BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

2.1.1 Sejarah Munculnya Chat (IM) Pada Perangkat HP

Dari Warnet ke Mobile Internet, istilah yang tidak aneh lagi pada masa saat ini. Internet saat ini tidak hanya dapat di akses di komputer saja akan tetapi di handphone kita sendiri sudah bisa internet yang dari berteknologi GSM(2G) sampai sekarang yang kita kenal 3G dan HSDPA(3,5G). Meskipun sudah bisa internet di handphone akan tetapi browser yang dipakai saat itu adalah WAP browser yang tampilan, struktur dan besarnya file sangat berbeda dengan internet biasa di komputer. Dimana yang bisa tampil di internet handphone hanya teks. Dan apabila ada gambar, bukan format foto atau image yang biasa tampil di internet komputer. Gambarnya berupa titik-titik yang persis kita lihat di wallpaper handphone monokrom.

Seiring berkembangnya teknologi jaringan, transfer data dan tentu saja teknologi pendukung di handphone seperti layar dan browser. Mulai dari teknologi CSD (Circuit Switch Data), HSCSD (High Speed Circuit Switch Data) sampai akhirnya GPRS datang. GPRS (General Packet Radio Service) adalah system transportasi data baru yang memanfaatkan gelombang radio sehingga data yang diakses dapat menjadi lebih cepat. Hadirnya GPRS ini kebetulan bersamaan dengan perkembangan handphone yang makin bagus mulai dari layar monokrom hingga berwarna, memori internal yang semakin besar sehingga di buat browser xHTML yang sudah bisa menampilkan situs internet di handphone seperti asli-

nya. Juga dengan perkembangan teknologi jaringan dapat menumbuhkan mobile internet. Mulai dari hadirnya EDGE (Enhanced Data for GSM Evolution), sampai akhirnya 3G menawarkan akses internet yang cukup baik kecepatan-nya. Dan ada lagi HSDPA(3.5G). Tidak hanya dari jaringan yang menawarkan akses internet, juga berbagai inovasi lainnya berperan dalam perkembangan teknologi mobile internet. Salah satu perkembangan yang ada yaitu banyak-nya browser yang bisa dipakai untuk membuka situs internet di handphone. Browser mulai dibuat sejak bahasa pemrograman seperti Java dan CC+ bisa diterapkan di handphone.

Hingga saat ini perkembangan browser sangat pesat. Dalam satu tahun terakhir , banyak pengembang yang mulai menawarkan browser di handphone. Yang mana suatu adaptasi dari browser serupa yang dipakai untuk versi internet biasa (yang di pakai PC atau laptop). Browser-browser ternama seperti IE, NetFront, AvantGo, Opera, Mozilla sampai Safari di kembang untuk versi Handphone. Instant messenger (IM) atau pesan instan berbasis data GPRS di handphone hadir dengan perkembangan-nya jaringan internet dan bahasa pemrograman seperti Java dan CC+ di handphone, dimana sebelum-nya awal instant messenger (IM) dirancang untuk internet di komputer biasa. Adapun penyelenggara pesan instant lewat internet (IM), diantaranya Yahoo Messenger yang dikenal sebagai YM, Google Talk yang dikenal sebagai GTalk, MSN Messenger dari Microsoft yang dikenal MSN, dan AOL instant Messenger yang dikenal dengan sebutan AOL, dan banyak lagi. Banyak pembuat software yang merancang program yang memungkinkan dapat menjalankan IM di handphone yaitu mig33 (www.mig33.com), bing (www.bing,im), YMTiny

(www.orisinil.com/ymtini/), shMessenger (www.shmessenger.ro). ebuddyBeta (http://get.ebuddy.com/), dan banyak lagi.

2.2 Landasan Teori

2.2.1 Teknologi Informasi

Hingga saat ini Teknlogi Informasi yang semakin banyak diminati oleh masyarakat adalah Handphone, yang digunakan untuk berkomunikasi baik dalam jarak jauh maupun dekat.

2.2.1.1 Konsep Dasar Teknologi

Perkembangan teknologi informasi semakin hari semakin pesat. Salah satunya yaitu penggunaan handphone. Handphone adalah suatu alat komunikasi wireless yang dapat juga digunakan untuk mengirim pesan dan membuka software aplikasi yang dibuat oleh suatu perusahaan ponsel. Salah satu contoh aplikasi handphone berupa Bluetooth yaitu standard wireless networking yang fungsinya untuk mengirim dan menerima pesan atau file dengan hubungan radio jarak dekat atau short-range, sehingga hubungan antar handphone, mobile PC, PDA, dan lainnya dapat dilakukan tanpa gangguan kabel atau wireless.

2.2.1.2 Definisi Bluetooth

Bluetooth adalah spesifikasi industri untuk jaringan kawasan pribadi (*personal area networks* atau PAN) tanpa kabel. Bluetooth menghubungkan dan dapat dipakai untuk melakukan tukar-menukar informasi diantara peralatan-peralatan. Spesifikasi dari peralatan bluetooth ini dikembangkan dan didistribusikan oleh kelompok. Bluetooth beroperasi dalam pita frekuensi 2,4 Ghz dengan menggunakan sebuah *frequence hopping traceiver* yang mampu

menyediakan layanan komunikasi data dan suara secara real time antara host-host bluetooth dengan jarak terbatas. Kelemahan teknologi ini adalah jangkauannya yang pendek dan kemampuan transfer data yang rendah.

1. Asal nama bluetooth dan lambangnya yaitu nama "bluetooth" berasal dari nama raja diakhir abad ke sepuluh, Harald Blantad yang di Inggris juga dijuluki Harald Bluetooth kemungkinan karena memang giginya berwarna gelap. Ia adalah raja Denmark yang telah berhasil menyatukan suku-suku yang sebelumnya berperang, termasuk suku dari wilayah yang sekarang bernama Norwegia dan Swedia. Bahkan wilayah Scania di Swedia, tempat teknologi bluetooth ini ditemukan juga termasuk daerah kekuasaannya. Kemampuan raja itu sebagai pemersatu juga mirip dengan teknologi bluetooth sekarang yang bisa menghubungkan berbagai peralatan seperti komputer personal dan handphone. Sedangkan logo bluetooth berasal dari penyatuan dua huruf Jerman yang analog dengan huruf H dan B (singkatan dari Harald Bluetooth) yaitu ᚼ (Hagall) dan ᛒ (Berkanan) yang kemudian digabungkan,berikut contoh gambar logo bluetooth:



2. Sejarah awal mula dari Bluetooth adalah sebagai teknologi komunikasi wireless (tanpa kabel) yang beroperasi dalam pita frekuensi 2,4 GHz unlicensed ISM (*Industrial, Scientific and Medical)* dengan menggunakan sebuah *frequency hopping tranceiver* yang mampu menyediakan layanan komunikasi data dan suara secara *real-time* antara host-host bluetooth dengan jarak jangkauan layanan yang terbatas (sekitar 10 meter). Bluetooth berupa card yang menggunakan frekuensi radio standar IEEE 802.11 dengan jarak

layanan yang terbatas dan kemampuan data transfer lebih rendah dari card untuk Wireless Local Area Network (WLAN).

Pembentukan bluetooth dipromotori oleh 5 perusahaan besar yaitu Ericsson, IBM, Intel, Nokia dan Toshiba membentuk sebuah Special Interest Group (SIG) yang meluncurkan proyek ini. Pada bulan juli 1999 dokumen spesifikasi bluetooth versi 1.0 mulai diluncurkan. Pada bulan Desember 1999 dimulai lagi pembuatan dokumen spesifikasi bluetooth versi 2.0 dengan tambahan 4 promotor baru yaitu 3Com, Lucent Technologies, Microsoft dan Motorola. Saat ini, lebih dari 1800 perusahaan di berbagai bidang bergabung dalam sebuah konsorsium sebagai adopter teknologi blueooth. Walaupun standar Bluetooth SIG saat ini 'dimiliki' oleh group promotor tetapi ia diharapkan akan menjadi sebuah standar IEEE (802.15).

3. Time Slot: kanal dibagi dalam time slot-time slot, masing-masing mempunyai panjang 625 ms. Time slot-time slot tersebut dinomori sesuai dengan *clock bluetooth* dari *master piconet.* Batas penomoran slot dari 0 sampai dengan 227-1 dengan panjang siklus 227. Di dalam time slot, *master* dan *slave* dapat mentransmisikan paket-paket dengan menggunakan skema TDD (*Time-Division Duplex).* Master hanya memulai melakukan pentransmisiannya pada nomor time slot genap saja sedangkan slave hanya memulai melakukan pentransmisiannya pada nomor time slot ganjil saja.
4. Protokol: maksud dari protokol adalah untuk mempercepat pengembangan aplikasi-aplikasi dengan menggunakan teknologi Bluetooth. Layer-layer bawah pada stack protokol bluetooth dirancang untuk menyediakan suatu dasar yang fleksibel untuk pengembangan protokol yang lebih lanjut.

Protokol-protokol yang lain seperti RFCOMM diambil dari protokol-protokol yang sudah ada dan protokol ini hanya dimodifikasi sedikit untuk disesuaikan dengan kepentingan bluetooth. Stack protokol bluetooth dapat dibagi ke dalam empat layer sesuai dengan tujuannya.

5. Pengukuran: ada tiga aspek dalam melakukan pengukuran Bluetooth yaitu pengukuran RF(*Radio Frequency),* protokol dan *profile.* Pengukuran radio dilakukan untuk menyediakan compatibility perangkat radio yang digunakan di dalam sistem dan untuk menentukan kualitas sistem serta dapat menggunakan perangkat alat ukur RF standar seperti *spectrum analyzer,transmitter analyzer, power meter, digital signal generator* dan *bit-error-rate tester* (BERT).
6. Fitur keamanan : dirancang untuk memiliki fitur-fitur keamanan sehingga dapat digunakan secara aman baik dalam lingkungan bisnis maupun rumah tangga.Fitur-fitur yang disediakan bluetooth antara lain sebagai berikut:

   Enkripsi data, Authentifikasi user, Fast frekuensi-hopping (1600 hops/sec), Output power control. Fitur-fitur tersebut menyediakan fungsi-fungsi keamanan dari tingkat keamanan layer fisisk/ radio yaitu gangguan dari penyadapan sampai dengan tingkat keamanan layer yang lebih tinggi seperti password dan PIN. Tetapi dari sebuah artikel internet, menurut penelitian dua mahasiswa Tel Aviv University, mengenai adanya kemungkinan Bluetooth bisa disadap dengan proses pairing berpasangan. Caranya adalah dengan menyiapkan sebuah kunci rahasia pada proses pairing. Selama ini dua perangkat bluetooth menyiapkan kunci digital 128 bit. Ini adalah kunci rahasia yang kemudian disimpan dan dipakai dalam proses enkripsi pada komunkasi

selanjutnya. Langkah pertama ini mengharuskan pengguna yang sah untuk menginputkan kunci rahasia yang sesuai, PIN empat digit keperangkat. Pesan lalu dikirim ke perangkat lainnya, dan ketika ditanya kunci rahasia, dia berpura-pura lupa. Hal ini memacu perangkat lain untuk memutus kunci dan keduanya lalu mulai proses pairing baru. Kesempatan ini kemudian bisa dimanfaatkan oleh hacker untuk mengetahui kunci rahasia yang baru. Selain mengirim ini ke perangkat Bluetooth yang dituju, semua perangkat Bluetooth yang ada dalam jangkauan itu juga tetap dapat disadap.

2.2.1.3 Definisi Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan class dan operation dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C. 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 Desain), Jim Rumbaugh OMT (Object Modeling Technique), dn Ivar Jacobson OOSE (Object-Oriented Software Enginering). UML merupakan *standardized modelling language* yang terdiri dari kumpulan-kumpulan digram, dikembangkan untuk membantu para pengembang sistem dan software agar bisa menyelesaikan tugas-tugas seperti: Spesifikasi, Visualisasi, Desain Arsitektur, Konstruksi, Simulasi dan testing, Dokumentasi. UML dikembangkan sebagai ide dasar untuk mempromosikan hubungan dan produktifitas antara para pengembang dari *oject-oriented system.*

1. Sejarah UML

Sejarah UML cukuplah panjang. Sampai di era tahun 1990 seperti kita ketahui puluhan metodologi pemodelan berorientasi object telah bermunculan di dunia. Diantaranya adalah:

a. metodologi booch
b. metodologi coad
c. metodologi OOSE
d. metodologi OMT
e. metodologi shlaer-mellor
f. metodologi wirfs-brock dan lain-lain.

Massa itu terkenal dengan massa perang metodologi (method war) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerja sama dengan group/perusahan lain yang menggunakan metodologi yang berlainan.Dimulai pada bulan Oktober 1994 Booch, Rumbaugh dan Jacobson,

yang merupakan tiga tokoh yang boleh dikata metodologinya banyak digunakan mempelopori usaha untuk penyatuan metodologi pendesainan berorientasi objek. Pada tahun 1995 direalise draft pertama dari UML (versi 0.8). Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh Object Management Group (OMG-http://www.omg.org). Tahun 1997 UML versi 1.1 muncul, dan saat ini versi terbaru adalah 1.5 yang dirilis bulan maret 2003. Booch, Rumbaugh dan Jacobson menyusun tiga buku serial tentang UML pda tahun 1999. Sejak saat itulah UML telah menjelma menjadi standar bahasa pemodelan untuk aplikasi berorientasi objek.

2. Kategori diagram *Unified modeling language* (UML)

Adapun kategori *Unified modeling language* (UML) yaitu:

a. Structural Diagram: kita menggunakan *structural diagram* untuk menampilkan blok bangunan dari sistem kita yaitu merupakan fitur yang tidak berubah bersama waktu. Diagram ini menjawab pertanyaan, ada apa disana?
b. Behavioral Diagram: kita menggunakan *behavioral diagram* untuk menampilkan bagaimana sistem kita merespon permintaan atau apa saja seiring waktu.
c. Interaction Diagram: merupakan tipe dari *behavioral diagram*. Kita menggunakan *interaction diagram* untuk melukiskan perubahan dari pesan-pesan dalam suatu kolaborasi (kumpulan dari object-object yang sama) sehingga tujuan bisa tercapai.

Adapun penjelasan dari kategori diatas adalah

a. Structural diagram

1. Structural diagram (class diagram): digunakan untuk menampilkan entiti dunia nyata, elemen dari analisa dan desain, atau implementasi class dan relasinya.
2. Structural diagram (Composite structure diagram): Digunakan untuk menampilkan bagaimana sesuatu itu dibuat.
3. Structural diagram (Deployment diagram): Digunakan untuk menampilkan arsitektur run-time dari suatu sitem, kerangka hardware, ruang lingkup software, dan sebagainya.
4. Structural diagram (Component diagram): Digunakan untuk menampilkan organisasi dan hubungan antar sistem.
5. Structural diagram (Package diagram): Digunakan untuk mengorganisir elemen model dan menampilkan ketergantungan antara mereka.

b. Behavioral Diagram

1. Behavioral Diagram (Activity diagram): Digunakan untuk menampilkan arus data dari kebiasaan antar object.
2. Behavioral Diagram (Use-case diagram): Digunakan untuk menampilkan layanan yang bisa diminta oleh actor dari sistem kita.
3. Behavioral Diagram (State machine diagram / Protokol state machine diagram): Digunakan untuk menampilkan urutan proses dari suatu object dan kondisinya saat ini.

c. Interaction Diagram

1. Interaction Diagram (Overview diagram): digunakan untuk menampilkan banyak skenario interaksi (urutan dari kebiasaan) bagi suatu kolaborasi

(kumpulan elemen yang sama dan saling bekerja agar tercapai tujuan yang diinginkan).

2. Interaction Diagram (Sequence diagram): Digunakan untuk fokus pada perubahan pesan antara group dari suatu object dan urutan pesan tersebut.
3. Interaction Diagram (Communication diagram): Digunakan untuk fokus pada perubahan pesan antara grup dari suatu object dan relasi dari object-object tersebut.
4. Interaction Diagram (Timing diagram): Digunakan untuk menampilkan perubahan dan hubungan terhadap waktu nyata atau terhadap proses sistem.

Karena UML sangatlah fleksibel, kita akan menjumpai cara dalam meng-kategorikan diagram kita. Pohon kategori di bawah ini cukup terkenal:

1. Static diagram: Menampilkan fitur statis dari sistem. Kategori ini hampir sama dengan *structural diagram.*
2. Dynamic diagram: Menampilkan bagaimana proses perubahan yang terjadi dalam sistem sepanjang waktu. Kategori ini mencakup *UML state-machine diagram* dan *timing diagram.*
3. Functional diagram: Menampilkan detail dari proses dan algoritma. Kategori ini mencakup *use-case, interaction,* dan *acivity diagram.*

3. Class diagram

Class adalah sebuah spesifikasi yang jika diinstalisasi 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 :

•Private, tidak dapat dipanggil dari luar class yang bersangkutan

• Protected, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya

• 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.

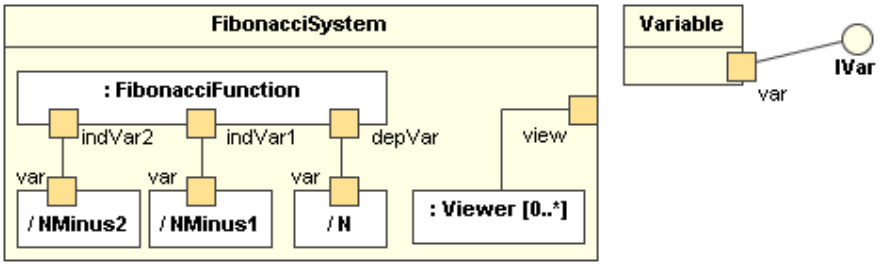
Hubungan Antar Class:

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 di-passing dari satu class kepada class lain. Hubungan dinamis dapat digambarkan dengan menggunakan sequence diagram yang akan dijelaskan kemudian.

4. Composite structure diagram

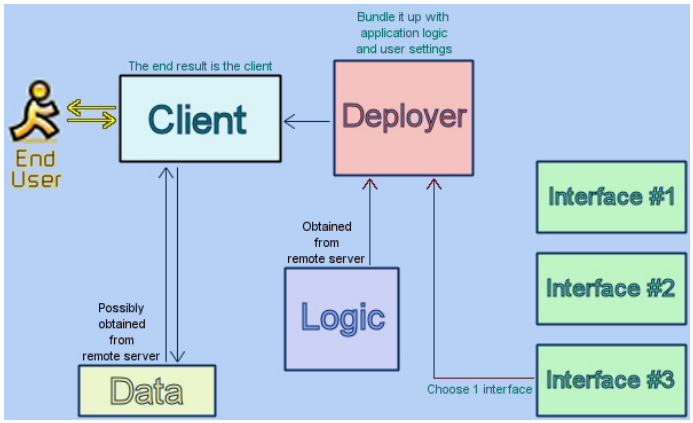
Komposit dalam struktur diagram Unified Modeling Language (UML) adalah jenis struktur statis diagram, yang menunjukkan struktur internal dari satu kelas dan kolaborasi yang membuat struktur ini mungkin. Diagram ini dapat termasuk komponen internal, pelabuhan melalui bagian-bagian yang saling berinteraksi dengan atau melalui contoh yang kelas berinteraksi dengan bagian-bagian dan dengan dunia luar, dan konektor antara bagian atau pelabuhan. J komposit struktur adalah serangkaian unsur yang saling bekerja sama di runtime untuk mencapai beberapa tujuan. Setiap elemen memiliki peran yang ditentukan dalam kolaborasi.



Gambar 2.0. Composite structure diagram

5. Deployment diagram

Deployment diagram di Unified Modeling Language berfungsi untuk model fisik deployment dari seni di deployment target. Deployment diagram menunjukkan "alokasi seni ke Nodes menurut Deployments ditetapkan di antara mereka. Deployment sebuah artifact ke node ditunjukkan dengan meletakkan artifact dalam node. Contoh dari node (dan perangkat dan pelaksanaan lingkungan) digunakan dalam deployment diagram untuk menunjukkan keserbaragaman node ini. Misalnya, beberapa contoh aplikasi server eksekusi lingkungan dapat digunakan di dalam satu perangkat node mewakili aplikasi server clustering.

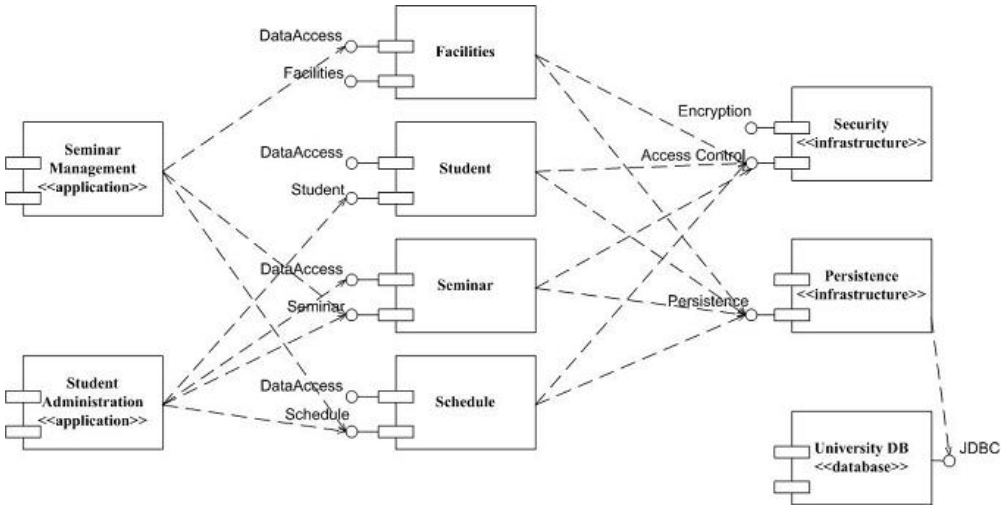


Gambar 2.1. Deployment diagram

6. Component diagram

Komponen kabel bersama-sama dengan menggunakan sebuah assembly konektor yang diperlukan untuk menghubungkan satu komponen antarmuka dengan antarmuka yang disediakan dari komponen lain. Ini menggambarkan layanan konsumen - penyedia layanan hubungan antara kedua komponen. An assembly konektor adalah "penghubung antara dua komponen yang menentukan bahwa salah satu komponen menyediakan layanan lain yang membutuhkan

komponen. An assembly konektor adalah konektor yang ditetapkan dari antarmuka diperlukan atau port yang diberikan antarmuka atau pelabuhan. Bila menggunakan komponen diagram untuk menampilkan struktur internal dari komponen, yang disediakan dan antarmuka yang diperlukan meliputi komponen dapat melimpahkan ke antarmuka yang sesuai yang berisi komponen. delegasi Konektor adalah "konektor yang menghubungkan eksternal kontrak dari komponen (sebagaimana ditentukan oleh port) ke internal realisasi bahwa perilaku oleh komponen dari bagian-bagian. Komponen diagram dapat digunakan untuk mengilustrasikan struktur sistem kompleks sewenang-wenang. Contoh di atas menggambarkan apa yang khas Asuransi kebijakan sistem administrasi mungkin. Adalah mungkin untuk membayangkan bahwa setiap komponen yang digambarkan dalam diagram di atas akan, pada gilirannya, ada komponen lainnya yang menggambarkan diagram struktur internal mereka.



Gambar 2.2. Component diagram

7. Package diagram

Package diagram dalam Unified Modeling Language menggambarkan dependensi antara paket yang membentuk model.Isi

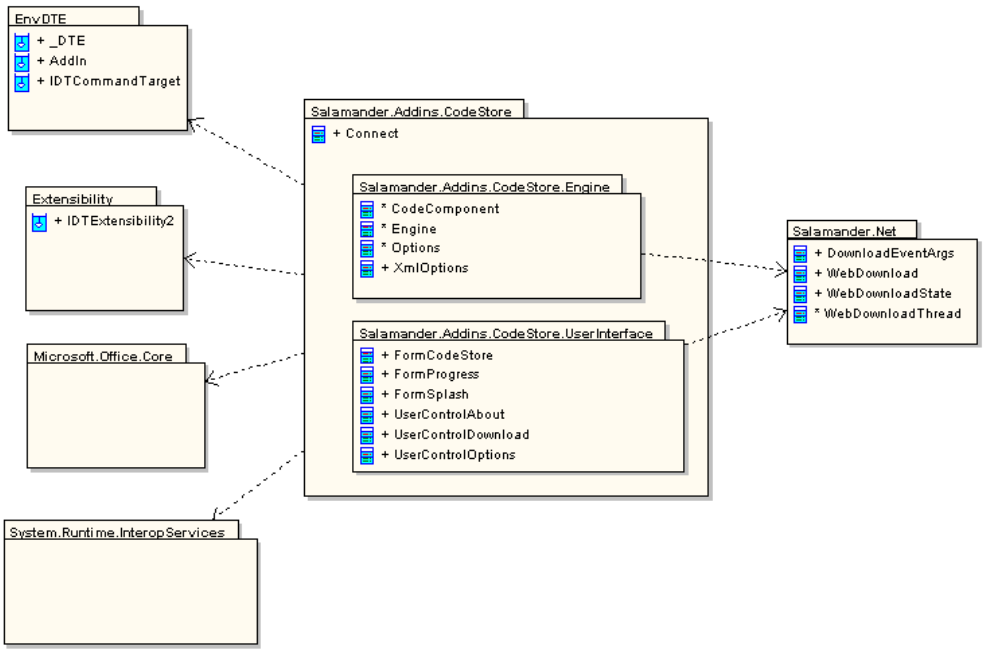
1.Sekilas

2.Penggunaan

3.Referensi

4.Pranalaluar

Di samping standar UML Ketergantungan hubungan, ada dua jenis khusus ditetapkan antara dependensi paket: Paket impor Paket menggabungkan Paket impor adalah "suatu hubungan antara impor namespace dan sebuah paket, menunjukkan bahwa impor namespace menambahkan nama-nama anggota paket ke nama sendiri.Secara default, sebuah unlabeled ketergantungan antara dua paket dianggap sebagai paket impor hubungan. Sebuah paket adalah gabungan "yang diarahkan hubungan antara dua paket, yang menunjukkan bahwa isi dua paket akan digabungkan. Hal ini sangat mirip dengan generalisasi dalam arti bahwa sumber unsur konseptual menambahkan karakteristik dari target elemen-nya sendiri karakteristik yang satu unsur yang menggabungkan karakteristik dari keduanya. Diagram dapat menggunakan paket paket berisi kasus gunakan untuk mengilustrasikan fungsi software sistem.Paket diagram dapat menggunakan paket yang mewakili berbagai lapisan dari perangkat lunak sistem untuk mengilustrasikan layered arsitektur software sistem. Dependensi antara paket ini dapat perhiasan dengan label / stereotip untuk menunjukkan mekanisme komunikasi antara lapisan. Contoh gambar package diagram:



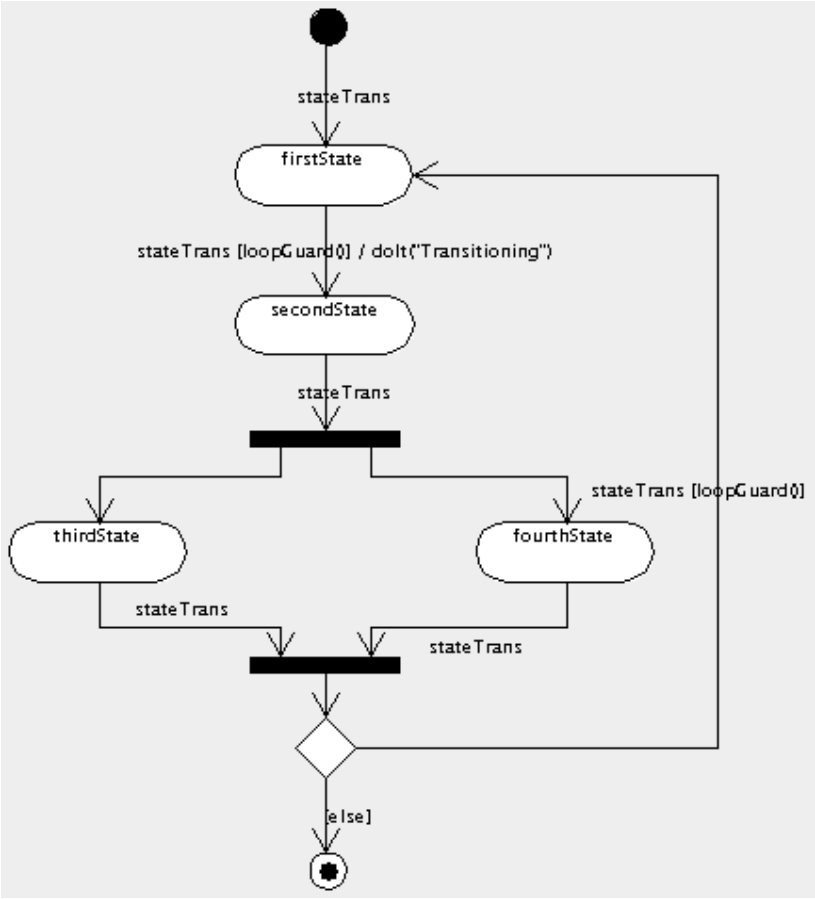
Gambar 2.3. Package diagram

8. Activity diagram

Activity diagram atau Ringkasan diagram adalah diagram teknik loosely ditetapkan untuk menampilkan workflows dari stepwise kegiatan dan tindakan, dengan dukungan untuk pilihan, dan perulangan concurrency. [1] Pada Unified Modeling Language, diagram aktivitas dapat digunakan untuk menjelaskan bisnis dan operasional langkah demi langkah workflows komponen dalam suatu sistem. Suatu kegiatan diagram keseluruhan menunjukkan aliran kontrol.

Dalam SysML kegiatan diagram telah diperpanjang untuk menunjukkan langkah-langkah yang mengalir di antara menyampaikan unsur fisik (misalnya, bensin) atau energi (misalnya, torque, tekanan). Perubahan tambahan membolehkan diagram untuk terus mendukung perilaku dan data terus mengalir. Dalam UML 1.x, suatu kegiatan diagram adalah variasi dari Negara UML

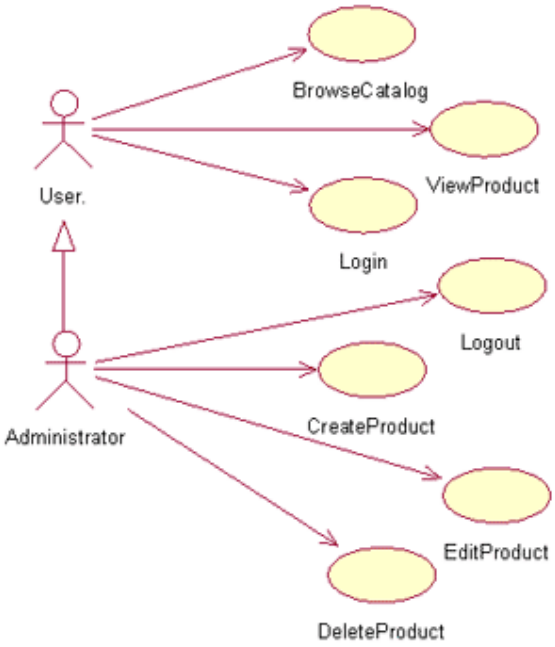
diagram di mana "negara" merupakan kegiatan, dan transisi yang mewakili selesainya kegiatan mereka. Konstruksi Ringkasan diagram biasanya digunakan untuk pemodelan proses bisnis. Mereka terdiri dari: Initial node. Ringkasan terakhir node. Kegiatan Titik dari diagram awal adalah node, dan kegiatan terakhir adalah node berakhir. Diagram suatu kegiatan dapat memiliki nol atau lebih kegiatan node terakhir. Di antara kegiatan yang diwakili oleh Rectangles bulat. Berikut gambar contoh activity diagram:



Gambar 2.4. Activity diagram

9. 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 aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor 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 meng-include fungsionalitas use case lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa use case yang di-include akan dipanggil setiap kali use case yang meng-include dieksekusi secara normal. Sebuah use case dapat di-include oleh lebih dari satu use case lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang common. Sebuah use case juga dapat meng-extend use case lain dengan behaviour-nya sendiri. Sementara hubungan generalisasi antar use case menunjukkan bahwa use case yang satu merupakan spesialisasi dari yang lain. Contoh gambar use case dapat dilihat sebagai berikut:



Gambar 2.5. Use case diagram

10. State machine diagram / protokol state machine diagram diagram

State diagram adalah jenis diagram yang digunakan dalam ilmu komputer dan bidang terkait untuk menggambarkan perilaku dari sistem. State diagram memerlukan sistem yang dijelaskan terdiri dari jumlah yang terbatas, kadang-kadang, ini memang terjadi, sementara di lain kali ini adalah wajar abstak. Ada banyak bentuk diagram negara yang berbeda dan ada sedikit berbeda semantics.

State diagram digunakan untuk menggambarkan perilaku dari sistem. State diagram dapat menjelaskan kemungkinan negara sebagai obyek peristiwa terjadi. Setiap diagram biasanya merupakan objek dari sebuah kelas yang berbeda dan melacak status dari objek melalui sistem. State diagram dapat digunakan untuk grafis mewakili negara terbatas mesin. Hal ini diperkenalkan oleh Taylor Booth

dalam buku 1967 "berurut Mesin Automata dan Teori". Lain mungkin adalah perwakilan negara transisi table.

Directed graph yaitu classic bentuk diagram negara terbatas untuk

a. State q adalah mesin diarahkan grafik dengan elemen sebagai berikut:
   vektor terbatas biasanya diwakili oleh kalangan dan dicap dengan unik designator simbol atau kata-kata yang ditulis di dalam mereka;

b. Simbol Σ * Input: a terbatas kumpulan masukan atau simbol designators;

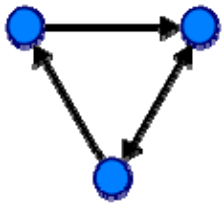
c. Output simbol Z: koleksi yang terbatas output atau simbol designators; Output fungsi ω mewakili pemetaan dari input menjadi output simbol simbol, denoted matematis sebagai ω: $\Sigma \times T \rightarrow Z$.

a. Edges o: mewakili "transisi" antara dua negara yang disebabkan oleh input (diidentifikasi oleh mereka diambil pada simbol "pinggiran"). Sebuah 'ujung' biasanya diambil sebagai panah diarahkan dari sekarang menuju negara-negara berikutnya. Pemetaan ini menjelaskan negara transisi yang terjadi pada masukan dari simbol tertentu. Ini ditulis sebagai δ matematis: $\Sigma \times T \rightarrow Z$

b. Start state q0: (tidak akan ditampilkan pada contoh di bawah). Awal negara q0 $\in$ T biasanya diwakili oleh sebuah panah tanpa asal pointing ke negara. Dalam teks tua sejak awal state tidak akan ditampilkan dan harus inferred dari teks.

c. Accepting state(s) F: Jika digunakan, misalnya untuk menerima automata, M $\in$ T adalah menerima negara. Biasanya diambil sebagai dua lingkaran. Kadang-kadang menerima negara (s) berfungsi sebagai "Final" (terdiam, terperangkap) menyatakan. Deterministic terbatas untuk negara mesin (DFA), nondeterministic terbatas negara mesin (NFA), nondeterministic umum terbatas mesin negara (GNFA), atau mesin Moore, input adalah denoted pada

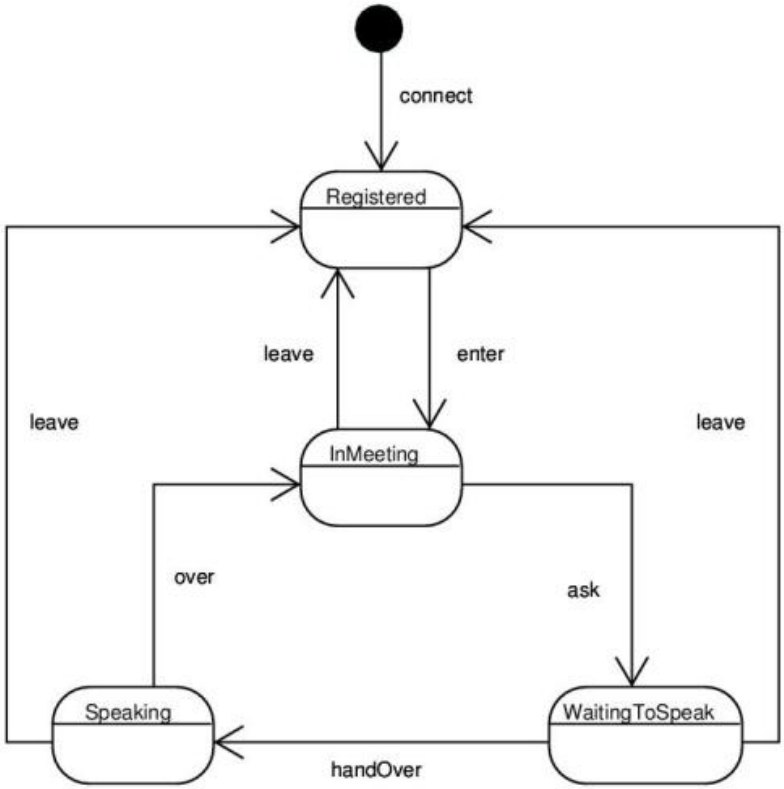
setiap tepi. Bertepung untuk mesin, input dan output yang signified pada setiap ujung, dipisahkan dengan tanda garis miring "/": "1 / 0" menandakan negara perubahan atas hadapi simbol "1" menyebabkan simbol "0" untuk output. Untuk mesin Moore negara output biasanya tertulis di dalam lingkaran negara, juga lepas dari negara designator dengan slash "/". Ada juga varian yang menggabungkan kedua notations. Misalnya, jika suatu negara memiliki jumlah output (misalnya "yang = motor jarum jam = 1, b = lampu hati-hati tidak aktif = 0") diagram ini harus mencerminkan: misalnya "q5 / 1,0" designates negara dengan keluaran yang q5 = 1, b = 0. Designator ini akan ditulis di dalam lingkaran state diagram.contoh gambar:



Gambar 2.6. direct graph diagram

The Unified Modeling Language (UML) state diagrm pada dasarnya adalah sebuah Harel dengan standar statechart notasi [6], yang dapat menjelaskan berbagai sistem, program-program dari komputer ke proses bisnis. Berikut ini adalah elemen dasar notational yang dapat digunakan untuk membentuk sebuah diagram: Diisi lingkaran, yang pada awal state Hollow lingkaran berisi penuh lingkaran yang lebih kecil, menunjukkan akhir state (jika ada) Rounded rectangle, menunjukkan sebuah state. Atas segi empat berisi nama state. Dapat berisi garis horisontal di tengah-tengah, di bawah ini dimana kegiatan yang dilakukan di state yang ditunjukkan Arrow, menunjukkan transisi. Nama acara (jika ada) transisi ini menyebabkan label panah tubuh. J menjaga ekspresi dapat ditambahkan sebelum

"/" dan ditutupi dalam kurung kotak-(eventName [guardExpression]), menunjukkan bahwa ekspresi ini harus benar untuk transisi ke berlangsung. Jika tindakan ini dilakukan selama masa transisi, hal ini ditambahkan ke label berikut sebuah "/" (eventName [guardExpression] / tindakan). Thick horisontal sesuai dengan baik x> 1 baris dan 1 baris masuk atau keluar masuk dan 1 line x> 1 meninggalkan baris. Ini menunjukkan bergabung, masing-masing.Menurut Pilone [7], satu-satunya standar menjaga kondisi ELSE. Tidak ada contoh lain yang diberikan dalam publikasi. Contoh gambar state diagram:

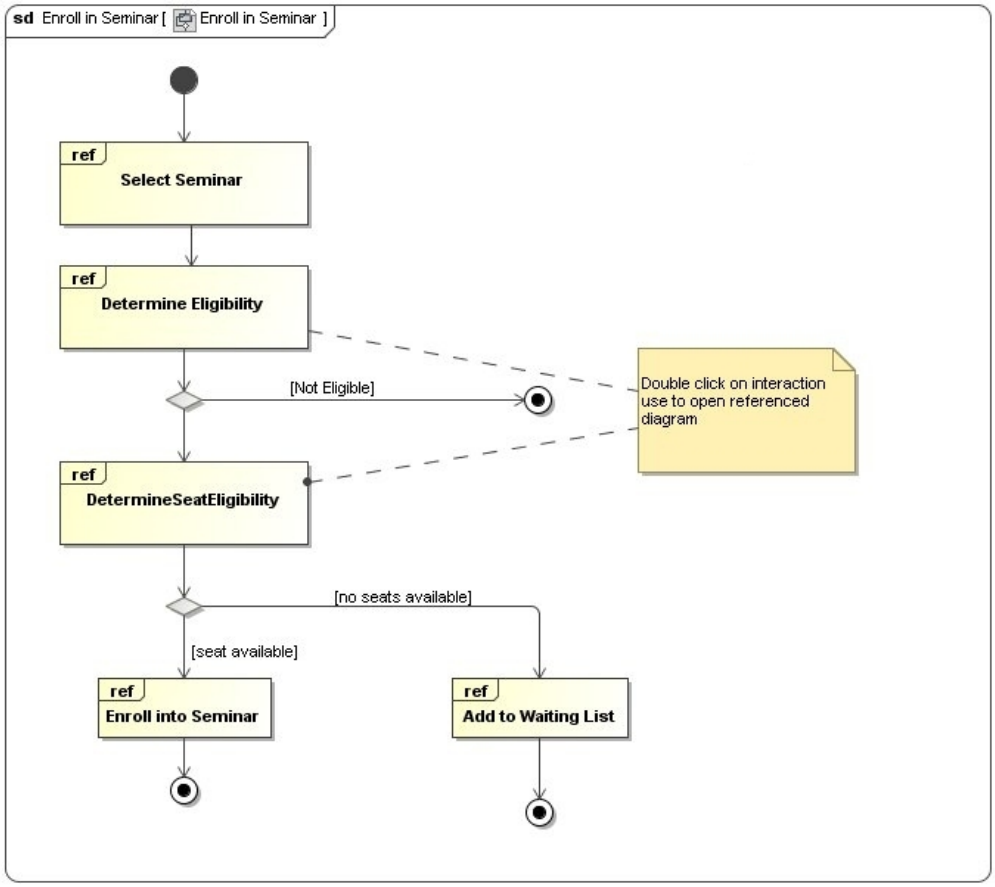


Gambar 2.7. state diagram

11. Overview diagram

Overview diagram di Unified Modeling Language (UML) adalah jenis kegiatan diagram yang mewakili interaksi node diagram. Mereka yang tinggi

tingkat struktur mekanisme urutan diagram. Interaksi Sekilas diagram menggambarkan ikhtisar dari aliran kontrol di mana setiap node dapat berupa diagram interaksi. Interaksi Overview diagram diagram dan menggabungkan aktivitas di urutan diagram-line. Diagram interaksi ini dapat termasuk urutan, komunikasi, dan interaksi Sekilas waktu diagram. Dalam diagram node adalah interaksi dari urutan diagram, dan ujung-ujungnya menunjukkan urutan ini interaksi yang terjadi. Anda juga dapat menggunakan diagram interaksi Sekilas deconstruct ke sebuah kompleks skenario yang lain memerlukan beberapa jika-maka-jalan lain yang akan digambarkan sebagai satu urutan diagram. Sebagian besar notasi untuk diagram interaksi Sekilas sama untuk kegiatan diagram. Misalnya, awal, akhir, keputusan, menggabungkan, garpu dan bergabung semua node yang sama. Namun, interaksi Sekilas diagram baru memperkenalkan dua elemen: terjadi interaksi dan interaksi elemen.contoh gambar:

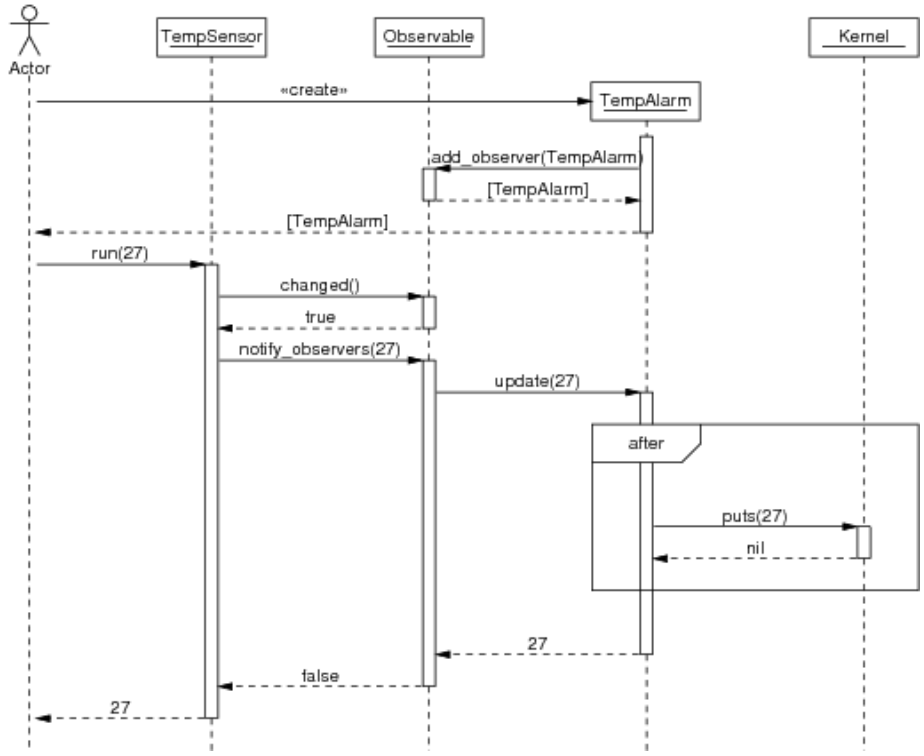


Gambar 2.8. Overview diagram

12. 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 vertikal (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 men-trigger 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. Contoh gambar:



Gambar 2.9. Sequence diagram1

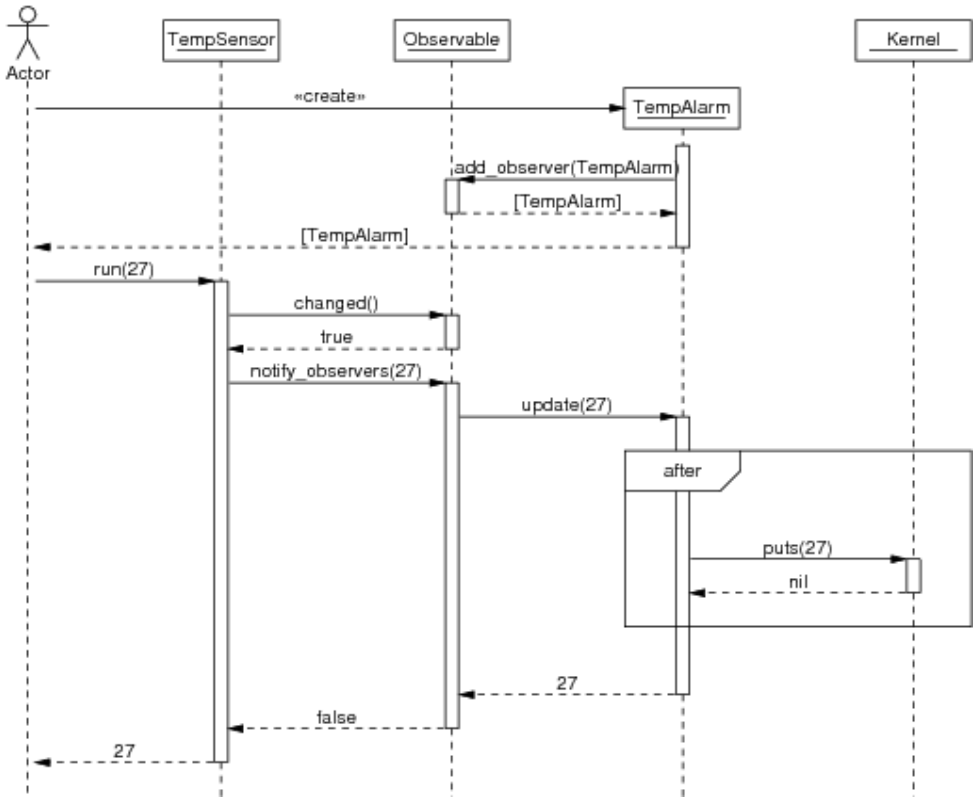
13. Communication diagram

Communication diagram dalam Unified Modeling Language (UML) 2.0, merupakan versi sederhana dari UML 1.x kolaborasi diagram. UML memiliki empat jenis diagram Interaksi:

1. sequence diagram

2. communication diagram
3. Interaksi overview diagram
4. Timing diagram

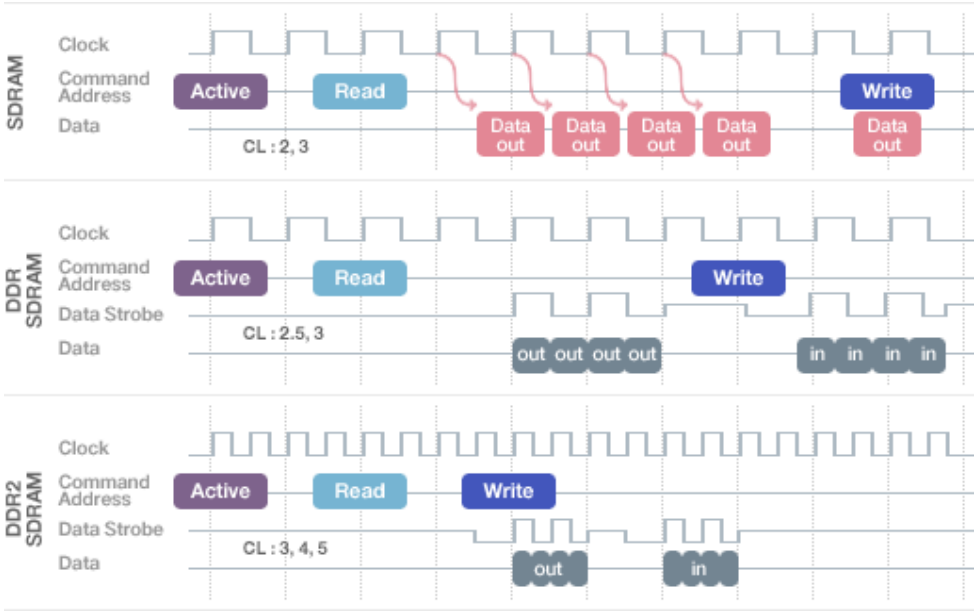
Communication diagram model interaksi antara obyek atau bagian dari segi sequenced pesan. Communiation diagram merupakan kombinasi dari informasi yang diambil dari Kelas, urutan, dan kasus Gunakan diagram menjelaskan baik statis dan dinamis struktur perilaku dari sistem. Namun, komunikasi menggunakan diagram bebas-bentuk susunan obyek dan link seperti yang digunakan dalam Object diagram. Untuk mempertahankan susunan pesan sedemikian bebas-bentuk diagram, pesan yang diberi label dengan nomor kronologis dan ditempatkan di dekat link yang dikirim melalui pesan. Membaca diagram melibatkan komunikasi pesan mulai dari 1.0, dan mengikuti pesan dari objek ke objek. Komunikasi diagram menunjukkan banyak informasi yang sama seperti urutan diagram, tetapi karena bagaimana informasi yang disajikan, beberapa lebih mudah untuk menemukan dalam satu diagram daripada yang lain. Menunjukkan diagram komunikasi yang masing-masing elemen berinteraksi dengan baik, namun urutan diagram menunjukkan urutan interaksi yang terjadi lebih jelas. Contoh gambar communication diagram:



Gambar 3.0. Sequence diagram2

14. Timing diagram.

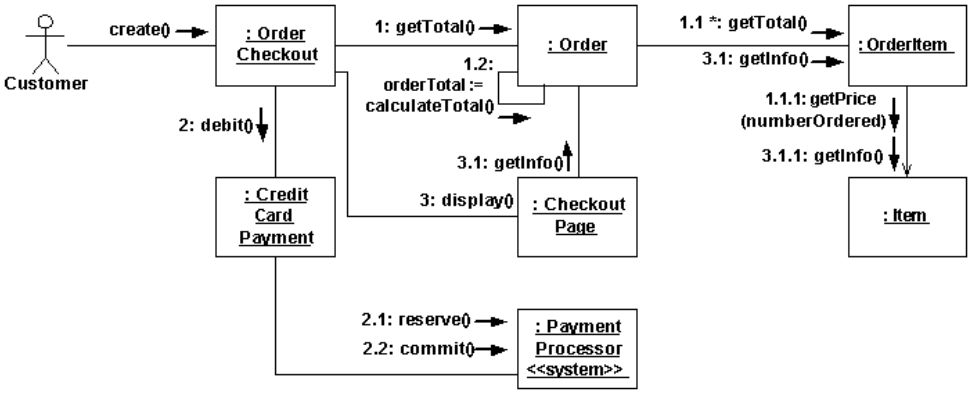
Timing diagram adalah artifact baru ditambahkan ke UML 2.0 dan digunakan untuk menampilkan interaksi antara objek. Tujuan dari diagram ini adalah untuk menggali perilaku salah satu atau lebih benda yang diberikan selama jangka waktu tertentu. VPUML memberikan banyak fitur untuk membantu Anda membuat diagram dan waktu selesai untuk menyesuaikan dengan mudah. Artikel ini akan memperkenalkan beberapa komponen utama dari waktu ke diagram dan menampilkan langkah demi langkah cara membuat diagram waktu selesai.contoh gambar:



Gambar 3.1. timing diagram

15. Collaboration diagram

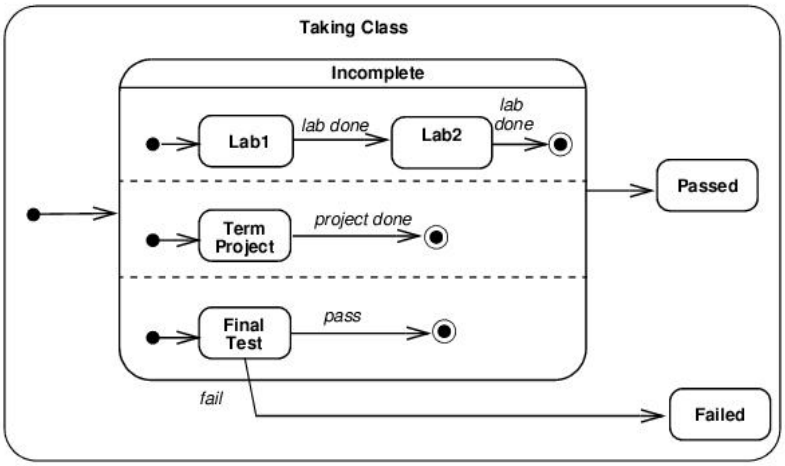
Collaboration diagram juga menggambarkan interaksi antar objek seperti sequence diagram, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian message.Setiap message memiliki sequence number, di mana message dari level tertinggi memiliki nomor 1. Messages dari level yang sama memiliki prefiks yang sama. Contoh gambar:



Gambar 3.2. collaboration diagram

16. statechart diagram

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya) suatu objek pada sistem sebagai akibat dari stimuli yang diterima. Pada umumnya statechart diagram menggambarkan class tertentu (satu class dapat memiliki lebih dari satu statechart diagram). Dalam UML, state digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar state umumnya memiliki kondisi guard yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. Action yang dilakukan sebagai akibat dari event tertentu dituliskan dengan diawali garis miring. Titik awal dan akhir digambarkan berbentuk lingkaran berwarna penuh dan berwarna setengah. Contoh gambar:



Gambar 3.3. Statechart diagram

Adapun metode pembuatan software aplikasi chatting pada perangkat handphone ini menggunakan bluetooth dan software Java. Sun Microsystem telah mendefinisikan tiga platform java, yang masing-masing diarahkan untuk tujuan tertentu dan untuk lingkungan komputasi yang berbeda-beda:

a. Java Standard Edition (J2SE) : Didesain untuk jalan pada komputer desktop.
b. Java Enterprise Edition (J2EE) : Dengan built-in mendukung untuk servlets, JSP, dan XML, edisi ini ditujukan untuk aplikasi berbasis server.
c. Java Micro Edition (J2ME) : Didesain untuk piranti dengan memori terbatas, layar display terbatas dan power pemrosesan yang juga terbatas.

Seperti yang telah disebutkan sebelumnya, J2ME dirancang untuk dapat menjalankan program java pada perangkat-perangkat semacam handphone dan PDA, yang memiliki karakteristik yang berbeda dengan sebuah komputer biasa, misalnya kecilnya jumlah memori pada handphone.

2.2.1.3.1 Konsep Dasar Komponen

J2ME memiliki komponen sebagai berikut:

a. Java Virtual Mechine (JVM). Komponen ini merupakan lingkungan tempat eksekusi program Java berlangsung dimana objek saling berinteraksi. Virtual Machine menyebabkan Java mempunyai kemampuan penanganan memori yang lebih baik, keamanan yang lebih tinggi serta portabilitas yang besar. Pada CLDC, ini disebut K Virtual Machine, menggambarkan singkatan kilo byte yang bahwa Virtual Machine ini berkerja pada lingkungan yang sangat kecil.
b. Java Aplicatioan Programming Interface (Java API). Merupakan kumpulan library untuk menjalankan dan mengembangkan pada emulator atau handled devices.

c. Tools lain untuk pengembangan aplikasi Java, seperti emulator

2.2.1.4 Konsep Dasar Arsitektur

J2ME terbagi dalam 2 bagian yaitu :

A. J2ME Configuration adalah spesifikasi yang mendefenisikan lingkungan kerja J2ME (dapat berupa JVM lengkap atau bagian tertentu) dan kumpulan API dasar yang mengimplementasikan fitur standard handled devices.

Terdapat 2 kategori J2ME Configuration :

1. Connected Limited Devices Configuration (CLDC)

   Konfigurasi untuk perangkat wireless (handphone, organizer / PDA, two way pages) dengan memori kecil (160 – 512 KBites) dengan koneksi jaringan yang tetap / tidak. Keterbatasan pada CLDC menyebabkan beberapa fitur pada J2SE tidak disertakan. Fitur – fitur yang tidak disertakan tersebut seperti:

- Floating Point yang terdapat dalam kelas java.lang.Double dan java.lang.float
- Finalisasi Objek yang terdapat dalam kelas java.lang.object
- JNI yang memungkinkan untuk mengakses library yang dibuat tidak dengan bahasa java
- Error Exception yang terdapat dalam kelas java.lang.error , java.lang.OutMemory dan java.lang.VirtualMachineError

  Pada CLDC juga ditambahkan fitur yang tidak terdapat pada J2SE. Fitur – fitur khusus ini seperti :

- Generic Connection Framework, merupakan kelas yang digunakan sebagai framework koneksi secara umum. Ini dilakukan dengan memperlakukan koneksi seperti Uniform Resource Locator (URL). Kelas ini digunakan untuk melakukan koneksi ke file, protocol HTTP, socket atau port.

- Kelas javax.microedition yang merupakan API untuk implementasi pada wireless programming.
- Mendefenisikan hierarki tipe koneksi dasar untuk koneksi jaringan, sedangkan defenisi protocol jaringan dilakukan oleh CDC.

2. Connected Device Configuration (CDC)

   Konfigurasi untuk perangkat wireless dengan memori 2 MBites lebih (Set Tob Box, Internet Enabled Screen Phones, Internet TV, High – End Communicator, Car Televition, Auto Mobile Entertainment / Navigation System) dengan koneksi jaringan yang lebih kokoh / dapat kebeberapa jaringan.

B. J2ME Profile adalah spesifikasi yang mendefenisikan implementasi tambahan yang sangat spesifik dari Handled Devices. Terdapat 5 kategori J2ME Profile :

- Mobile Information Devices Profile (MIDP). Profil yang menyediakan library Java untuk implementasi dasar antarmuka (GUI), jaringan (Networking), basisdata (Database), penyimpanan tetap (persistant strage), daur hidup aplikasi (Aplication life cicle), dan timer (Personal Information Management). MIDP dirancang khusus untuk wireless phone dan pager serta dibangun berdasarkan CLDC.
- Foundation Profile (FP). Profile dasar untuk non – GUI network devices pada CLDC dengan ROM 1024 k dan RAM 512 k.
- Personal Profile. Profile dengan GUI yang memilki tingkat keakuratan web tinggi dan mampu menjalankan applet

- RMI Profile
- PDA Profile

2.2.1.5 Konsep Dasar Midlet

Aplikasi yang berjalan diatas MIDP disebut MIDlet. Aplikasi ini tidak berinteraksi langsung dengan hardware tapi melalui program khusus yang disebut Application Management Software (AMS). Pengaplikasian API ditentukan oleh interaksi antara Aplikasi dengan MIDlet serta bagaimana MIDlet dikontrol. MIDlet terdiri atas :

a. MIDlet Suite dan Descriptor.

Aplikasi dalam J2ME dikemas dalam file *.JAR dan selanjutnya terdapat Descriptor yang disebut application descriptor berupa file *.JAD untuk mendeskripsikan file JAR tersebut.

b. Java Class

c. Resource

Siklus hidup (LifeCycle) dari sebuah MIDlet ditangani oleh AMS. AMS adalah program bawaan (built in) yang melakukan penanganan eksekusi aplikasi atau sebagai lingkungan tempat MIDlet diciptakan, dijalankan, dihentikan dan dihancurkan. AMS disebut juga dengan Java Application Manager (JAM). MIDlet memiliki beberapa state (MIDlet State) yaitu Pause, Active, Destroy.

Nama Status	Keterangan
Paused	Status ini terjadi ketika MIDlet selesai diinisialisasi dan tidak melakukan aksi apapun atau terjadi kesalahan ketika akan melakukan *start.* Pada status ini, MIDlet tidak boleh sedang mengunci sumber daya karena akan mengakibatkan proses lain tidak dapat menggunakan sumber daya tersebut.
Active	Status ini terjadi ketika MIDlet sedang aktiv / berjalan dengan normal
Destroyed	Status ini terjadi ketika MIDlet berhenti berjalan, sehingga sumber daya yang digunakan dibebaskan.

Tabel 2.2.1.4 Daftar jendela-jendela di dalam Midlet State

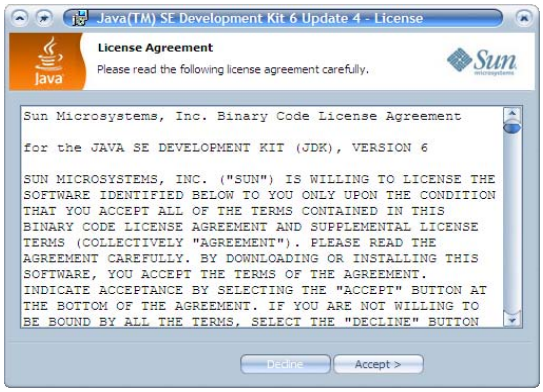
2.2.2 IDE NetBeans pada J2ME

Aplikasi MIDP (mobile), MIDlet juga dapat dirancang dengan NetBeans. Untuk itu memerlukan NetBeans 6.0 Mobility Pack. Dan juga instalasi sun_java_wireless_toolkit-2_5_1-windows sebagai J2ME.Di dalam NetBeans,semua perancangan dan pemrograman dilakukan di dalam kerangka sebuah proyek. Proyek NetBeans merupakan sekumpulan file yang dikelompokkan di dalam satu kesatuan. Pada umumnya sebuah proyek, beserta file-file isinya, akan ditempatkan di dalam satu direktori. Meskipun demikian ada juga proyek yang dapat menangani file-file yang terpisah direktorinya.

2.2.2.1 Instalasi

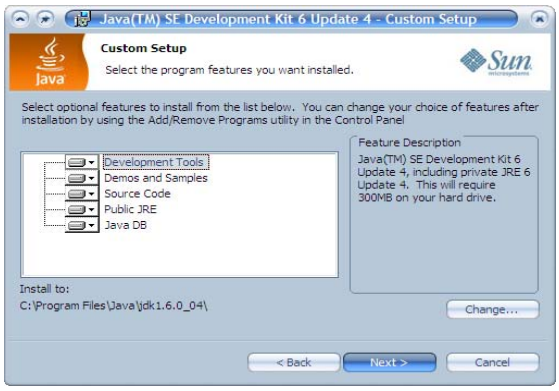
Berikut adalah langkah-langkah untuk menginstal **JDK-6u4-windows-i586 p.exe** telah terinstal dengan langkah sebagai berikut:

a. Klik dua kali ikon **JDK-6u4-windows-i586-p.exe.** Installer akan mengekstrak file JDK dan membuat direktori yang diperlukan, dan menyalin file-file kedalam direktori tersebut. Tahapan instalisasi dimulau dengan tampilan lisensi JDK. Klik Accept (Gambar 2.2.3.1.0).



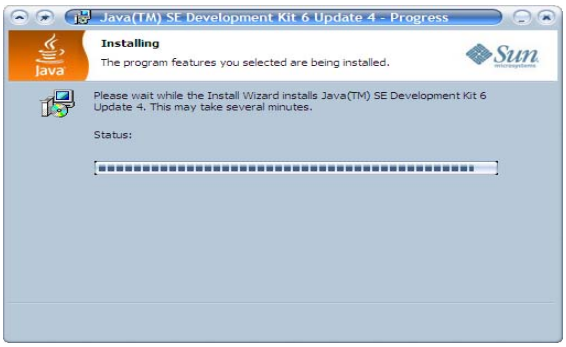
Gambar 2.2.3.1.0 lisensi

b. Selanjutnya, instalisasi akan menawarkan fitur apa saja yang ingin diinstal. Silahkan pilih fitur yang ingin diinstal, disarankan untuk menginstal semua fitur yang ditawarka.(Gambar 2.2.3.1.1). klik Next.



Gambar 2.2.3.1.1 Fitur instalasi JDK

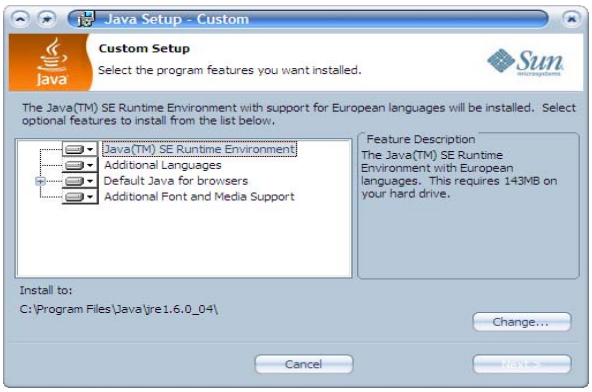
c. Proses instalasi sedang berlangsung (Gambar 2.2.3.1.2)



Gambar 2.2.3.1.2 Proses instalasi JDK

d. Saat proses instalasi JDK berlangsung, akan muncul tampilan untuk menginstal J2SE Runtime Evironment, yang merupakan lingkungan Java Virtual Machine. Klik Next.

e. Selanjutnya instalasi akan menawarkan fitur apa saja yang diinstal. Klik Next.



Gambar 2.2.3.1.3 Fitur instalasi JRE

f. Instalisasi J2SE Runtime Evironment sedang berlangsung. Setelah proses instalasi berakhir, instalasi JDK pun berakhir.



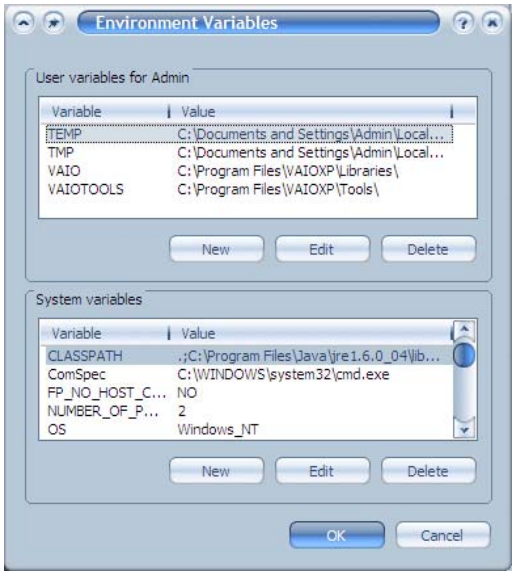
Gambar 2.2.3.1.4 Proses instalasi JRE



Gambar 2.2.3.1.5 Proses instalasi selesai

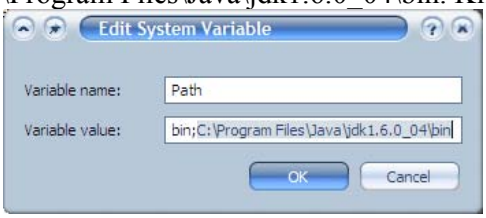
Setelah menginstal JDK lakukan langkah selanjutnya yaitu setting Path sebagai berikut:

a. Klik kanan pada **My Komputer -> Properties.** Pilih tab **Advanced -> Environment Variables.**



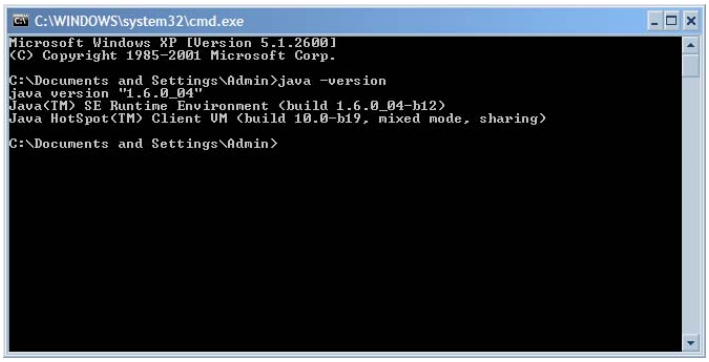
Gambar 2.2.3.1.6 Setting Path

b. Di bagian **Sistem variables** pilih **Path.** Klik Edit. Tambahkan direktori tempat JDK diinstal, yaitu c:\Program Files\Java\jdk1.6.0_04\bin. Klik Ok



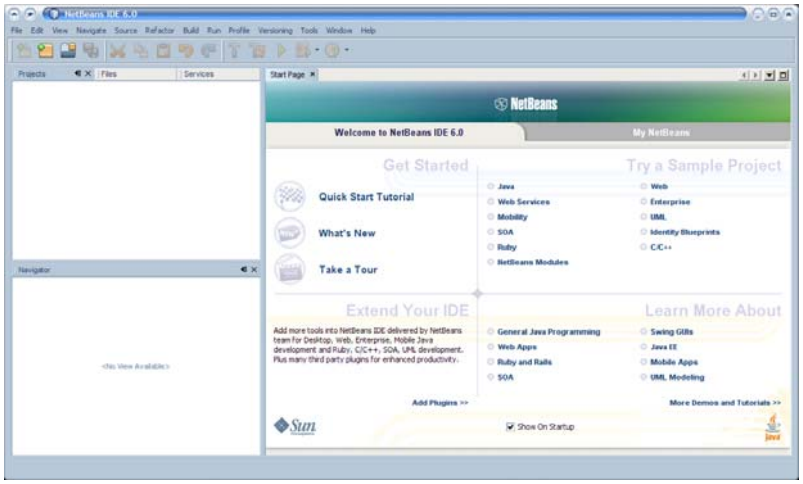
Gambar 2.2.3.1.7 Edit Variabel Path

Dan langkah selanjutnya pengujian JDK, caranya buka Command Prompt dan ketikkan **java –version** (gambar 2.2.3.1.8)



Gambar 2.2.3.1.8 Cek java version

2.2.2.2 Jendela-jendela IDE



***Gambar* 2.2.3.2.0** Tampilan IDE NetBeans

Untuk menampilkan berbagai macam jendela, pakailah menu Window. Misalnya untuk memunculkan palet komponen, jalankan menu Window | GUI Editing Palette.

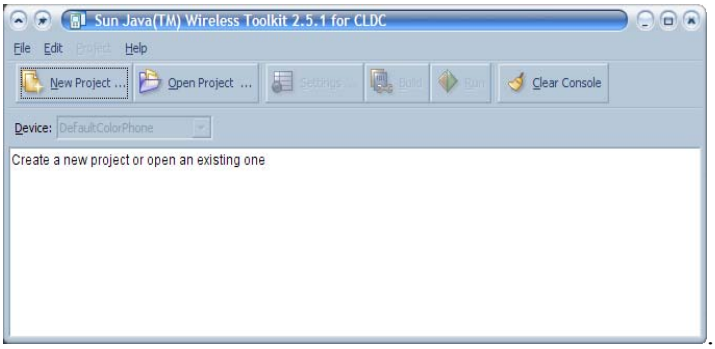
Daftar jendela-jendela pokok dari NetBeans dapat dilihat dalam table 2.2.3.1.

Jendela	**Keterangan**
Projects	Jendela terpenting yang berisi daftar dari semua kandungan proyek, merupakan *logical-view* dari isi proyek, seperti daftar paket dan web-page. Klik kanan pada sebuah node proyek akan menampilkan menu popup yang berisi operasi-operasi yang dapat dilakukan terhadap proyek.
Files	Jendela ini menampilkan daftar structural dari file-file yang tidak Nampak dalam projects. Dari jendela ini sebuah file sumber dapat dibuka untuk diedit. File sumber dapat berupa file Java, file HTML, file konfigurasi proyek, selain file sumber, juga didaftar file-file output hasil proses build. Misalnya file "class", JAR, WAR, file Javadoc.
Navigator	Jendela ini menampilkan informasi mengenai file-file sumber yang berada dalam proyek yang sedang aktif. Ditampilkan elemen-elemen file dalam bentuk daftar atau bentuk pohon (*inheritance-tree).*
Runtime	Jendela ini menampilkan informasi runtime, seperti proses yang sedang berjalan, sesi debugging, aneka

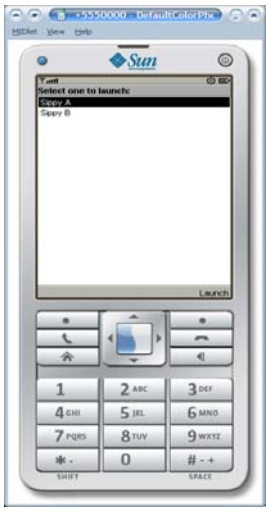
	servis yang terhubung ke IDE.
Source-Editor	Editor teks untuk mengedit file-file sumber. Editor ini dapat dibuka dengan klik ganda pada sebuah node dalam jendela Projects.
Matisse GUI Builder	Sebuah editor visual untuk merancang dan mengedit form GUI tampilan dari program atau proyek java yang sedang dibuat. Form ini mendefinisikan tampilan design-time berupa komponen-komponen GUI dan layout. Form ini juga membuka jendela palette, jendela inspector dan jendela properties.
GUI Editng | Palette	Jendela Palette menampilkan daftar semua komponen yang terinstal didalam NetBeans. Misalnya komponen yang berasal dari paket Swing, AWT,Beans. Juga daftar layout yang dapat dipakai untuk mengatur format susunan komponen. Anda juga dapat menambahkan komponen atau paket sendiri.
GUI Editing | Inspector	Jendela Inspector menunjukan struktur dari form. Daftar layout dan komponen yang ada dalam form ditampilkan dalam bentuk pohon(*inheritance-tree).*
Properties	Menampilkan daftar properti dan tingkah-laku (behavior) dari sebuah komponen, layout atau form.
Debugging	NetBeans memiliki JDPA debugger. Jendela debugger menampilkan informasi runtime dari program java yang sedang dieksekusi. Kita juga dapat mengawasi, mendefinisikan atau melakukan breakpoint, watch, step-through, *vriable-examination,class-examination,*serta *calstack,thread* dan *session.*
Favorites	Jendela ini dapat dipakai untuk memudahkan pengaksesan sebuah lokasi dalam sistem komputer, seperti file-file dan direktori yang berada diluar proyek.
Output	Jendela ini akan menampilkan output dari program jika ada. Misalnya program menuliskan teks ke standar *output-stream.* Maka tampilannya dapat dilihat di jendela ini. Selain itu, hasil dari proses build (atau kompilasi) juga akan ditampilkan disini. Kita dapat melihat pesan keberhasilan atau pesan error disini.

Table 2.2.3.2.1 Daftar jendela-jendela pokok yang ada di dalam NetBeans

Dan berikut adalah tampilan dari sun_java_wireless_toolkit-2_5_1-windows sebagai pengoperasian J2ME + Emultor.



***Gambar* 2.2.3.2.2** Tampilan sun_java_wireless_toolkit-2_5_1-windows



***Gambar* 2.2.3.2.3** Tampilan Emilator pada sun_java_wireless_toolkit

Fungsi Emilator pada sun_java_wireless_toolkit yaitu untuk menjalankan MIDlet yang bekerja pada lingkungan J2ME *toolkit*, MIDlet beroperasi pada *toolkit* saja karena emilator ini dapat langsung melakukan mengeksekusi program dan menampilkan dalam bentuk screen.