yang terlibat dalam kegiatan sistem informasi seperti operator, pemimpin sistem informasi dan sebagainya.



Gambar 2.2 Lima Komponen Sistem Informasi

2.6 Konsep Dasar Sistem Komputer

Komputer berasal dari kata to compute yang artinya menghitung. Tetapi apabila kita tinjau dari kata tersebut dengan fungsi komputer dewasa ini kita melihat ketidakcocokan, dimana komputer sekarang dapat digunakan untuk berbagai pekerjaan. Secara umum komputer dapat diartikan serangkaian atau sekelompok

mesin elektronika yang terdiri dari ribuan bahkan jutaan komponen yang saling bekerjasama, serta membentuk suatu sistem kerja yang rapi dan teliti. Sistem kerja ini kemudian dapat digunakan untuk melaksanakan serangkaian pekerjaan secara otomatis berdasarkan urutan instruksi atau program yang telah diberikan kepadanya.

Komputerisasi bermakna sebagai penggunaan komputer sebagai alat bantu dalam kegiatan pengolahan data, untuk menggantikan prosedur pengolahan data secara manual. Prosedur pengolahan data yang dilakukan secara manual meliputi kegiatan pengumpulan data, melakukan pengelompokan, pengurutan, penghitungan, yang pada akhirnya menyusunnya dalam sejumlah bentuk laporan untuk berbagai keperluan yang ada. (Edi Purwono, 2002).

2.7 Perancangan Sistem

Dalam suatu proses pengembangan software, analisa dan rancangan telah merupakan terminologi yang sangat tua. Pada saat masalah ditelusuri dan spesifikasi dinegoisasikan, dapat dikatakan kita berada pada tahap rancangan. Merancang adalah menemukan suatu cara untuk menyelesaikan masalah, salah satu tool / model untuk merancang pengembangan software yang berbasis object oriented adalah UML.

2.7.1 Teknik Dasar OOA/D (Object-Oriented Analysis/Design)

Dalam dunia pemodelan, metodologi implementasi obyek walaupun terikat kaidah-kaidah standar, namun teknik pemilihan obyek tidak terlepas pada subyektifitas software analyst & designer.

Beberapa obyek akan diabaikan dan beberapa obyek menjadi perhatian untuk diimplementasikan di dalam sistem. Hal ini sah-sah saja karena kenyataan bahwa suatu permasalahan sudah tentu memiliki lebih dari satu solusi. Ada 3 (tiga) teknik/konsep dasar dalam OOA/D, yaitu pemodulan (encapsulation), penurunan (inheritance) dan polymorphism.

1. Pemodulan (Encapsulation)

   Pada dunia nyata, seorang ibu rumah tangga menanak nasi dengan menggunakan rice cooker, ibu tersebut menggunakannya hanya dengan menekan tombol. Tanpa harus tahu bagaimana proses itu sebenarnya terjadi. Disini terdapat penyembunyian informasi milik rice cooker, sehingga tidak perlu diketahui seorang ibu. Dengan demikian menanak nasi oleh si ibu menjadi sesuatu yang menjadi dasar bagi konsep information hiding.

2. Penurunan (Inheritance)

   Obyek-obyek memiliki banyak persamaan, namun ada sedikit perbedan. Contoh dengan beberapa buah mobil yang mempunyai kegunaan yang berbeda-beda. Ada mobil bak terbuka seperti truk, bak tertutup seperti sedan dan minibus. Walaupun demikian obyek-obyek ini memiliki kesamaan yaitu teridentifikasi sebagai obyek mobil, obyek ini dapat dikatakan sebagai obyek induk (parent). Sedangkan minibus dikatakan sebagai obyek anak (child), hal ini juga berarti semua operasi yang berlaku pada mobil berlaku juga pada minibus.

3. Polymorphism

   Pada obyek mobil, walaupun minibus dan truk merupakan jenis obyek mobil yang sama, namun memiliki juga perbedaan. Misalnya suara truk lebih keras dari pada minibus, hal ini juga berlaku pada obyek anak (child) melakukan metoda yang sama dengan algoritma berbeda dari obyek induknya. Hal ini yang disebut polymorphism, teknik atau konsep dasar lainnya adalahruang lingkup / pembatasan. Artinya setiap obyek mempunyai ruang lingkup kelas, atribut, dan metoda yang dibatasi.

2.7.2 Pengenalan UML

UML sebagai sebuah bahasa yang memberikan vocabulary dan tatanan penulisan kata-kata dalam ‘MS Word’ untuk kegunaan komunikasi. Sebuah bahasa model adalah sebuah bahasa yang mempunyai vocabulary dan konsep tatanan / aturan penulisan serta secara fisik mempresentasikan dari sebuah sistem. Seperti halnya UML adalah sebuah bahasa standard untuk pengembangan sebuah software yang

dapat menyampaikan bagaimana membuat dan membentuk model-model, tetapi tidak menyampaikan apa dan kapan model yang seharusnya dibuat yang merupakan salah satu proses implementasi pengembangan software.

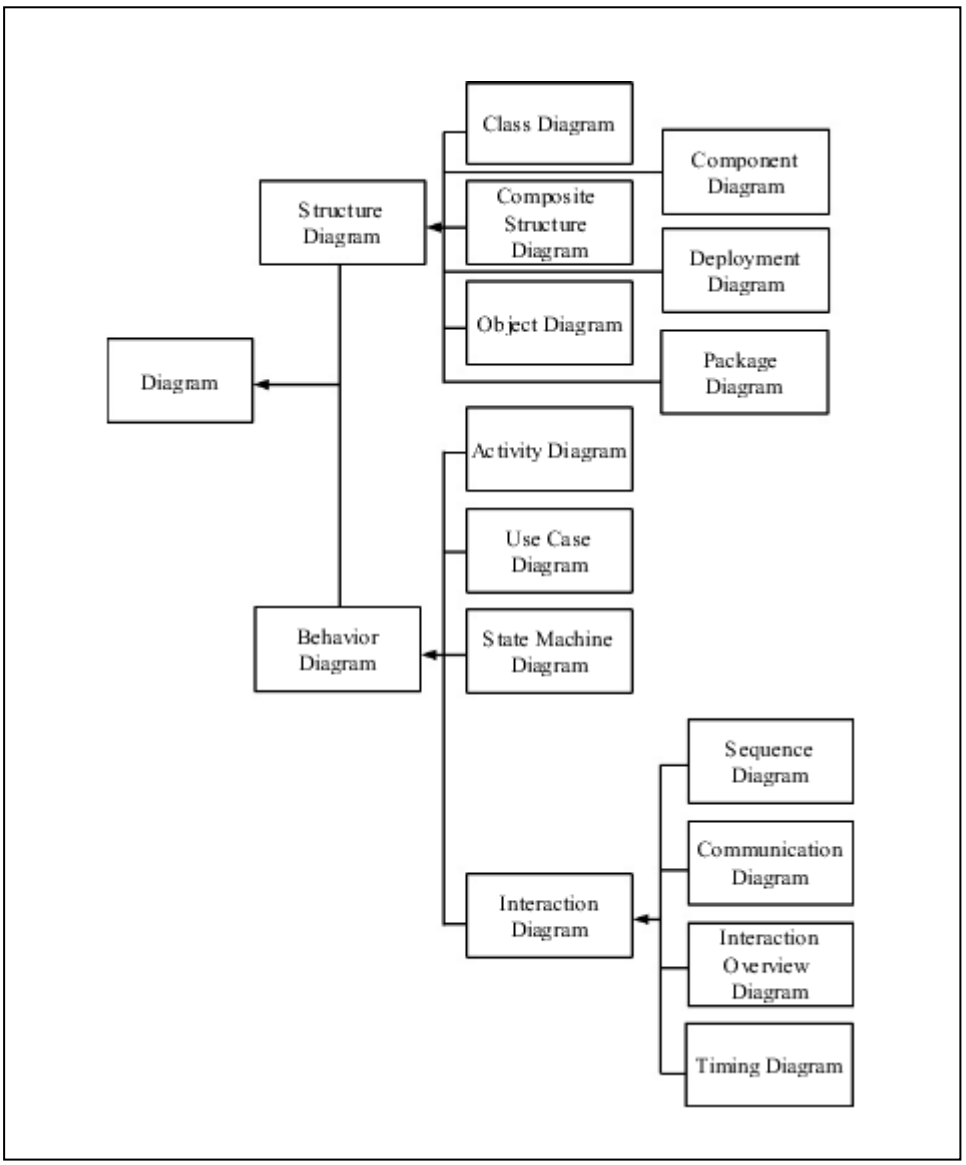
UML tidak hanya merupakan sebuah bahasa pemograman visual saja, namunjuga dapat secara langsung dihubungkan ke berbagai bahasa pemograman, seperti JAVA, C++, Visual Basic, atau bahkan dihubungkan secara langsung ke dalam sebuah object-oriented database. Begitu juga mengenai pendokumentasian dapat dilakukan seperti; requirements, arsitektur, design, source code, project plan, tests, dan prototypes.

2.7.3 Diagram-Diagram UML

UML 2 terdiri dari 13 jenis diagram resmi seperti tertulis dalam Tabel 2.1 dan mengklasifikasi mereka seperti pada gambar 2.2. Meskipun jenis-jenis diagram ini merupakan cara orang-orang memperlakukan UML dan cara mengorganisasikan buku ini, para perancang UML tidak memandang diagram sebagai bagian yang sentral. Dan hasilnya, jenis-jenis diagram sebagai bagian yang sentral. Acapkali secara legal dapat menggunakan elemen-elemen dari satu jenis diagram untuk diagram yang lain. Standard UML menunjukkan bahwa elemen-elemen tertetu hanya diambil dari jenis diagram tertentu, tetapi ini bukanlah hal yang dianjurkan.

Tabel 2.1 Tabel Jenis Diagram UML

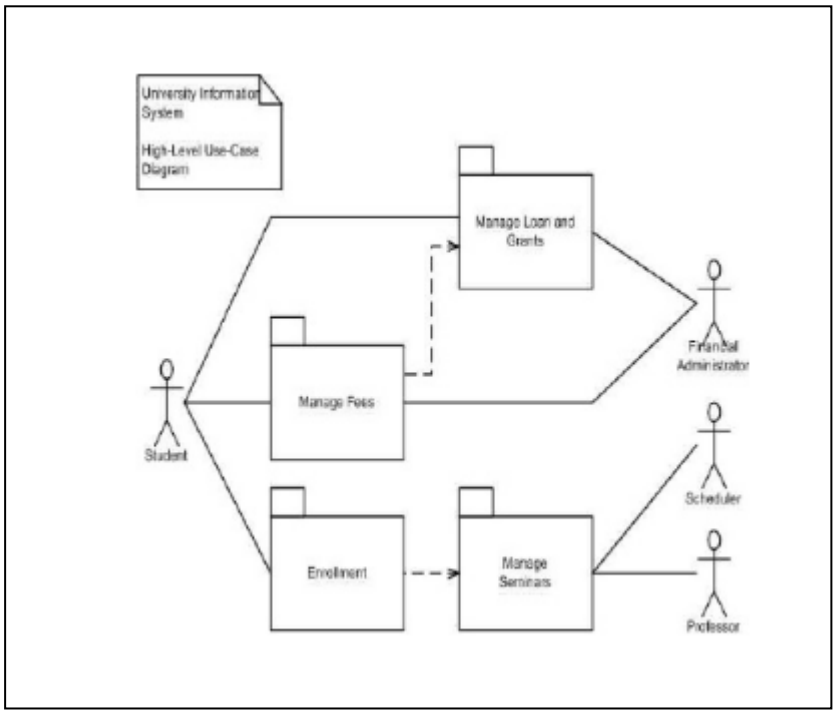
No.	Diagram	Kegunaan	Turunan
1	Activity	Behavior prosedural dan paralel	Di UML 1
2	Class	Clas, fitur, dan hubungan-hubungan	Di UML 1
3	Communication	Interaksi antar objek, penekanan pada jalur	Diagram kolaborasi UML 1
4	Component	Struktur dan koneksi komponen	Di UML 1
5	Composite structure	Dekomposisi runtime sebuah class	Baru di UML 2
6	Deployment	Pemindahan artifak ke node	Di UML 1
7	Interaction overview	Campuran sequence dan activity diagram	Baru di UML 2
8	Object	Contoh konfigurasi dari contoh-contoh	Tidak resmi di UML 1
9	Package	Struktur hierarki compile time	Tidak resmi di UML 1
10	Sequence	Interaksi antar objek, penekanan pada sequence	Di UML 1
11	State machine	Bagaimana even mengubah objek selama aktif	Di UML 1
12	Timing	Interaksi antar objek, penekanan pada timing	Baru di UML 2
13	User case	Bagaimana pengguna berinteraksi dengan sebuah sistem	Di UML 1



Gambar 2.3. Klasifikasi Diagram UML

1. Use Case Diagram

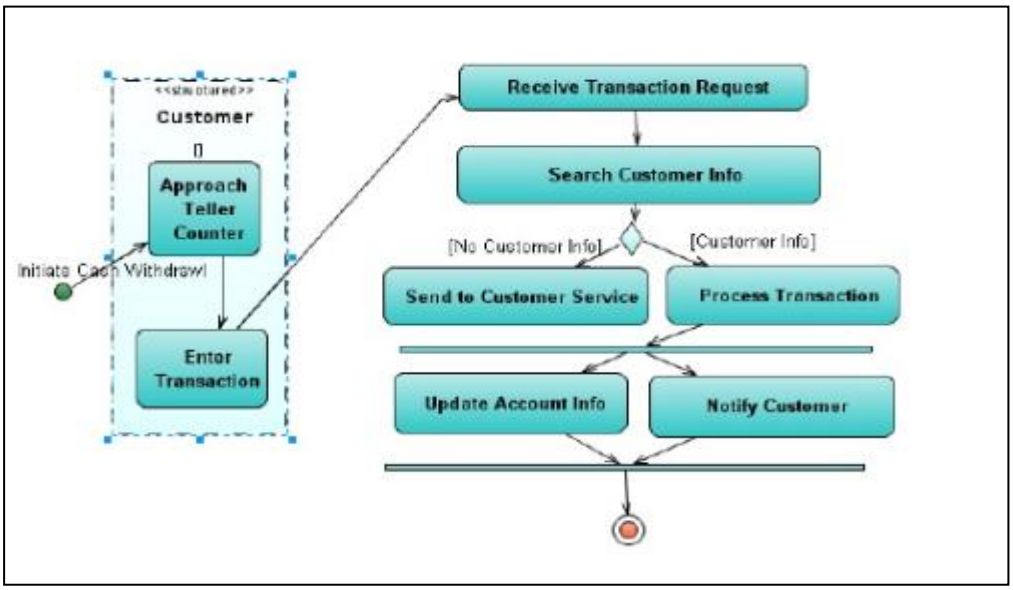
   Pada umumnya use case sistem merupakan sebuah interaksi dengan perangkat lunak, sedangkan sebuah use case bisnis berbicara tentang bagaimana sebuah bisnis menanggapi seorang konsumen atau sebuah event. Use case diagram berfungsi untuk menggambarkan kebutuhan sistem dari sudut pandang user, mengfokuskan pada proses komputerisasi (automated processes), menggambarkan hubungan antara use case dan actor, serta menggambarkan proses sistem (kebutuhan sistem dari sudut pandang user).



Gambar 2.4. Contoh Use Case Diagram

2. Activity Diagram

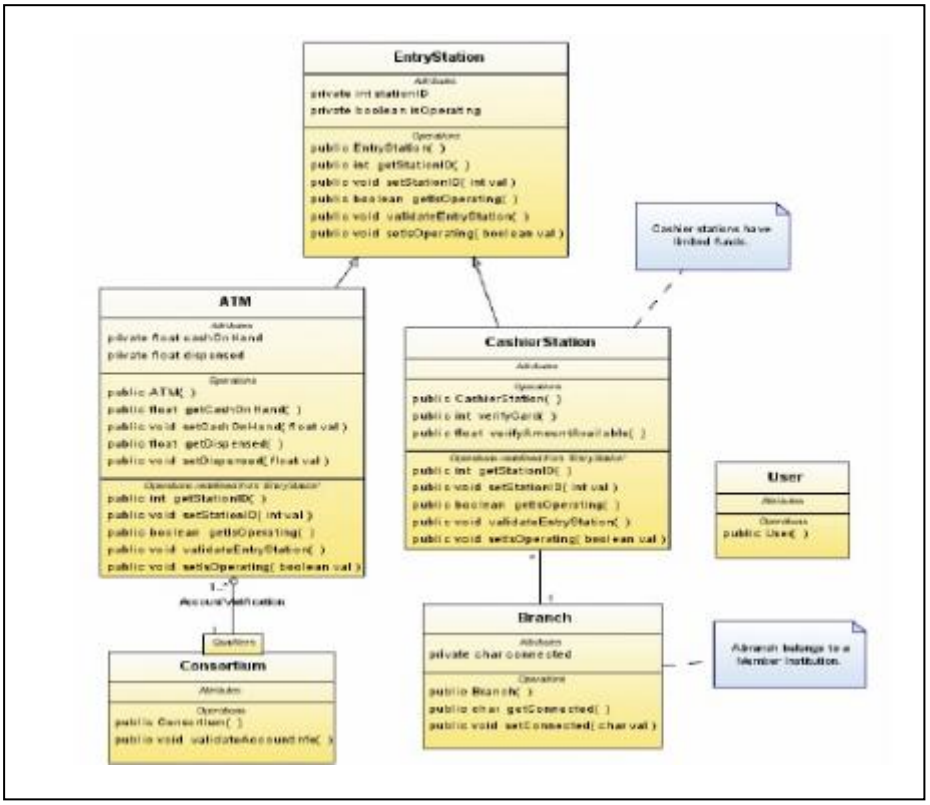
   Activity diagram adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan jalur kerja. Dalam beberapa hal, diagram ini memainkan peran mirip sebuah diagram alir, tetapi perbedaan prinsip antara diagram ini dan notasi diagram alir adalah diagram ini mendukung behavior parallel.



Gambar 2.5. Contoh Activity Diagram

3. Class Diagram

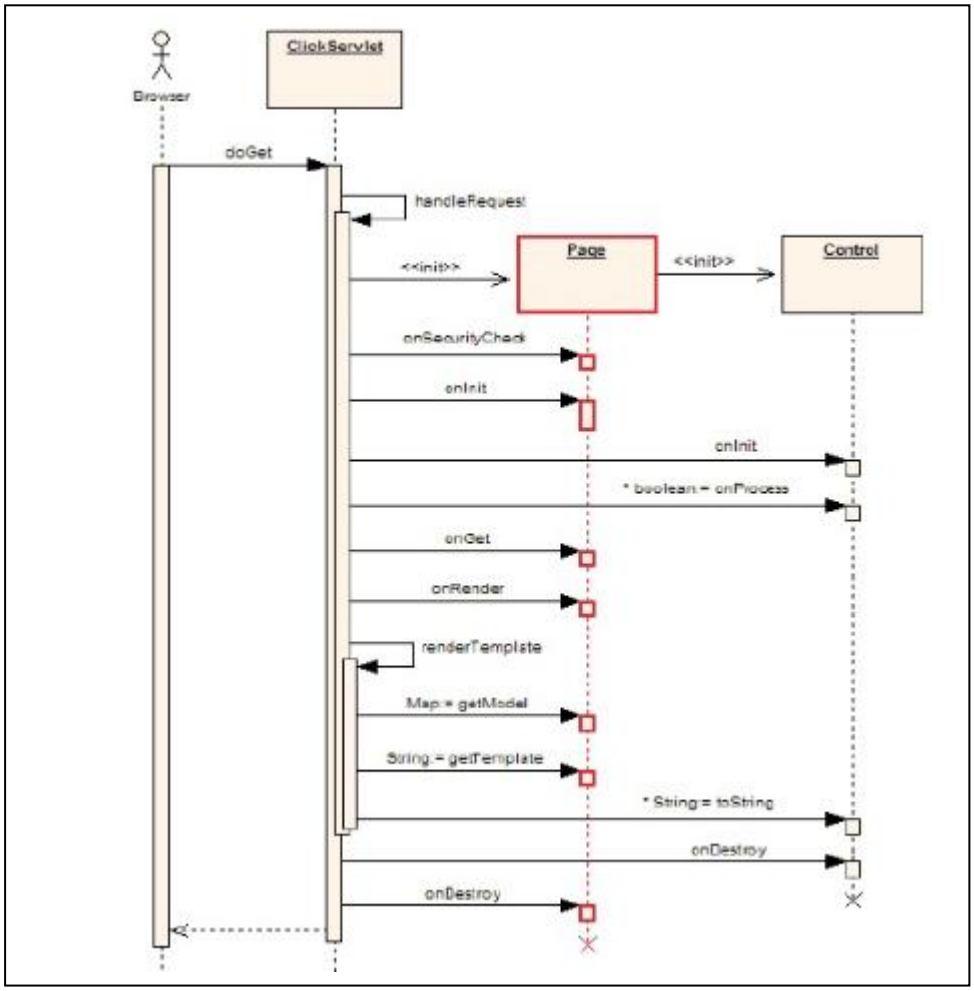
Kelas diagram adalah representasi visual dari sebuah aplikasi yang menunjukkan class dan hubungan antar class. Class diagram juga mendeskripsikan jenis-jenis objek, properti dan operasi sebuah class dan batasan-batasan yang terdapat dalam hubungan objek tersebut



Gambar 2.6. Contoh Class Diagram

4. Sequence Diagram

Sequence diagram secara khusus menjabarkan behavior sebuah skenario tunggal. Diagram ini tersebut menunjukkan sejumlah objek contoh dan pesan-pesan yang melewati objek- objek ini di dalam use case.



Gambar 2.7 Contoh Sequence Diagram

2.8 Desain Sistem

Desain sistem adalah tahapan pertama dalam perancangan, disini keputusan pemecahan masalah tingkat tinggi. Selama desain sistem pengembangan membuat keputusan strategi untuk software secara keseluruhan. Pada tahapan ini diformulasikan arsitektur untuk sistem dan memilih strategi global serta kebijakan untuk tahapan selanjutnya yaitu tahapan desain rinci.

2.8.1 Membuat Arsitektur

Prinsip-prinsip arsitektur adalah sebagai berikut:

1. Bedakan antara aplikasi untuk oprasinal dan aplikasi pendukung keputusan.
2. Dekomposisi sistem yang besar kedalam lapisan-lapisan atau bagian-bagian.
3. Pisahkan aplikasi logika dengan antar muka pemakai.
4. Pertimbangkan *reification* (sesuatu yang abstrak sebagai sebuah objek)
5. Substitusi query dengan bahasa pemograman.
6. Pertimbangkan antar muka utama pengguna dan sistem lain.

2.8.2 Memilih Pengendalian Eksternal

Tahapan berikutnya adalah menentukan bagaimana aplikasi dikendalikan, khususnya ketika berintraksi dengan pengguna. Terdapat empat platform pengendalian yang dapat dipilih yaitu:

1. Procedure-driven control
2. Event-driven control
3. Concurrent control
4. Declarative control

2.8.3 Memilih Pendekatan Manajemen Data

Beberapa alternatif manajemen data adalah menggunakan memori, file, dan DBMS. Beberapa aplikasi membutuhkan pelayanan database formal, yang lain cukup mengunakan memori dan file yang lebih sederhana dan lebih murah. Beberapa aspek yang perlu dipertimbangkan dalam pemilihan manajemen data adalah:

1. Kekutan data
2. Biaya pemeblian
3. Biaya siklus hidup
4. Ukuran data
5. Kinerja
6. Ekstensibilitas
7. Akses simultan
8. Crash recovery
9. Integritas

10. Dukungan transaksi
11. Distribusi
12. Bahasa query
13. Keamanan
14. Metadata

Beberapa aspek yang perlu diperhatikan dalam penggunaan DBMS adalah:

1. Ukuran tempat penyimpanan
2. Waktu respon
3. Perbaikan kinerja
4. Beban administrasi
5. Model penguncian

2.8.4 Memilih Paradigma Database

Ada beberapa tipe database yang bias dipilih yaitu hirarical database, network database, relational database, dan object-oriented database. Dua tipe pertama sudah dianggap kadaluarsa, maka penjelasan tipe relational dan object-oriented adalah sebagai berikut:

2.8.4.1 Relational Database

Dalam database relasional data digambarkan secara logis dalam bentuk table. RDBMS mengelola table data dan truktur asisiasi yang mampu meningkatkan fungsi dan performasi table tersebut. Tiga aspek utama RDBMS adalah:

1. Data dipresentasikan dalam table
2. Operator digunakan untuk manipulasi table
3. Pembatas

Kelemahan RDBMS adalah:

1. Navigasi yang lambat
2. Kurangnya fitur
3. Protocol penguncian yang tidak fleksibel
4. Jenis data sedikit

5. Table sebagai satu-satunya sebagai paradigm
6. Tidak familiar bagi progremer

Keungulan RDBMS adalah:

1. Teori standar
2. Ketersediaan
3. Ekstensibilitas
4. Akses data yang deklaratif
5. Kamus data
6. Query yang cepat

Aplkasi yang cocok adala:

1. Aplikasi pendukung keputusan
2. Aplikasi bisnis biasa
3. Aplikasi yang mengunakan 4GL
4. Aplikasi terintegrasi
5. Implementasi yang konservatif

2.8.4.2 Object-Oriented Database

OO_DBMS adalah tempat penyimpanan yang tangguh yang diciptakan oleh bahasa pemograman berorientasi objek. OO_DBMS mengelola data, kode pemograman, dan struktur asosiasi yang membentuk database berorientasi objek. OO_DBMS diciptakan akibat motivasi ketidak effesienan RDBMS, yang terlalu lambat untuk query yang menelususri objek ke objek, tipe data yang terlalu sederhana, dan kelemahan mengkombinasikan query dengan bahasa pemograman, lebih cepat, mendukung pewarisan, identitas objek, dan fitur canggih lainnya.

Kelemahan OO_DBMS adalah:

1. Teori belum lengkap dan belum standar
2. Kemungkinanan kerusakan database
3. Kurang ekstensibilitas logika

Keunggulan OO_DBMS adalah:

1. Nafigasi yang cepat
2. Fitur canggih
3. Protokol penguncian yang fleksibel
4. Tipe sistem yang seragam

Aplikasi yang cocok untuk OO_DBMS adalah:

1. Aplikasi perancangan teknik
2. Aplikasi mulimedia
3. Knowledge bases
4. Aplikasi yang membutuhkan distribusi dan kesimultanan
5. Aplikasi yang membutuhkan fitur yang canggih
6. Peralatan elektronik dengan software tertanam

2.8.5 Memilih Startegi Interaksi Data

Terdapat beberapa strategi untuk menggabungkan kapabilitas DBMS kedalam sebuah aplikasi:

1. Batch preprocessor dan postpricessor, query database untuk mendapatkan file input, jalankan apliasi, kemudian analisis output dan simpan hasil kedatabase.
2. Script file, adalah script yang digunakan untuk menjalankan perintah untuk menginplementasikan suatu fungsi. Biasanya digunakan untuk interaksi database sederhana.
3. Embeddad statis SQL.
4. Custom application progreming interface, yaitu melakukan pengkapsulan permintaan membaca dan menulis database kedalam sedikit metode aplikasi.
5. Methods stored in the database. Metode-metode yang disimpan sangat efisien dan menyediakan lebih banyak kesmpatan pengguna ulang.
6. 4GL, dengan 4GL mengurangi waktu pengembangan aplikasi, tapi tidak cocok untuk aplikasi yang membutuhkan pemograman kompleks.

2.8.6 Memilih Pendekatan untuk Identitas Data

Terdapat dua pilihan untuk menentukan identitas data yaitu:

1. Exixtence-based identity, adalah pembuatan identitas oleh sistem secara otomatis ketika objek tercatat kedalam database, identitas ini efesien, seragam, dan berukuran kecil, tapi menyulitkan inspeksi dan menyulitkan untuk database yang tidak memiliki kemampuan semantik.
2. Value-based identity, adalah identitas yang dibuat berdasarkan kombinasi atribut dari setiap objek, memudahkan pengguna mengidentifikasi data, memfasilitasi perbaikan database, dan memudahkan distribusi database. Tapi sulit untuk diubah.

2.8.7 Memilih Kebijakan Default

Selama desain sistem perlu ditetapkan kebijakan default yang meningkatkan keseragaman dan kesederhanaan pada tahapan berikutnya, diantaranya:

1. Asosiasi, gunakan pendekatan dasar untuk membuat aplikasi dan gunakan secara konsisten pada aplikasi.
2. Null, tetapkan kebijakan umum untuk nilai null untuk setiap atribut.
3. Nama atribut, nama atribut harus bersifat unik secara global atau secara local.
4. Nama peran, intepretasikan nama peran sebagai nama acuan atau imbuhan bagi primary keys.
5. Data lanjutan, data lanjutan biasa hasil komputasi atau evaluasi.

2.8.8 Transformasi Model Objek

Transformasi adalah pemetaan dari domain model objek dengan berbagai model objek. Menerapkan serangkaian transformasi, dapat menyederhanakan dan mengoptimalkan model.

2.8.9 Elaborasi Model Objek

Harus menangani masalah-masalah tambahan selama desain rinci:

1. Tambahkan candidate key. Harus meninjau model objek dan menambahkan kunci kandidat untuk kombinasi atribut yang unik.
2. Tetapkan domain. Selama desain menetapkan domain untuk setiap atribut, selama pelaksanaan dapat menentukan tipe data untuk setiap domain. Domain

mempromosikan konsistensi antara atribut dan mengurangi pelaksanaan keputusan.

3. Tentukan nulls. Konsisten dengan kebijakan umum Anda dari desain sistem, Anda harus menetapkan diperbolehkannya nulls atau nilai-nilai default untuk setiap atribut.
4. Perkiraan penyimpanan fisik. Untuk setiap kelas dan asosiasi harus memperkirakan jumlah kejadian, laju pertumbuhan, dan periode retensi.

2.8.10 Elaborasi Model Fungsional

Untuk menguji use case yang telah dianalisis apakah sudah mewakili dalam penjelasan dari aliran sistem informasi pengadaan barang,

2.9 Implementasi

Pada tahap implementasi ini merupakan penerapan aplikasi dari hasil perancangan sistem yang ada untuk mencapai suatu tujuan yang diinginkan. Implementasi melaksanakan perintah-perintah yang secara terstruktur dari awal sampai akhir.

2.10 Konsep Dasar Java

2.10.1 Sejarah Singkat Bahasa Pemrograman Java

Java secara resmi diperkenalkan oleh SUN pada dunia pada tanggal 23 Mei 1995. Sedangkan sejarah pembangunan Java sendiri sudah dimulai sejak tahun 1991. Saat itu tim “Stealth Project” mengadakan pertemuan (brainstorming) untuk menciptakan suatu sistem software yang mampu berjalan pada alat-alat elektronik (small devices).

James Gosling berkonsentrasi pada ide pembuatan bahasa pemrograman. Pada Juni 1991, muncullah bahasa interpreter “Oak” yang menjadi cikal bakal dari Java. Kemudian secara resmi pada tahun 1995 Java diperkenalkan bersama dengan browser HotJava, Java pun merambah ke dunia web.

Kenyataan ini mungkin agak sedikit berbeda dengan ide pembuatan Java pada awalnya. Internet ternyata membantu menjadikan Java terkenal seperti sekarang

ini. Memang harus diakui karena semakin berkembangnya Internet, maka fokus pemograman saat ini mengarah ke pemograman Internet itu sendiri.

Saat ini Java dibagi menjadi 3 macam framework atau teknologi yaitu J2SE untuk pemogramman aplikasi berbasis console dan destop, kemudian J2EE untuk pemogramman aplikasi berskala enterprise seperti aplikasi web-base (JSP dan Servlet), komponen (EJB), Web Services dan lain – lain. Kemudian framework yang terakhir adalah J2ME untuk pemogramman small divice seperti hanphone dan pda.

2.10.2 Kelebihan dan Karakteristik Java

Sintaks bahasa pemogramman Java adalah pengembangan dari bahasa pemogramman C/C++. Sehingga bagi mereka yang sudah terbiasa dengan C/C++, tidak akan mengalami kesulitan mempelajari bahasa pemogramman Java.

Java adalah bahasa pemogramman yang sederhana dan tangguh. Berikut ini adalah beberapa karakteristik dari Java sesuai dengan white paper dari Sun.

1. Berorientasi Object, java telah menerapkan konsep pemograman berorientasi object yang modern dalam implementasinya.
2. Robust, java mendorong pemograman yang bebas dari kesalahan dengan bersifat strongly typed dan memiliki run-time checking.
3. Protable, program java dapat berjalan pada sistem operasi apapun yang miliki Java Virtual Machine.
4. Multithreading, Java mendukung pemograman multithreading dan telah terintegrasi secara langsung dalam bahasa Java.
5. Dinamis, program Java dapat melakukan sesuatu tindakan yang ditentukan pada saat eksekusi program dan bukan pada saat kompilasi.
6. Sederhana, Java menggunakan bahasa yang sederhana dan mudah dipelajari.
7. Terdistribusi, Java didesain untuk berjalan pada lingkungan yang terdistribusi seperti halnya internet.
8. Aman, aplikasi yang dibuat dengan bahasa java lebih dapat dijamin keamanannya terutama untuk aplikasi internet.

9. Netral secara arsitektur, Java tidak terikat pada suatu mesin atau sistem operasi tertentu.

2.11 Kelebihan dan Karakteristik Java

Membuat program dengan bahasa pemrograman Java memerlukan beberapa perangkat bantu (tool). Perangkat bantu ini digunakan untuk menulis kode sumber, menguji, dan mendebug program yang dibuat. Java Developer's Kit (JDK) adalah kumpulan tool yang dapat diunduh (didownload) secara gratis dari situs Sun Mycrosistem. Setidaknya ada 7 program bantu (tool) yang tersedia pada Java Developer's Kit (JDK) seperti yang dijelaskan dalam tabel berikut ini :

Tabel 2.2 Penjelasan Program Bantu Java

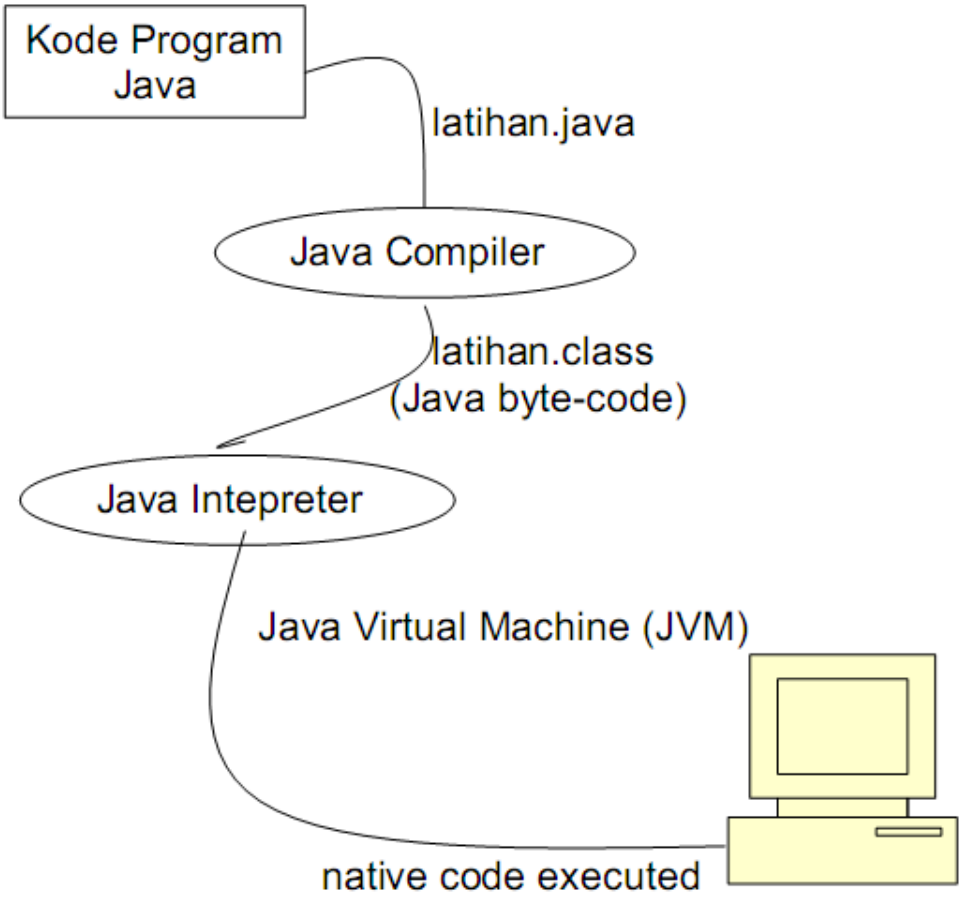
Nama Program Bantu	Penjelasan
Javac	Kompilator Java untuk mengubah file sumber ke dalam file yang dapat dieksekusi. File ini disebut file byte code dan memiliki ekstensi class.
Java	Intepreter Java untuk mengeksekusi file kelas (class) atau file byte code.
Jdb	Debugger Java, yang bekerja seperti interpreter mengeksekusi kelas Java dan juga memiliki kemampuan untuk menghentikan eksekusi program pada titik-titik henti terpilih dan untuk menampilkan nilai-nilai variabel kelas. Kemampuan ini penting untuk mencari kesalahan program.
Javap	Disassembler Java, menerima file byte code dan menampilkan kelas, variabel dan metode yang telah dikompilasi menjadi byte code. Program bantu ini juga mengidentifikasi instruksi yang digunakan untuk mengimplementasikan tiap-tiap metode. Anda dapat menggunakan program bantu ini untuk mendapatkan kode sumber dari kelas Java yang sudah terkom-pilasi.

Appletviewer	Appletviewer adalah alat bantu (tool) untu menampilkan applet Java yang berada di dalam web pada sistem lokal maupun pada situs web yang dapat diakses. Program bantu (tool) ini dapat digunakan untuk menguji applet yang anda buat.
Javadoc	Program bantu dokumentasi otomatis yang digunakan untuk mengkonversi kode sumbe Java ke dalam file Hypertext Markup Language (HTML).
javah	Program bantu file header C, digunakan untuk membangkitkan header bahasa C. File yang dibangunkan dipakai untuk mengembangkan kelas Java yang dibuat dari bahasa selain Java (bahasa C).

2.12 Mekanisme Kompilasi Dan Eksekusi Aplikasi Java

Untuk membuat kode program bahasa Java dapat menggunakan text editor seperti notepad (Microsoft Windows) atau vi (Linux). Kode program Java harus disimpan dalam file dengan ekstensi java, misalnya hello.java, latihan.java dan sebagainya. Perlu diperhatikan program Java memiliki case sensitive, artinya antara huruf besar dan huruf kecil dibedakan. Jadi harus berhati-hati dalam membuat sebuah program Java.

Setelah itu program java yang sudah disimpan di dalam file harus dikompilasi dengan javac yang telah ada di dalam Java Development Kit (JDK). Hasil dari kompilasi kode sumber program Java adalah file berekstensi .class yang merupakan byte-code Java yang berupa kode-kode mesin yang dihasilkan Java Virtual Machine (JVM), kemudian JVMlah yang akan mengintepretasikan kode-kode tersebut ke kode native atau kode-kode mesin dari arsitektur yang bersangkutan. Sehingga program Java dapat dijalankan (dieksekusi) di semua platform, baik Sun Solaris, Windows 9x/NT, Linux atau sistem operasi lain yang mendukung Java Platform.



Gambar 2.8 MekanismeAplikasi Java

2.13 MySQL

MySQL merupakan sebuah sistem manajemen database open source yang populer dan gratis untuk platform UNIX. Sistem manajemen database MySQL menggunakan kumpulan perintah sederhana untuk memasukkan, memanggil, menghapus, dan memperbarui data, sehingga dapat mengembangkan database yang kompleks.

MySQL menjadi solusi yang tepat dalam aplikasi *database*. Kemampuan MySQL yang dapat digunakan oleh berbagai sistem operasi menjadikan MySQL *software*

pilihan terlebih ada dukungan komunitas, *mailing list* dan *homepage* khusus yang menyediakan *tutorial* serta dokumentasi lengkap.

Beberapa kemampuan MySQL adalah sebagai berikut:

1. MySQL bisa diakses dan dimanipulasi dari sejumlah bahasa pemrograman terkenal, diantaranya adalah C, C++, Java, Perl, Phyton, dan PHP.
2. MySQL mendukung tipe data yang umum digunakan, termasuk FLOAT, DATETIME, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, SET, dan ENUM.
3. MySQL bisa diperoleh secara gratis termasuk aplikasi-aplikasi lain yang diperlukan dalam memakai MySQL.
4. MySQL merupakan *database* yang dapat menangani beberapa *user* sekaligus dalam waktu bersamaan tanpa mengalami konflik atau masalah, sehingga memungkinkan diakses oleh banyak *client* dalam waktu bersamaan.
5. Sekuritas merupakan hal penting dalam MySQL terbukti dengan beberapa lapisan *security* seperti nama *host*, ijin akses *user* dengan sistem perijinan mendetail dan *password* yang terenkripsi.

2.14 PhpMyAdmin

PhpMyAdmin adalah sebuah sistem administrasi MySQL berbasis web yang dibuat dengan PHP oleh Tobias Ratschiller, PhpMyAdmin ditujukan untuk menangani administrasi database pada MySQL lewat interface. Hampir semua query dari MySQL dapat dilakukan menggunakan PhpMyAdmin. PhpMyAdmin

memudahkan pengolahan database MySQL tanpa harus berhadapan dengan text mode MySQL.

Kemudahan dalam pengolahan database dengan interface web yang familiar menjadi kelebihan tersendiri PhpMyAdmin. Sampai saat ini PhpMyAdmin terus dikembangkan dengan penambahan skin (model tampilan), penambahan bahasa, dan pengurangan bug (kesalahan dalam program yang mengakibatkan jalannya program menjadi kurang baik).