BAB II

TINJAUAN PUSTAKA

2.1 Profil Tempat Kerja Praktek

2.1.1 Sejarah PPPPTK-BMTI Bandung

Pada Pelita I (1969/1970 s.d 1974/1975), Pemerintah Republik Indonesia menempatkan pembangunan pendidikan teknologi sebagai bagian integral REPELITA mengisi kebutuhan terhadap tenaga kerja teknik. Sebelum Pelita I dimulai Direktur Pendidikan Teknologi, Kolonel Amir Gondokusuma, telah melakukan analisis kebutuhan, analisis jabatan, hingga analisis kemampuan,yang kemudian dijabarkan dalam bentuk Kurikulum STM Pembangunan.

Tahun pertama Pelita I dimulai dengan pembangunan delapan STM Pembangunan, dengan dukungan sumber daya yang dimiliki oleh Indonesia sendiri. Tahun kedua Pelita I (1970-1971), pembangunan pendidikan teknik ditingkatkan lagi dengan membangun lima Tehcnical Training Centre (Balai Latihan Pendidikan Teknik) dengan pinjaman dana dari World Bank, dan bantuan tenaga ahli dari UNESCO serta Pemerintah Inggris. Tahun keempat Pelita I (1972-1973), diadakan proyek Peningkatan Mutu Pengajaran Teknik (PMPT), dengan pusat penyelenggaraan di STM Instruktor (bekas SGPT) di Jalan Dr. Rum No. 9 Bandung, dengan sasaran utama mendukung peningkatan mutu guru teknik pada proyek-proyek STM Pembangunan dan BLPT.

Sejalan dengan perkembangan yang semakin intensif pembangunan pendidikan teknik, antara lain dengan penambahan BLPT menjadi sembilan atas bantuan World Bank dan rehabilitasi 27 STM atas bantuan pinjaman dari Pemerintah Belanda maka dirasakan perlunya pelembagaan proyek-proyek penataran guru teknik. Melalui bantuan tenaga ahli dari Australia Mr. Ian Scoot tahun 1972-1973, dan Mr. Ken Sharp tahun 1974-

1975, dirumuskan suatu bentuk kelembagaan, yang waktu itu disebut TTUC (Tehnical Teacher Upgrading Centre).

Sejak tahun 1975/1976 kegiatan-kegiatan penataran telah dimulai dioperasikan secara melembaga oleh TTUC dalam status keproyekan.Pengakuan terhadap fungsi dan peranan pendidikan teknologi sebagai bagian integral program pembanguan nasional (REPELITA) mulai diperoleh pada masa jabatan Direktur Pendidikan Menengah Kejuruan, Drs. Soenaryo, M.Sc. tahun 1972-1979 yang pada saat itu juga sistem pendidikan menengah kejuruan dirumuskan secara konsepsional.

Seiring dari dampak positip dari peran TTUC terhadap pengembangan kemampuan guru maka pada tahun 1978 diterbitkan Surat Keputusan Nomor 0205/O/1978 tentang Pelembagaan Unit Pelaksana Teknis (UPT) Pusat Pengembangan Penataran Guru Teknologi (PPPG Teknologi) Bandung bersamaan dengan PPPG lain dan SK Pengangkatan Kepala Pusat pertama, Drs. Jorlin Pakpahan, sedangkan operasional PPPG Teknologi Bandung secara efektif dimulai pada tahun 1980. Direktur Dikmenjur Ir. Hadiwaratama M.Sc (1981-1983), telah berupaya melakukan penyempurnaan peran TTUC menjadi "Pusat Pengembangan Pendidikan Teknologi" yang selanjutnya dikenal dikenal juga dengan sebutan Technical Education Development Centre (TEDC). Adanya program penataran guru STM dalam bentuk Program Diploma III Guru Teknologi (bulan Juli 1982) dan dengan misi "menciptakan guru STM menjadi guru profesional" sebagai bagian dari upaya memantapkan organisasi dan manajemen, PPPG Teknologi Bandung juga dituntut untuk menyajikan program dan manajemen yang berkualitas profesional.Pada tahun 1986, kampus PPPG Teknologi Bandung dipindahkan dari Jalan Dr. Rum Bandung ke Cimahi, dalam kerangka mengembangkan fungsi sebagai Pusat Pengembangan Pendidikan Teknologi.

Tahun 1990 dengan Surat Keputusan Mendikbud Nomor 0529/O/1990, tanggal 14 Agustus 1990, diadakan perampingan organisasi sekaligus

diikuti dengan kebijakan strategis yakni pengakuan pengakuan terhadap fungsi dan peranan PPPG Teknologi Bandung sebagai "Pusat Pengembangan Pendidikan", yang berarti semakin terbukanya peluang PPPG Teknologi Bandung melakukan program pengembangan sebagai salah satu sub-sistem yang sangat vital dalam pembangunan pendidikan menengah teknologi di Indonesia.

Peranan dan fungsi PPPG Teknologi Bandung makin diakui keberadaanya, baik ditingkat nasional maupun ditingkat internasional dan untuk lebih meningkatkan peran dan fungsinya sebagai lembaga pendidikan di tanah air, maka pada tahun 2007, Pusat Pengembangan Penataran Guru Teknologi Bandung sesuai dengan Peraturan Menteri Pendidikan Nasional No. 8 Tahun 2007 namanya menjadi Pusat Pengembangan dan Pemberdayaan Pendidik dan Tenaga Kependidikan (PPPPTK) bidang Mesin dan Teknik Industri.

Dengan demikian semakin terbukanya peluang PPPPTK Bidang Mesin dan Teknik Industri melakukan program pengembangan sebagai salah satu sub sistem yang sangat vital dalam pembangunan pendidikan di Indonesia.

Visi :

Terpercaya dan berstandar internasional dalam layanan pembentukan insan profesional.

Misi :

1. Meningkatkan layanan fasilitasi pengembangan dan pemberdayaan PTK.
2. Memperluas kerjasama dengan DU-DI dalam pengembangan dan inovasi pendidikan teknologi.
3. Mewujudkan kerjasama dalam layanan otonomi satuan pendidikan & MBS.
4. Melaksanakan penjaminan mutu diklat PTK serta masyarakat secara efisien dan efektif.

Tugas :

PPPPTK Bidang Mesin dan Teknik Industri Bandung mempunyai tugas melaksanakan pengembangan dan pemberdayaan pendidik dan tenaga kependidikan Bidang Mesin dan Teknik Industri.

Fungsi :

Untuk menyelenggarakan tugas tersebut, PPPPTK Bidang Mesin dan Teknik Industri Bandung mempunyai fungsi:

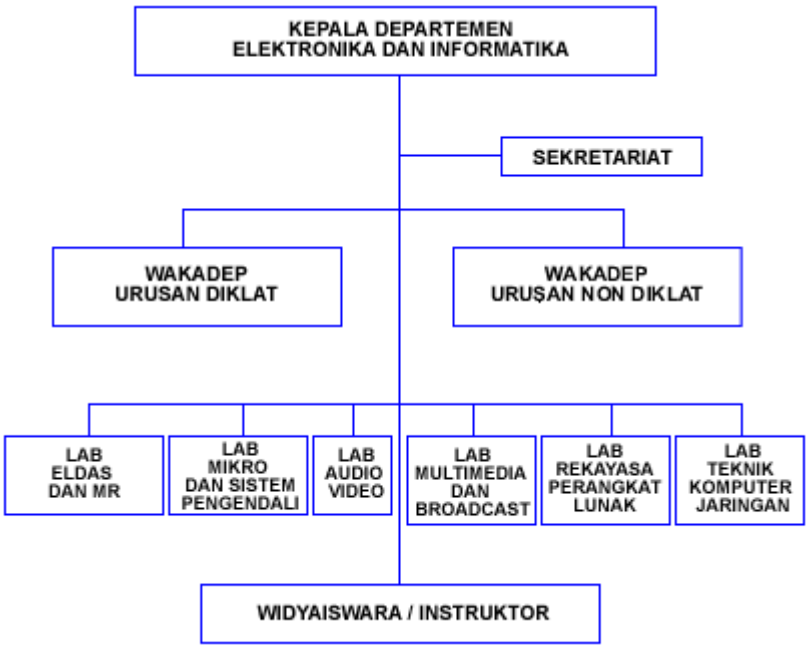
1. Penyusunan program pengembangan dan pemberdayaan pendidik dan tenaga kependidikan.
2. Pengelolaan data dan informasi peningkatan kompetensi pendidik dan tenaga kependidikan.
3. Fasilitasi dan pelaksanaan peningkatan kompetensi pendidik dan tenaga kependidikan
4. Evaluasi program dan fasilitasi peningkatan kompetensi pendidik dan tenaga kependidikan.
5. Pelaksanaan urusan administrasi PPPPTK.

2.1.2 Logo Instansi



Gambar 2.1 Logo Instansi

2.1.3 Struktur Organisasi dan Job Description



Adapun gambaran tugas dari masing-masing jabatan adalah sebagai berikut :

1. Kepala Departemen Elektronika dan Informatika bertugas, berwenang, dan bertanggung jawab antara lain :
   a. Melakukan perencanaan dan pengembangan SDM
   b. Melakukan evaluasi dan penyusunan organisasi sesuai tuntutan internal maupun eksternal
   c. Menyusun program kerja
   d. Melaporkan semua kegiatan dan hasilnya serta memberikan saran atau usul kepada Kepala Instansi.
2. Sekretariat bertugas, berwenang, dan bertanggung jawab antara lain :
   a. Menulis seluruh laporan kegiatan kerja kepada Kepala
   b. Menyusun daftar Diklat
   c. Membantu Kepala Departemen dalam pembuatan laporan
3. Wakadep Urusan Diklat bertugas, berwenang, dan bertanggung jawab antara lain :
   a. Membantu Kepala Departemen dalam urusan Diklat
   b. Mengatur jadwal Diklat
   c. Mengawasi kegiatan Diklat
4. Wakadep Urusan Non Diklat bertugas, berwenang, dan bertanggung jawab antara lain :
   a. Membantu Kepala Departemen menyusun program kerja
   b. Membantu Kepala Departemen dalam pengembangan SDM
5. Kepala Lab. Eldas dan MR bertugas, berwenang, dan bertanggung jawab antara lain :
   a. Pengembangan system di bidangnya
   b. Bertanggung jawab secara penuh terhadap segala kegiatan yang ada di Lab. Eldas dan MR
   c. Memilih pengajar untuk program Diklat
   d. Mengawasi kegiatan Diklat di bidang Eldas dan MR
   e. Membuat laporan kegiatan Diklat di bidang Eldas dan MR untuk kemudian diserahkan kepada Kepala Departemen

6. Kepala Lab. Mikro dan Sistem Pengendali
    a. Pengembangan system di bidangnya
    b. Bertanggung jawab secara penuh terhadap segala kegiataan yang ada di Lab. Mikro dan Sistem Pengendali
    c. Memilih pengajar untuk program Diklat
    d. Mengawasi kegiatan Diklat di bidang Mikro dan Sistem Pengendali
7. Kepala Lab. Audio dan Video
    a. Pengembangan system di bidangnya
    b. Bertanggung jawab secara penuh terhadap segala kegiataan yang ada di Lab. Audio dan Video
    c. Memilih pengajar untuk program Diklat
    d. Mengawasi kegiatan Diklat di bidang Audio dan Video
8. Kepala Lab. Multimedia dan Broadcast
    a. Pengembangan system di bidangnya
    b. Bertanggung jawab secara penuh terhadap segala kegiataan yang ada di Lab. Multimedia dan Broadcast
    c. Memilih pengajar untuk program Diklat
    d. Mengawasi kegiatan Diklat di bidang Multimedia dan Broadcast
9. Kepala Lab. Rekayasa Perangkat Lunak
    a. Pengembangan system di bidangnya
    b. Bertanggung jawab secara penuh terhadap segala kegiataan yang ada di Lab. Rekayasa Perangkat Lunak
    c. Memilih pengajar untuk program Diklat
    d. Mengawasi kegiatan Diklat di bidang Rekayasa Perangkat Lunak
10. Kepala Lab. Teknik Komputer Jaringan
    a. Pengembangan system di bidangnya
    b. Bertanggung jawab secara penuh terhadap segala kegiataan yang ada di Lab. Teknik Komputer Jaringan
    c. Memilih pengajar untuk program Diklat
    d. Mengawasi kegiatan Diklat di bidang Teknik Komputer Jaringan
11. Widyaiswara / Instruktor

a. Pengajar dalam kegiatan Diklat
b. Memberikan materi kepada peserta Diklat sesuai bidangnya
c. Membuat laporan hasil evaluasi Diklat
d. Menyerahkan hasil laporan kepada masing-masing Kepala Laboratorium

2.2 Landasan Teori

2.2.1 Analisa Sistem

2.2.1.1 Pengertian Sistem

Sistem berasal dari bahasa Latin (systēma) dan bahasa Yunani (sustēma) adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi. Istilah ini sering dipergunakan untuk menggambarkan suatu set entitas yang berinteraksi, di mana suatu model matematika seringkali bisa dibuat.

Pada prinsipnya, setiap sistem selalu terdiri atas empat elemen :

a. Objek, yang dapat berupa bagian, elemen, ataupun variabel. Ia dapat benda fisik, abstrak, ataupun keduanya sekaligus tergantung kepada sifat sistem tersebut
b. Atribut, yang menentukan kualitas atau sifat kepemilikan sistem dan objeknya
c. Hubungan internal, di antara objek-objek di dalamnya
d. Lingkungan, tempat di mana sistem berada.

2.2.1.2 Pengertian Sistem Informasi

Sistem informasi adalah sekumpulan komponen pembentuk sistem yang mempunyai keterkaitan antara satu komponen dengan komponen lainnya yang bertujuan menghasilkan suatu informasi dalam suatu bidang tertentu. Dalam sistem informasi diperlukannya klasifikasi alur informasi, hal ini disebabkan keanekaragaman kebutuhan akan suatu informasi oleh pengguna informasi. Kriteria dari sistem informasi antara lain, fleksibel, efektif dan efisien.

Menurut Kertahadi (dalam Fatta, 2007) sistem informasi adalah suatu alat untuk menyajikan informasi sedemikian rupa sehingga bermanfaat bagi penerimanya. Tujuannya adalah untuk menyajikan informasi guna pengambilan keputusan pada perencanaan, pemrakarsaan, pengorganisasian, pengendalian kegiatan operasi suatu perusahaan yang menyajikan sinergi organisasi pada proses (Murdick & Ross, dalam Fatta 2007).

2.2.1.3 Pengertian Inventori

Inventory meliputi semua barang yang dimiliki perusahaan atau instansi pada saat tertentu, dengan tujuan untuk dijual kembali atau dikonsumsikan dalam siklus operasi normal perusahaan atau instansi sebagai barang yang dimiliki untuk dijual atau diasumsikan untuk dimasa yang akan datang, semua barang yang berwujud dapat disebut sebagai inventory, tergantung dari sifat dan jenis usaha perusahaan.

2.2.1.4 Pengertian Aplikasi

Aplikasi merupakan penerapan, menyimpan hal, data, permasalahan pekerjaan suatu sarana atau media yang digunakan untuk menerapkan atau mengimplikasikan hal atau permasalahan tersebut, sehingga berubah menjadi suatu bentuk yang baru tanpa menghilangkan nilai-nilai dasar dari hal, data, permasalahan atau pekerjaan. Jadi dalam hal ini hanya bentuk dari tampilan data yang berubah, sedangkan isi yang termuat dalam data tersebut tidak mengalami perubahan.

Jadi program aplikasi adalah sekumpulan instruksi yang tersusun menurut urutan yang logis dan dikenal oleh komputer atau program yang dipakai untuk melakukan atau mempermudah suatu pekerjaan.

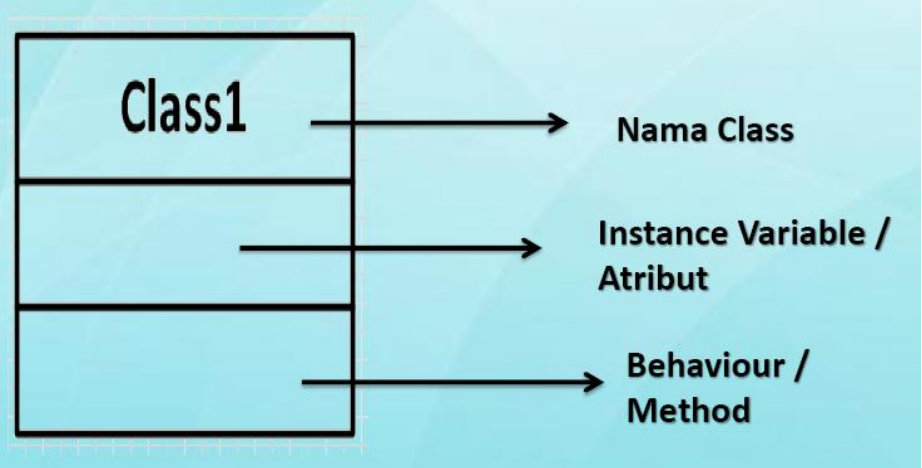
2.2.2 Object Oriented Programming (OOP)

Object-Oriented Programming (OOP) adalah sebuah pendekatan untuk pengembangan / development suatu software dimana dalam struktur software tersebut didasarkan kepada interaksi object dalam penyelesaian suatu proses/tugas. Interaksi tersebut mengambil form dari pesan-pesan dan mengirimkannya kembali antar object tersebut, object adalah sebuah structure yang menggabungkan data dan prosedur untuk bekerja bersama-sama. Object akan merespon pesan tersebut menjadi sebuah tindakan /action atau metode.

Dengan OOP, kita dapat mengimplementasikan object data yang tidak hanya memiliki ciri khas (attribut), melainkan juga memiliki metode untuk memanipulasi attribut tersebut. Singkatnya, OOP memiliki keunggulan dari konsep pemrograman terstruktur, selain itu juga memiliki kemampuan untuk mengimplementasikan objek dalam kehidupan nyata.

Suatu program disebut dengan pemrograman berbasis obyek (OOP) karena terdapat :

1. kelas: kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi object. Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak menggunakan OOP). Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.



Gambar 2.9 Gambaran Class

2. Objek: membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer. Objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.
3. Abstraksi: Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.
4. Enkapsulasi: Memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak. Hanya metode dalam objek tersebut yang diberi ijin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.
5. Polimorfisme melalui pengiriman pesan. Tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan;

metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesa tersebut dikirim.

6. Inheritas: Mengatur polimorfisme dan enkapsulasi dengan mengijinkan objek didefinisikan dan diciptakan dengan jenis khusus dari objek yang sudah ada – objek-objek ini dapat membagi (dan memperluas) perilaku mereka tanpa haru mengimplementasi ulang perilaku tersebut (bahasa berbasis-objek tidak selalu memiliki inheritas.)

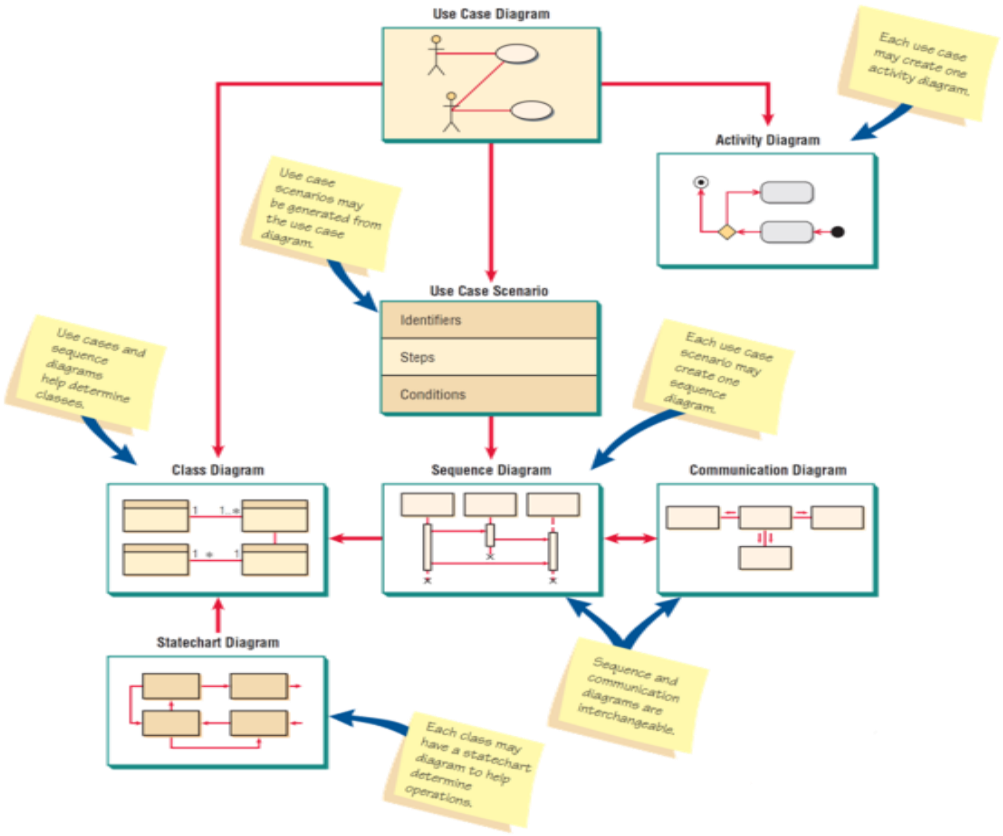
Dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut.

2.2.3 Perangkat Analisa System

2.2.3.1 Unified Modeling Language (UML)

UML (Unified Modeling Language) adalah sebuah bahasa untuk menetukan, visualisasi, kontruksi, dan mendokumentasikan artifact (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak. Artifact dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak. UML sendiri juga memberikan standar penulisan sebuah sistem blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem software.

Diagram-diagram yang ada dalam UML dapat dikelompokan menjadi 2 kelompok besar, yaitu diagram yang menggambarkan model struktur statis dari sistem (model statis) dan digram yang menggambarkan struktur dinamis dari sistem (model dinamis).



Gambar 2.4 Relasi antar Diagram UML

1. Use Case Diagram

   Use case adalah abstraksi dari interaksi antara sistem dan actor. Use case bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. *Use case* merupakan konstruksi untuk mendeskripsikan bagaimana sistem akan terlihat di mata user. Sedangkan use case diagram memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan client.

2. Use Case Scenario

   Use case skenario merupakan hasil instansiasi dari setiap use case.

3. Activity Diagram

   Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga

dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.

4. Class Diagram

   Class adalah dekripsi kelompok obyek-obyek dengan property, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya class diagram dapat memberikan pandangan global atas sebuah system. Hal tersebut tercermin dari class- class yang ada dan relasinya satu dengan yang lainnya. Sebuah sistem biasanya mempunyai beberapa *class diagram*. Class diagram sangat membantu dalam visualisasi struktur kelas dari suatu system.

5. Statechart Diagram

   Menggambarkan semua *state* (kondisi) yang dimiliki oleh suatu *object* dari suatu *class* dan keadaan yang menyebabkan *state* berubah. Kejadian dapat berupa *object* lain yang mengirim pesan. *State class* tidak digambarkan untuk semua *class*, hanya yang mempunyai sejumlah *state* yang terdefinisi dengan baik dan kondisi *class* berubah oleh st*ate* yang berbeda.

6. Sequence Diagram

   Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah scenario. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

7. Collaboration Diagram

   Menggambarkan kolaborasi dinamis seperti *sequence diagrams*. Dalam menunjukkan pertukaran pesan, *collaboration diagrams* menggambarkan *object* dan hubungannya (mengacu ke konteks). Jika penekannya pada waktu atau urutan gunakan *sequencediagrams,* tapi jika penekanannya pada konteks gunakan *collaboration diagram*.

2.2.3.2 Entitiy Relational Diagram (E-R Diagram)

Entity Relational Diagram merupakan salah satu pemodelan data konseptual yang paling sering digunakan dalam proses pengembangan basis data bertipe relasional.

Model E-R adalah rincian yang merupakan representasi logika dari data pada suatu organisasi atau area bisnis tertentu. Model E-R terdiri dari beberapa komponen dasar yaitu sebagai berikut:

1. Entitas

   Entitas adalah sesuatu atau objek di dunia nyata yang dapat dibedakan dari sesuatu atau objek yang lainnya. Sebagai contoh, setiap mahasiswa dalam suatu universitas adalah suatu entitas. Setiap fakultas dalam suatu universitas adalah juga suatu entitas. Dapat dikatakan bahwa entitas bisa bersifat konseptual/abstrak atau nyata hadir di dunia nyata.
2. Atribut

   Atribut adalah properti deskriptif yang dimiliki oleh setiap anggota dari himpunan entitas. Sebagai contoh entitas mahasiswa, atribut-atribut yang dimiliki adalah nim, nama mahasiswa, alamat dan lain-lain.
3. Hubungan antar relasi (Relationship)

   Hubungan antar relasi adalah hubungan antara suatu himpunan entitas dengan himpunan entitas yang lainnya. Misalnya, entitas mahasiswa memiliki hubungan tertentu dengan entitas matakuliah (mahasiswa mengambil matakuliah). Pada penggambaran model E-R, relasi adalah perekat yang menghubungkan suatu entitas dengan entitas yang lainnya.
4. Kardinalitas/Derajat Relasi

   Kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. Sebagai contoh: entitas-entitas pada himpunan entitas mahasiswa dapat berelasi dengan satu entitas, banyak entitas atau tidak satupun entitas dari himpunan entitas kuliah. Kardinalitas relasi yang terjadi di antara dua himpunan entitas dapat berupa:

   a. Satu ke Satu (One to One)

      Setiap entitas pada himpunan entitas A berhubungan dengan paling banyak Setiap entitas pada himpunan entitas A dapat berhubungan

dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya, dimana setiap entitas pada himpunan entitas B berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas A.

b. Banyak ke Satu (Many to One)

Setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B, tetapi tidak sebaliknya dengan entitas B.

c. Banyak ke Banyak (Many to Many)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, dan demikian sebaliknya, dimana setiap pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A.

2.2.4 Perangkat Lunak Pembangun Sistem

2.2.4.1 Java

Java adalah sebuah bahasa pemrograman, Java dapat membuat seluruh bentuk aplikasi, desktop, web dan lainnya, sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain. Java adalah bahasa pemrograman yang berorientasi objek (OOP) dan dapat dijalankan pada berbagai platform sistem operasi. Perkembangan Java tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat open source.java sudah memiliki library database yang disebut JDBC.

Java Platform adalah sebuah platform software yang berbeda dari berbagai platform lainnya seperti Microsoft Windows, Macintosh, OS / 2, UNIX dan NetWare,. Namun dirancang untuk menghasilkan dan berjalan sangat interaktif, dinamis, dan applet yang aman dan aplikasi-aplikasi pada sistem jaringan komputer. Java Platform berada di atas sistem operasi ini. Aplikasi yang ditulis dalam bahasa Java untuk mengkompilasi arsitektur netral JVM (Java Virtual Machine) bytecodes, yang tidak sekedar kode mesin normal. Selanjutnya interpreter java menterjemahkan kedalam kode

eksekusi. Keuntungan utama dari pendekatan ini adalah agar memungkinkan untuk menjalankan aplikasi Java pada sistem apapun, selama masih menerapkan sistem JVM.Java Platform ini terdiri dari dua bagian utama, yaitu Java Virtual Machine (JVM) dan Java API.

1. Sebagai Mesin Virtual. Java Virtual Machine (JVM) adalah suatu mesin virtaul yang menginterpretasikan bytecode-bytecode java ke dalam bahasa mesin. Bytecode java adalah adalah file hasil kompilasi kode java (ekstension filenya adalah .class). Jika sutu program java bernama ContohJava. Java dikompilasi maka hasilnya adalah: ContohJava.class (inilah yang disebut bytecode java). Sebenernya paradigma pemrogramman dengan cara menginterpretasikan kode (dalam java adalah bytecode) menurut para ahli kurang bagus dari sisi performance (kecepatan). Tapi JVM mencoba mengatasi masalah ini dengan menerapkan teknik just in time (JIT) compilation yaitu java bytecode langsung dikompilasi menjadi bahasa mesin untuk kode-kode program yang dijalankan secara berulang-ulang.

   JVM ada dalem paket yang disebut JRE (Java Run Time Environtment), yaitu paket lingkungan yang dibutuhkan ketika menjalankan aplikasi java. JRE inilah yang menterjemahkan Byte code hasil kompilasi ,menjadi native pada platform yang akan menjalankan aplikasi kita. Pada JRE inilah bersifat spesifik platform sehingga terdapat : jre untuk linux, jre untuk windows dan sebagainya. Salah satu komponen penting dari JRE adalah JVM (Java Virtual Machine).
2. Application Programming Interface (API). Adalah sekumpulan (libraries) komponen software (kelas-kelas dan interface-interface java) siap pakai yang memiliki berbagai kegunaan dan kemampuan yang berbeda-beda. Sekumpulan kelas-kelas dan interface-interface yang saling berkaitan diorganisasikan dalam suatu library. Library ini dikenal dengan sebutan package (paket). Paket inilah untuk

menambahkan fungsionalitas yang sudah dibuat dan dapat dirubah, sehingga menghemat waktu coding.

Berikut adalah kumpulan library dasar di java :

1. java.lang : adalah paket yang berisi kelas-kelas elemen dasar (untuk melakukan pemrogramman). Jika kita menggunakan kelas-kelas ini dalam kode kita tidak perlu melakukan import lagi.
2. java.io : adalah paket yang berisi kelas-kelas untuk melakukan manipulasi masukan dan keluaran (input dan output) termasuk penanganan masalah file.
3. java.util : adalah paket yang berisi kelas-kelas pelengkap seperti kelas struktur data dan kelas penanggalan.
4. java.net : adalah paket yang berisi kelas-kelas yang memungkinkan komputer saling berinteraksi dengan komputer yang lainnya melalui jaringan TCP/IP.
5. java.awt : adalah paket yang berisi kelas-kelas yang memungkin programmer untuk membuat aplikasi dengan antarmuka grafis, Graphical User Interface (GUI).
6. Java.awt.peer : merupakan paket yang berisi keseluruhan definisi interfece. Setiap antarmuka java.awt.peer berkaitan dengan salah satu atau java.awt Komponen MenuComponent kelas. The interface dalam paket ini menentukan metode yang harus didukung oleh komponen GUI pada platform tertentu.
7. java.applet : adalah paket yang berisi kelas-kelas yang memungkin programmer untuk membuat antarmuka (applet) yang interaktif di web.Semua program yang ditulis dalam java pasti memanfaatkan API Java ini.

2.2.4.2 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (Database Management System) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL

merupakan perangkat lunak basis data open source yang paling digemari, karena perangkat lunak ini merupakan perangkat lunak basis data yang powerful dan cukup stabil untuk digunakan sebagai media penyimpanan data.

MySQL memiliki beberapa keistimewaan, antara lain :

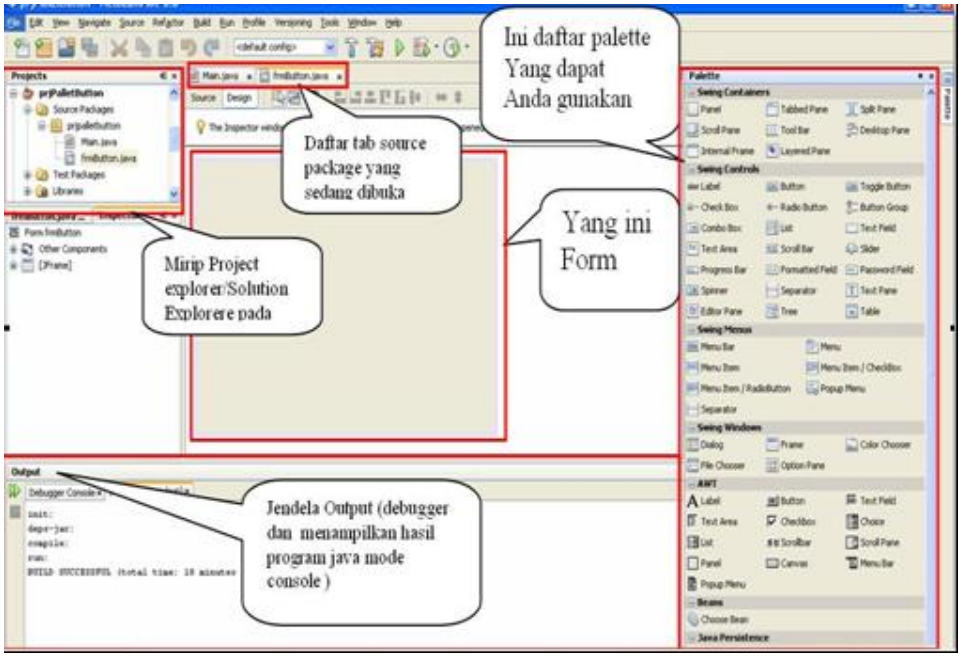
1. Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. *Multiuser*. MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
3. *Performance tuning*. MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
4. Jenis Kolom. MySQL memiliki tipe kolom yang sangat kompleks, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.
5. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (query).
6. Keamanan. MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.
7. Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
8. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).

9. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
10. Antar Muka. MySQL memiliki interface (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).
11. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan (*tool*)yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
12. Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle

2.2.4.3 Netbeans IDE 6.9.1

NetBeans IDE adalah sebuah lingkungan pengembangan - sebuah alat untuk pemrogram menulis, mengompilasi, mencari kesalahan dan menyebarkan program. Netbeans IDE ditulis dalam Java - namun dapat mendukung bahasa pemrograman lain (JavaScript , PHP , Python , Ruby , Groovy , C , C + + , Scala , Clojure , dan lain-lain) NetBeans juga mengacu pada kedua platform kerangka untuk aplikasi desktop Java, dan sebuah lingkungan pengembangan terpadu untuk pengembangan dengan Java.

NetBeans IDE dapat digunakan jika JVM diinstal, termasuk Windows, Mac OS, Linux, dan Solaris.Sebuah JDK diperlukan untuk pengembangan fungsionalitas Java, tetapi tidak diperlukan untuk pembangunan di bahasa pemrograman lain.



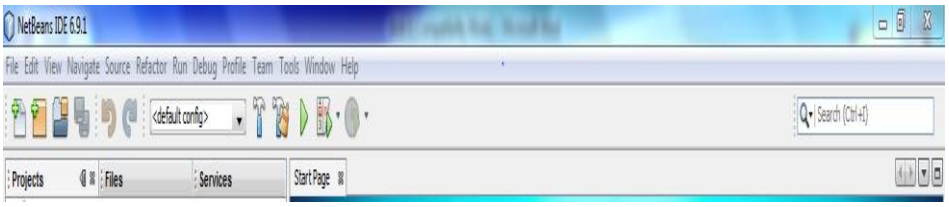
Gambar 2.5 Tampilan Netbeans 6.9.1

Komponen yang terdapat pada NetBeans

Komponen-komponen dapat diguna ulang sehingga mendukung RAD (RapidApplication development).

1. Jendela utama

   Disebut jendela utama karena merupakan pusat perintah pada IDE. Disini kitatemukan menu bar, beberapa toolbar, componen palette dan tab woskspace.



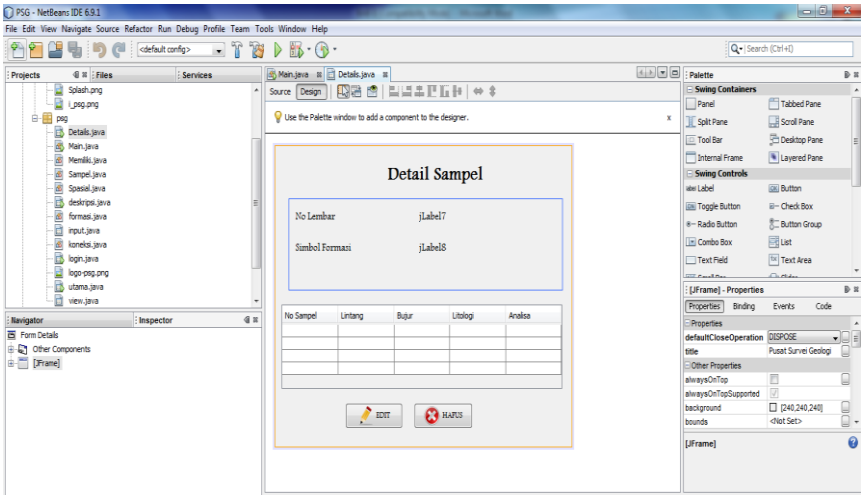
Gambar 2.6 Jendela utama pada Netbeans 6.9.1

2. Workspaces

   Sebuah workspaces adalah kumpulan-kumpulan dari jendela-jendela yang salingberhubungan erat untuk melakukan tugas-tugas tertentu. Jenis-jenis workspaces default, yaitu:

a. Editting

Editting terdiri dari jendela Explorer dan Propertis. Saat membuka sebuah file,Source Edittor secara otomatis terbuka. Source Edittor digunakan untuk mengeditfile-file java, HTML dan Plaint Text.



Gambar 2.7 Jendela editing pada Netbeans 6.9.1

b. GUI Editting

Workspaces GUI Editing digunakan untuk merancang dan membangun GUI.Didalamnya terdapat Explorer, Form Editor, Component Inspector dan SourceEdittor.

1. Form Editor

   Jendela Form Editor merupakan daerah utama untuk membuat dan memodifikasi sebuah GUI

2. Component Inspector

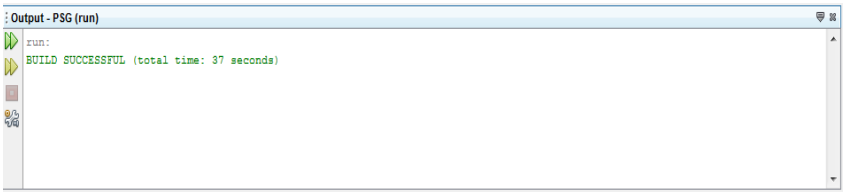
   Component Inspector memungkinkan kita untuk melihat komponenkomponen apa saja, baik yang tampak maupun yang tidak tampak. Disini kita mengubah properti dari komponen-komponen tersebut.

c. Browsing

Didalamnya terdapat Object Browser yang terdiri atas tiga panel dan jendela Properties.

d. Running

Didalamnya terdapat jendela Execution View dan Output. Saat mengeksekusiprogram, maka IDE akan secara otomatis ke Running Workspaces.



Gambar 2.8 Tampilan running pada Netbeans 6.9.1

e. Debugging

Debugging terdiri dari jendela-jendela Debugger dan Output. Jendela-jendela ini mempunyai tab panel untuk menset breakpoint, memonitor thread dan mengawasi nilai variabel-variabel. Jendela output menampilkan pesan dari Debugger

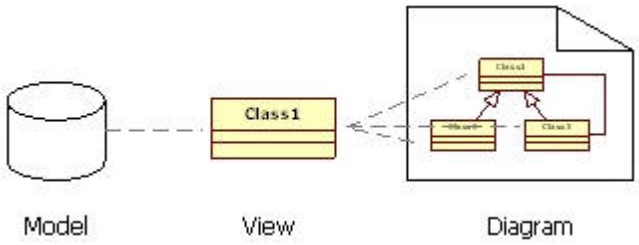
2.2.4.4 StarUML

StarUML adalah software permodelan yang mendukung UML (UnifiedModeling Language). Berdasarkan pada UML version 1.4 dan dilengkapi 11 macamdiagram yang berbeda, mendukung notasi UML 2.0 dan juga mendukungpendekatan MDA (Model Driven Architecture) dengan dukungan konsep UML.StarUML dapat memaksimalkan pruduktivitas dan kualitas dari suatu softwareproject.

Konsep Dasar dari StarUml:

1. Model, View and Diagram

   StarUML membuat perbedaan konseptual yang lebih jelas antara models, views and diagrams. Model adalah elemen yang memuat informasi untuk model software. View adalah suatu ekpresi visual dari informasi di dalam model dan Diagram adalah suatu koleksi dari elemen yang memberikan pemikiran user didalam mendesaian secara spesifik.



Gambar 2.10 Gambar Model, View dan Diagram

2. Project dan unit

a. Project

Project adalah unit manajemen dasar didalam StarUML. Suatu project dapat mengatur satu atau lebih model software. Project merupakan top-level package yang selalu ada didalam beberapa model software. Secara umum, satu project disimpan dalam satu file.

b. File Project

File project disimpan kedalam format XML dengan extension ".UML" . Semua model,viewsdan diagrams yang dibuat dengan StarUML disimpan dalam satu file project.

File project berisikan informasi sebagai berikut :

1. UML profiles used in the project
2. Unit files referenced by the project
3. Information for all models contained in the project
4. Information for all diagrams and views contained in the project

c. Units

Ada beberapa kasus dimana satu project perlu disimpan didalam beberapa file file kecilsehingga para pengembang dapat bekerja didalam satu project secara bersamaan. Didalamkasus ini suatu project dapat mengatur bermacam – macam unit. Suatu unit mempunyai3struktur hirarki dan berisikan beberapa sub-unit.

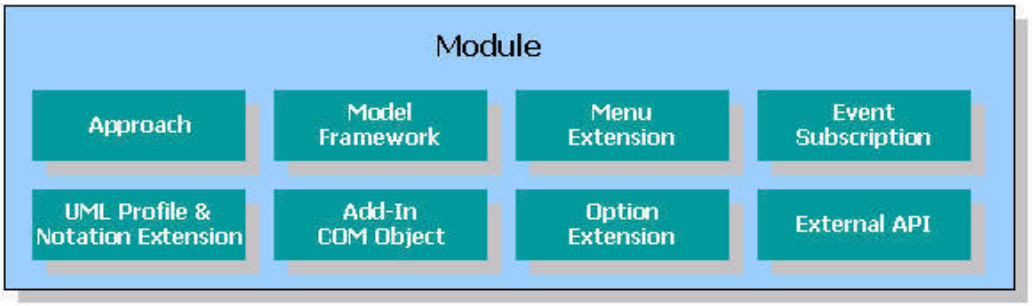
Unit disimpan sebagai ".UML " file danbeberapa mengacu pada file project (.UML) atau unit file lainnya (.UNT).

d. Komposisi Unit

Hanya package, subsystem dan elemen model yang dapat membentuk satu unit. Semuaelemen dibawah jenis elemen package ini disimpan didalam masing – masing **file unit(.UNT).**

3. Module

Modul adalah suatu package yang menyediakan fungsi – fungsi baru dan feature sebagaiperluasan dari StarUML. Modul dapat dibuat sebagai kombinasi dari beberapa elemen –elemen extension dan juga membuat beberapa jenis elemen – elemen didalam suatu modul.



Gambar 2.11 Gambar Module