# BAB II

# TINJAUAN PUSTAKA

## II.1. Kecerdasan Buatan

Kecerdasan buatan adalah suatu ilmu yang mempelajari cara membuat komputer melakukan sesuatu seperti yang dilakukan manusi (Minsky,1989). Defenisi lain diungkapkan oleh H.A. Simon (1987). Kecerdasan buatan (*artificial intelligence*) merupakan kawasan penelitian, aplikasi dan instruksi yang terkait dengan pemograman komputer untuk melakukan sesuatu hal yang dalam pandangan manusia adalah cerdas.

Rich and knight (1991) mendefenisikan kecerdasan buatan (*Al*) sebagai suatu studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapat dilakukan lebih baik oleh manusia (Kusrini; 2006 :3).

## II.2. Sistem Pakar

Sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan,fakta, dan teknik penalaran dalam memcahkan masalah yang biasanya dapat dipecahkan oleh seorang pakar dalam bidang tersebut (Martin Dan Oxman, 1988).

Pada dasarnya sistem pakar diterapkan untuk mendukung aktivitas pemecahan masalah. Beberapa aktivitas pemecahan yang dimaksud antara lain : pembuatan keputusan (*decion making*), pemanduan pengetahuan (*knowledge fusing*), pembuatan desain (*designing*), perencanaan (*planning*), prakiraan

(*forecasting*), pengaturan (*regulating*), pengendalian (*controlling*), diagnosis (*diagnosing*), perumusan (*prescribing*) dan penjelasan (*expalaining*), pemberian nasihat (*advising*)dan pelatihan (*tutoring*). Selain itu sistem pakar juga dapat berfungsi asisten yang pandai dari seorang pakar (Kusrini; 2006 :11).

### II.2.1. Sejarah Sistem Pakar

*Expert System (ES)* mulai dikembangkan pada pertengahan tahun1960-an oleh *Artificial Inteligence Corporation*. Periode penelitian Artificial Inteligence ini didominasi oleh suatu keyakinan bahwa nalar yang digabung dengan komputer canggih akan menghasilkan prestasi pakar. Suatu usaha kearah ini adalah *General Purpose Problem-Solver (GPS).* GPS merupakan sebuah *predecessor* menuju *Expert System (ES).*

Pada pertengahan tahun 1960-an, terjadi penggantian dari program serba bisa *(general-purpose)* ke program yang spesialis *(special-purpose).* Dengan dikembangkannya DENDRAL oleh E. Feigenbaum dan kemudian diikuti oleh MYCIN.

Awal 1980-an, teknologi ES yang mula-mula dibatasi oleh suasana akademis mulai muncul sebagai aplikasi komersil, khususnya XCON, XSEL (dikembangkan dari R-1 pada digital Equipment Corp.) dan CATS-1 (dikembangkan oleh General Electric).

Sistem Pakar untuk melakukan diagnosis kesehatan telah dikembangkan sejak pertengahan tahun 1970. Sistem pakar untuk melakukan diagnosis pertama dibuat oleh Burce Buchanandan Edward Shortliffe di Stanford University. Sistem ini diberi nama MYCIN (Kusrini; 2006 : 12-13).

### II.2.2. Pemakai Sistem Pakar

Sistem pakar dapat digunakan oleh :

1. Orang awam yang bukan pakar untuk meningkatkan kemampuan mereka dalam memecahkan masalah.
2. Pakar sebagai asisten yang berpengetahuan.
3. Memperbanyak atau menyebarkan sumber pengetahuan yang semakin langka.

Sistem pakar merupakan program yang dapat menggantikan keberadaan seorang pakar. Alasan mengapa ES dikembangkan untuk menggantikan seorang sistem pakar **:**

1. Dapat menyediakan kepakaran setiap waktu dan di berbagai lokasi
2. Secara otomatis mengerjakan tugas-tugas rutin yang membutuhkan seorang pakar.
3. Seorang pakar akan pensiun atau pergi.
4. Menghadirkan/menggunakan jasa seorang pakar memerlukan biaya yang mahal.
5. Kepakaran dibutuhkan juga pada lingkungan yang tidak bersahabat (*hostile enviroitment*) (Kusrini; 2006 :14).

### II.2.3. Ciri-ciri Sistem Pakar

1. Terbatas pada bidang yang spesifik.
2. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti.

3. Dapat mengemukakan rangkaian alasan yang diberikannya dengan cara dapat dipahami.
4. Berdasarkan pada *rule* atau kaidah tertentu.
5. Dirancang untuk dapat dikembangkan secara bertahap.
6. Outputnya bersifat nasehat atau anjuran.
7. Output tergantung dari dialog dengan user.
8. *Knowledge base* dan *inference engine* terpisah (Kusrini; 2006 :14-15).

### II.2.4. Keuntungan Pemakaian ES

1. Membuat seorang yang awam dapat bekerja seperti layaknya seorang pakar.
2. Dapat bekerja dengan informasi yang tidak lengkap atau tidak pasti.
3. Meningkatkan *output* dan produktivitas. ES dapat bekerja lebih cepat dari manusia. Keuntungan ini berarti mengurangi jumlah pekerja yang dibutuhkan dan akhirnya akan mereduksi biaya.
4. Meningkatkan kualitas.
5. ES menyediakan nasehat yang konsisten dan dapat mengurangi tingkat kesalahan.
6. Membuat peralatan yang kompleks lebih mudah dioperasikan karena ES dapat melatih pekerja yang tidak berpengalaman.
7. Handal (*reliability*).
8. ES tidak dapat lelah atau bosan. Juga konsisten dalam member jawaban dan selalu memberikan perhatian penuh.
9. Memiliki kemampuan untuk memecahkan masalah yang kompleks.

10. Memungkinkan pemindahan pengetahuan ke lokasi yang jauh serta memperluas jangkauan seorang pakar, dapat diperoleh dan dipakai dimana saja. Merupakan arsip yang terpercaya dari sebuah keahlian sehingga *user* seolah-olah bekonsultasi langsung dengan sang pakar meskipun sang pakar sudah pensiun (Kusrini, S. Kom; 2006 :15).

### II.2.5. Arsitektur Sistem Pakar

Sistem pakar memiliki beberapa komponen utama yaitu antarmuka (*user interface*), basis data sistem pakar (*expert system database*). Fasilitas akusisi pengetahuan (*knowledge acquisition facility*), dan mekanisme inferensi (*inference mechanism*). Selain itu ada satu komponen yang hanya ada beberapa sistem pakar yaitu fasilitas penjelasan (*explanation facility*), (Martin dan Oxman, 1988).

Antarmuka pengguna adalah perangkat lunak yang menyediakan media komunikasi antara pengguna dengan sistem.

Basis data sistem pakar berisi pengetahuan setingkat pakar pada subyek tertentu. Berisi pengetahuan yang dibutuhkan untuk memahami, merumuskan, dan menyelesaikan masalah. Basis data ini terdiri dari 2 elemen dasar :

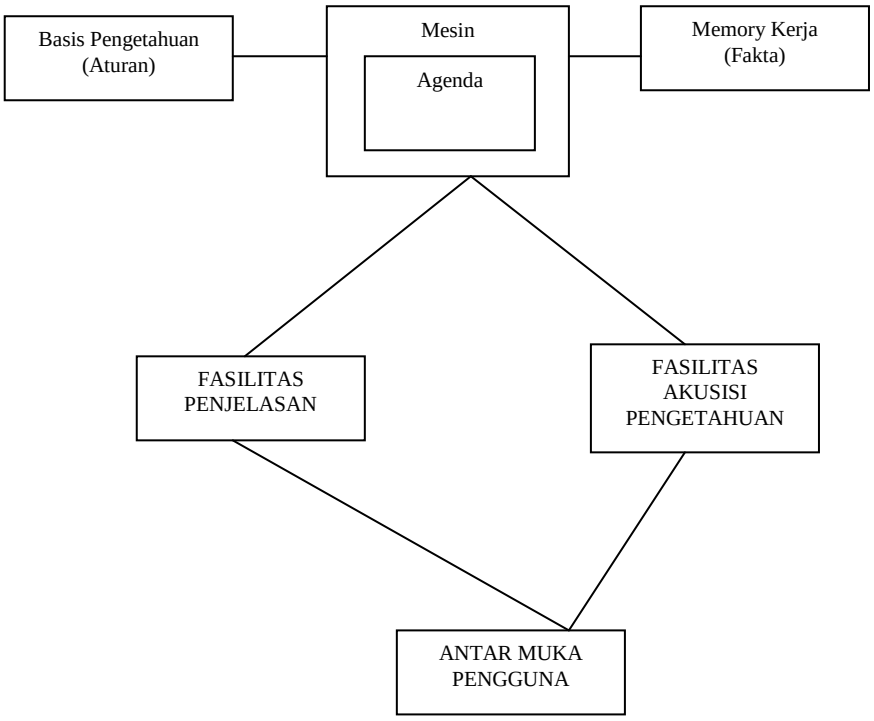
1. Fakta, situasi masalah dan teori yang terkait.
2. Heuristik khusus atau *rules*, yang langsung menggunakan pengetahuan untuk menyelesaikan masalah khusus.

Fasilitas penjelasan berguna dalam memberikan penjelasan kepada pengguna mengapa komputer meminta suatu informasi tertentu dari pengguna dan dasar apa yang digunakan komputer sehingga dapat menyimpulkan suatu kondisi.

Ada 4 tipe penjelasan yang digunakan dalam sistem pakar yaitu (Schupp, 1989) :

1. Penjelasan mengenai jejak aturan yang menunjukan status konsultasi.
2. Penjelasan mengenai bagaimana sebuah keputusan diperoleh.
3. Penjelasan mengapa sistem menanyakan suatu pertanyaan.
4. Penjelasan mengapa sistem tidak memberikan keputusan seperti yang dikehendaki pengguna.

Arsitektur dasar dari sistem pakar dapat dilihat pada gambar II.1. (Giarrantano dan Riley, 1994).




Gambar II.1. Arsitekur Sistem Pakar

Sumber : Kusrini (2006:19)

Memori kerja dalam arsitektur sistem pakar merupakan bagian dari sistem pakar yang berisi fakta-fakta masalah yang ditemukan dalam suatu sesi, fakta-

fakta tentang suatu masalah yang ditemukan dalam proses konsultasi (Kusrini; 2006: 17-19).

### II.2.6. Inferensi

Inferensi merupakan proses untuk menghasilkan informasi dari fakta diketahui atau diasumsikan. Inferensi adalah konklusi logis (*logical conclusion*) atau implikasi berdasarkan informasi yang tersedia.

Dalam sistem pakar, proses inferensi dilakukan dalam suatu modul yang disebut *inference engine* (mesin inferensi).

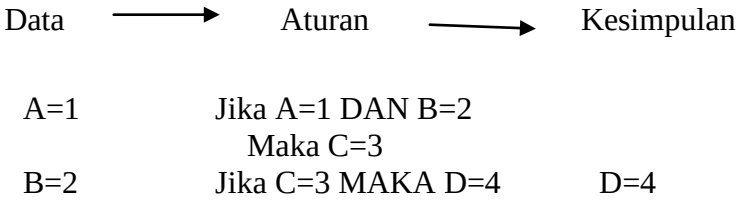
Ketika rerpresentasi pengetahuan (RP) pada bagian *knowladge base* telah lengkap, atau paling tidak telah berada pada level yang cukup akurat, maka RP tersebut telah siap digunakan. Inferensi engine merupakan modul yang berisi program tentang bagaimana mengendalikan proses *reasoning*.

Ada dua metode inferensi yang penting dalam sistem pakar yaitu runut maju (*forward chaining*), dan runut balik (*backward chaining*) (Kusrini; 2006:35).

1. **Runut Maju (*forward chaining*).**

   Runut maju berarti menggunakan himpunan aturan kondisi aksi. Dalam metode ini, data digunakan untuk menentukan aturan mana yang akan dijalankan. Kemudian aturan tersebut dijalankan. Mungkin proses menambahkan data ke memori kerja. Proses diulang sampai ditemukan suatu hasil.

   Gambar II.2. berikut ini menunjukan bagaimana cara kerja metode inferensi runut maju.




Gambar II.2. Runut Maju

Sumber : Kusrini (2006:36)

Metode runut maju cocok digunakan untuk menangani masalah pengendalian (*controlling*) dan peramalan (*prognogsis*) (Giaratatano dan Riley, 1994).

Berikut ini adalah contoh inferensi dengan menggunakan metode runut maju :

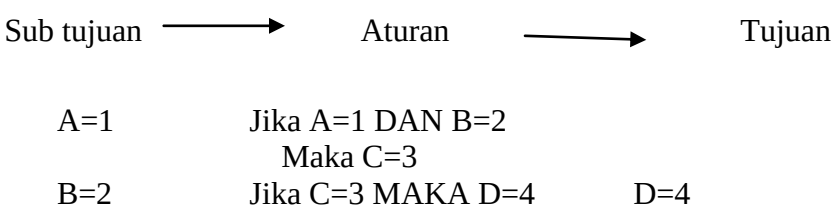
Jika penderita terkena penyakit epilepsy idiopatik dengan CF antara 0,4 s/d 0,6.

Maka berikan obat carbamazipine.

**2. Runut Balik (*backward chaining*).**

Runut balik merupakan metode penalaran kebalikan dari runut maju. Dalam runut balik, penalaran dimulai dengan tujuan merunut balik ke jalur yang akan mengarahkan ke tujuan tersebut (Giaratatano dan Riley, 1994).

Gambar II.3. berikut ini menunjukan proses penalaran menggunakan metode runut balik.




Gambar II.3. Runut Balik

Sumber : Kusrini (2006:37)

Runut balik disebut juga sebagai *goal driven reasoning*. Merupakan cara yang efesien untuk memecahkan masalah yang dimodelkan sebagai masalah pemilihan terstruktur. Tujuan dari inferensi ini adalah mengambil pilihan terbaik dari banyak kemungkinan. Metode inferensi runut balik ini cocok digunakan untuk memecahkan masalah diagnosis (Kusrini; 2006 : 36-37).

## II.3. Penyakit Paru-paru

### 1. Nyeri Dada

a. Pendahuluan : Merupakan presentasi utama penyakit dan sering menjadi indikasi penyakit yang serius. Kesalahan diagnosa nyeri dada secara potensial mematikan, dan riset di pakai untuk mengembangkan algoritme yang akurat dan efektif dari segi biaya bagi masalah ini. Etiologi banyak penyakit dapat menyebabkan nyeri dada : jantung, par, gastrointestinal, muskulosletal, dan vascular. Banyak etiologi mempunyai presentase yang tumpang tindih, dan diagnosis sering kali memerlukan tes tambahan. Etiologi paru nyeri dada sering tidak bias di pisahkan dari keluhan pasien pada presentasi. Hampir semua ahli sepakat tanpa adanya trauma, penyebab nyeri dada akut di anggap karena jantung sampai terbukti sebaliknya. Nyeri dada kronis tidak mempunyai persyaratan ini, namun tetap harus di dekati secara seksama.

b. Patofisiologi : sumber utama nyeri dada akut meliputi jantung, esophagus, vasculator paru, dan pleura. Pada sebagian besar pengalaman praktisi nyeri dada karena muskuloskelet sangat umum.

Organ visceral mempunyai persyaratan ganda dengan serabut-serabut viseralis dan somatic. Persyaratan visceral untuk organ di dalam dada adalah nevrus vagus. Serabut-serabut somatic merupakan aferen spinalis. Interaksi kedua persyaratan ini bertanggung jawab akan persepsi nyeri viseralis pada permukaan kulit, kesukaran melokalisasi rasa nyeri, dan nyeri yang dialihkan. Dinding dada hanya mempunyai persyaratan somatik. Walaupun dahulu di anggap saluran pernapasan tidak mempunyai reseptor nyeri ini kemungkinan tidak benar, dan cabang trakeobronkialis dapat memberikan reaksi terhadap iritasi terhadap sensasi nyeri.

c. Karakteristik : Lokasi : subternum, panggung, difus, lateral bahu, lengan unilateral, bilateral. Karakter : penumpuan rasa terbakar, pleuritik (berkaitan dengan pernapasan), rasa di robek, rasa penuh/begah, pita konstristik merupakan istilah yang umum. Gejala penyerta : mual, sesak napas, eksaserbasi, sendawa, kolaps kardiaovaskular, hemopstisis.eksaserbasi : aktivitas, makanan, posisi. Meringankan : istirahat, nitrat, antacid, posisi.

d. Pemeriksaan Lanjutan : Sebagian besar pasien yang dating dengan nyeri dada mendadak pada pusat perawatan darurat akan menjalani elektrokardiogram, foto toraks, dan enzim jantung termasuk troponim. Indicator biokimia pada cedera jantung seperti troponim mempunyai keterbatasan pada kedua istilah sensitivitas spesifilitas (J. Emergmed;2002;23;57).

e. Terapi Khusu : Opiate, nitrat untuk myeri karena jantung; opiate pleura karena embrio paru, NSAID untuk nyeri pleura kostokondritik bila di perlukan. Nyeri esophagus berespons terhadap nitrat, antacid; PPI dan penghambatan H2 sangat membantu pada nyeri akut dan kronis. Nyeri karena keganasan atau invasi atau invasi atau infeksi tulang, pleura memerlukan opiate dosis tinggi dan teknik kusus seperti blok syaraf. Neuralgia neuropatik dan pascaherpes dapat di obati dengan gabapentin (*Cohrane Database Syst Rev 2005; CD005152*) dan kapsaisin ( *J.neurol 1991;238 : 452*), walaupun kapsaisin hanya bekerja pada jumlah pasien tertentu (Edward Ringel ; 2012 : 8-10)

**2. Batuk**

a. Pendahuluan : Batuk merupakan gejala yang sering di temukan pada penyakit pernapasan, penyakit pernapasanatas, dan penyakit saluran pencernaan. Pada pemeriksaan batuk pertanyaan utama adala :

1).Apakah batuk berasal dari saluran pernapasan ?

2).Apakah batuk merupakan suatu kelainan, atau merupakan manifestasi masalah yang serius ?

3).Langkah tepat apakah yang sesuai untuk mencapai diagnosis ?

Karena dengan banyaknya gejala paru lainnya, perusahaan tetap memberikan jawaban. Batuk merupakan gejala yang tidak spesifik yang sedikit memberi petunjuk pada dokter. Ketika batuk merupakan satu gejala yang

ditemukan, perjalanan penyakit dapat juga di pantau sebagai sati indicator keberhasilan/kegagalan pengobatan ketika pasien diobati.

b. Patofisiologi : Saluran pernapasan mempunyai beberapa alat untuk mengeksperesikan ketidak senangnya atau iritasinya. Saluran pernapasan dan parenkim paru mempunyai beberapa resptor, tetapi batuk merupakan respon utama paru terhadap rangsangan bahaya. Reseptor iritan di seluruh saluran pernapasan dapat memicu batuk sebagai satu usaha untuk membersihkan materi-materi berbahaya. Hanya iritasi struktur dengan persarafan Vagus mempresipitasi batuk involunter. Mekanik batuk adalah ekspresi paksa terhadap glottis yang tertutup tiba-tiba terbuka dan memungkinkan kecepatan aliran udara sangat tinggi melalui aluran pernapasan dan agak membawa bahan-bahan berbahaya di dalamnya.

c. Karakteristik : perbedaan utamanya adalah antara kronis dan akut. Batuk kronis dengan berbagai cara di tetapkan sebagai lamanya 3-8 minggu. Batuk mendadak sering berkaitan dengan infeksi, inhalasi atau aspirasi, atau aktivitasi/eksaserbasi penyakit paru later/kronis (asma/PPOK). Pada pasien asma, batuk persisten, di malam hari yang tidak sempurna. Hemopti akut atau kronis merupakan benang merah dan meningkatkan kemungkinan keganasan proses yang merusak jaringan paru-paru ( seperti abses paru atau TB ). Merokok merubah ekologi penyakit dan meningkatkan kemungkinan PPOK dan keganasan.

d. Diagnosis Deferensial : Pembanding batuk akut terbatas : bronchitis, pneumonia, asma eksaserbasi, PPOK eksaserbasi, benda asing jarang,batuk sering di temukan pada PJK, EP, namun jarang di temukan sebagai satu keluhan.

e. Pengobatan Langsung : Pertama kenali dan tangani masalah yang mendasarinya. Batuk tidak dapat di tekan tanpa suatu diagnosis. Dikatakan bahwa pemantauan gejala dapat sangat membantu dalam waktu singkat, sementara terapi defenitif di jalankan atau untuk meringankan ketika masalah dasarnya dapat diobati. Pada keadaan tertentu kemungkinan dibutuhkan sebagai tambahan jangka panjang. Dekstronetorfan adalah efektif dan bukan narkotik. Narkotik umumnya efektif, tetapi ada resiko kecanduan (*Thorax* 2004; 59 :438, Murray dan Nadel 2005;1 : 831). Pasien usia lanjut menjadi goyah karena efek sedasi dan mencederai diri mereka. Pada keadaan tertentu, narkotika untuk batuk dapat memicu gagal napas pada pasien dengan retensi CO2 atau sesak waktu tidur ( *sleep apnea*) yang tidak di obati (Edward Ringel ; 2012 :10-12).

**3. Sesak Napas**

a. Pendahuluan : Sesak napas merupakan manifestasi dasar penyakit. Dengan berbagai cara di gambarkan sebagai haus udara, napas pendek, tidak mampu menarik napas dalam, dan banyak keluhan lainnya. Sesak merupakan suatu manifestasi ganguan interprestasi keseimbangan pengiriman oksigen ke jaringan. Tugas diagnostic

utama adalah untuk membdakan sesak paru dari sesak bukan karena paru. Sekalipun sesak paru cirri gejalanya sangat penting. Sesak dalam satu bentuk atau lainnya menyertai sebagian besar penyakit pernapasan. Pemantauan dan pengukuran sesak sangat berharga penangan penyakit. Terkadang di butuhkan untuk mengobati gejala sesak di bandingkan mengobati akar masalah.

b. Patofisiologi : Otak mengharapkan hubungan tertentu di antara tekanan oksigen darah dan tekanan karbon dioksida darah, reseptor regang dinding dada, kebutuhan oksigen jaringan, pengiriman oksigen, dan kerja pernapasan. Ganguan keseimbangan menyebabkan sesak napas. Pada praktiknya ada empat kondisi utama yang menyebabkan sesak napas. Penyakit oaru, penyakit jantung, anemia, dan dekondisi. Persepsi pasien akan sensai berbahaya, sangat bervariasi dari pasien ke pasien.

c. Karakteristik : dapat sebagai kronis/akut, episodic/konstan, progresif/stabil. Ortopnea adalah sesak napas pada posisi telentang platipnea adalah sesak napas yang di hasilkan dengan bangkit atau berdiri. Ada banyak upaya mengembangkan pengukuran objektif seperti skala Brog, kecuali masalah psikologis dan peseptual dapat saya temukan dan sangat bermanfaat untuk mengukur, di tinjau dari sudut aktivitas hidup sehari-hari. Asosiasi jantung amerika menggunakan Grade I sampai IV dlam menggolongkan sesak napas (*Cleveland Clin J Med* 2003; 70:89), yang sangat bermanfaat

dalam penggolongan parahnya penyakit(I tidak ada gangguan; II sesak napas aktivitas biasa; III sesak pada aktivitas ringan; IV sesak saat istirahat).

d. Diagnosa Deferensial : Sesak karena paru biasanya berkaitan dengan gejala lainnya. Sesak karena jantung berkaitan dengan gejala jantung lainnya. Dekondisi dan anemia dapat menyerupai keduanya namun lebih sering terlihat karena jantung. Keluhan menggerakkan lengan lebih sukar dari kaki berkaitan dengan penyakit paru ( N Engl J Med 1986; 314: 1485). Sesak akut menimbulkan pertimbangan akan PE, MI, gagal napas akut, atau pneumia karena berbagai etiologi. Seperti gejala paru lainnya, sesak dievakuasi oleh perusahaan yang mempertahankannya. Pada praktiknya sulit untuk membedakan esak karena jantung dari sesak karena paru, dan keduanya seringkali sering bersama-sama pada dekondisi.

e. Pemeriksaan Lanjutan : Untuk algoritme yang di usulkan. Akut : Minuman di perlukan anamnesis dan pemeriksaan menyeluruh, fototoraks, gas darah arteri,elektrokardiografi, dan hitung darah lengkap (CBC). Diagnosis sering kali jelas pada langkah pertama ini. Edema paru, insufisiensi mitral (MI), asma, penyakit paru obstruktif kronis eksaserbasi akut (AECOPD), pneumonia, sepsis dan emblio paru (PE) merupakan presentasi paling umum sesak akut. Lihat emboli paru untuk pembahasan lengkap peran D-

Drimer. Pada semua kasus fototoraks (CXR) mempunyai peran penting dalam menentukan arah evaluasi. Pertimbangkan CPET (lihat pemeriksaan aktivitas kardivaskular) untuk mendiagnosis sesak kronis yang sukar di sembuhkan.

f. Pengobatan Langsung : Opiat sangat baik untuk mengendalikan sesak napas penyakit paru lanjut, juga benzodiazepine. Oksigen juga di ketahui sangat bermanfaat (Edward Ringel ; 2012 : 12-16).

**4. mengi (*Wheezing*)**

a. Pendahuluan : Mengi merupakan suara paru tambahan yang di hasilkan semata-mata saluran pernapasan. Mengi didefenisikan sebagai bunyi/saluran berkelanjutan (terus menerus) yang di hasilkan saluran pernapasan besar dan laring. Bunyi dengan frekuensi dominant 400 Hz atau lebih tinggi di sebut mengi; bunyi dengan frekuensi dominant 200 Hz atau lebih rendah di defenisikan sebagai ronki.

b. Patofisiologi : periset mempunyai hipotesis bahwa mengi di hasilkan sesuatu getaran pancaran udara (suling), atau dengan kemungkinan mempunyai komponen keduanya, namun lebih mendekati buluh yang bergetar (klarinet). Mekanisme ini kompleks dan kemungkinan mempunyai komponen keduanya, namun lebih mendekati buluh yang bergetar. Pada keadaan yang tepat dinding saluran pernapasan dan kolam udara tertutup bergetar, dan

terciptalah suara. Sifat suara bukanlah petunjuk besar kecil sumbatan pernapasan yang dapat di percaya.

c. Karakteristik : Dengan adanya patofisiologi yang di uraikan di atas, pertanyaannya adalah apakah ada di temukan mengi atau tidak. Walaupun mengi inspirasi dilaporkan berhubungan dengan obstruksi saluran pernapasan bawah. Monofonik menunjukkan satu tempat obstruksi; polifonik menunjukkan banya tempat obstruksi. Mengi monofonik persisten sejati harus segera mencari sati obstruksi anatomi.

d. Diagnosis Diferensial : Dari sudut praktis, asma dan COPD erupakan sumber mengi yang paling sering. Dokter harus berhati-hati akan VCD, diagnosis yang salah satu dapat menyebabkan pengobatan selama berbulan-bulan bahkan bertahun-tahun yang tidak diperlukan. Fibrosis berat dapat menyebabkan “derita” yang dapat menyerupai mengi. Endema paru juga dapat mencetuskan terjadinya mengi

e. Pemeriksaan Lanjutan : Anamnesis yang cermat akan mengarahkan pemeriksaan lanjut. Kebanyakan penyakit paru obstruktif mempunyai pola penampilan tersendiri (lihat Asma, COPD, Bronkiektasis). Foto dan CT-Scan dapat membantu mendiagnosis bronkiektasis, perubahan emfisema, atau masa paru. Bronkoskopi akan sangat membantu bila di pikirkan ada kelainan anatomi

saluran pernapasan besar. Ketika seorang pasien tidak sesuai dengan kategori pikiran masalah psikogenik.

f. Pengobatan Simtomatik : Satu-satunya pendekatan yang terlihat adalah pengobatan yang mendasarinya (Edward Ringel ; 2012 :16-18).

**5. Sputum**

a. Pendahuluan : Produksi sputum merupakan gejala yang tidak khas pada banyak penyakit paru. Tujuan utama evaluasi produksi sputum adalah untuk mengenali materi yang di hasilkan dan selanjutnya mencoba untuk untuk menentukan penyebabnya. Terkadang “sputum” berasal dari saluran pernapasan atas, sehingga sangat penting satu pertanyaan singkat, untuk memastikan pasien benar-benar membantukan materi dari saluran pernapasan bagian bawah.

b. Patofisiologi : Mukus merupakan komponen normal saluran pernapasan, membentuk lapisan pelindung dari endotel pernapasan. Hopersekresi mucus merupakan kelainan, dengan stimulasi berlebihan dari sel-sel goblet dan sumber mucus dan submukosa. Sangat banyak mediator hipersekresi diidentifikasi dan berhubungan erat dengan berbagai komponen peradangan, seperti leukotrienm, prostaglandinm dan histanim; enzim bakteri, dan protease, terutama yang berasal dari neutrofil.

c. Karakterisasi : Warna : Jernih/putih biasanya sangat menunjukkan sel-sel peradangan. Kuning/hijau berhubungan dengan sel-sel nanah, (sputum dapat purulen pada pneumonia viru, bronchitis virus). Hijau terang berhubungan dengan pseudomonas. Sputum coklat, merah beruhubungan dengan darah, kerusakan jaringan dalam praktik hal ini jarang di temui, bahkan dalam praktik spesialis. Bau : dengan sengaja mencium sputum tidaklah benar dan bertentangan dengan prinsip kewaspadaan bersama. Bau seperti buah anggur berkaitan dengan pseudomonas. Kelainan yang terlibat secara makroskopik juga mungkin terjadi : Parasit yang dapat dilihat (mis. Askaris), dan sputum banyak "lapisan" klasik dari bronkiektasi adalah dua contohnya. Makanan, benda asing yang jelas merupakan tanda aspirasi yang utama
d. Diagnosa Diferensial : Pasien tidak mungkin mengacaukan sputum dengan sekresi dari system organ tubuh lainnya. Terkadang, dokter perlu membedakan dari raba hidung, isi lambung. Perbedaan utama berdasarkan gejala penyerta. Pada umumnya sputum terutama penampilan sputum akut, harus dianggap berbahaya sampai terbukti sebaliknya.
e. Pemeriksaan Lanjut : Pemeriksaan primernya terutama tergantung pada pemeriksaan mikroskopik sputum, biakan sputum, terkadang pemeriksaan lainnya seperti reaksi rantai polimerasi (PCR). Ditemukannya sel-sel batang bronkus, makrofag alveolus, dan

neutrofil menunjukkan specimen saluran pernapasan bawah. Sel-sel skuamosa, material yang diludahkan dengan konsistensi seperti air liur. Beberapa penulis merasa bahwa sputum eosinofilia sangat bermanfaat dalam diagnosis asma, bronchitis eosinofilia kronis (Chest 2006; 1229 :1344). Kesembuhan dari agen jamur dan parasit tidak pasti.

f. Pengobatan Simtomatik : Mukolitik seperti guafenesis 600 mg diminum dua kali (po bid), pelembaban, bronkodilator (agonis beta dan antikolinergik), preparat iodium (SSKI), inhalasi asetilsistein, inhalasi alfa dornase (hanya pasien CF). bersihkan di perbesar dengan alat-alat mekanik juga : katub geletar, fsioterapi dada getar (Edward Ringel ; 2012 : 18-19)

## II.4. Kamus Data

Kamus data (KD) atau data dictionary (DD) yang disebut juga dengan system dictionary data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan demikian KD, analisis sistem dapat mendefenisikan data yang mengalir di sistem dengan lengkap. KD dibuat pada pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem. Pada tahap analisis, KD dapat digunakan sebagai alat komunikasi antara analisis sistem dengan pemakai sistem tentang data yang mengalir di sistem. Pada tahap perancangan sistem KD digunakan untuk merancang input, merancang laporan-laporan dan database. KD dibuat

berdasarkan arus data yang ada di DAD. Arus data di DAD sifatnya global, hanya ditunjukkan nama arus datanya saja (Jogiyanto, HM; 2005 : 725).

### II.4.1. Isi Kamus Data

KD harus dapat mencerminkan keterangan yang jelas tentang data yang dicatatnya. Untuk maksud keperluan ini maka KD harus memuat hal-hal berikut ini :

1. Nama Arus Data

   Karena KD dibuat berdasarkan arus data yang mengalir di DAD, maka nama dari arus data juga harus dicatat di KD, sehingga mereka yang membaca DAD dan memerlukan penjelasan lebih lanjut tentang suatu arus data tertentu di DAD dapat langsung mencarinya dengan mudah di KD.

2. Alias

   Alias atau nama lain dari data dapat dituliskan bila nama lain ini ada. Alias perlu ditulis karena data yang mempunyai nama yang berbeda untuk orang atau departemen satu dengan yang lainnya. Misalnya bagian pembuat faktur dan langganan menyebut bukti penjualan sebagai faktur, sedang bagian gudang menyebutnya sebagai tembusan permintaan persediaan barang. Baik faktur dan tembusan permintaan persediaan ini mempunyai struktur data yang sama tetapi mempunyai struktur yang berbeda.

3. Bentuk Data

   Telah diketahui bahwa arus data mengalir.

   a. Dari kesatuan luar ke suatu proses, data yang mengalir ini biasanya tercatat di suatu dokumen atau formulir.

b. Hasil dari suatu proses ke kesatuan luar, data yang mengalir biasanya terdapat di media laporan atau *query* tampilan layar atau dokumen hasil cetakan komputer.
c. Hasil dari suatu proses yang lain, data yang mengalir biasanya dalam bentuk variabel atau parameter yang dibutuhkan oleh proses penerimanya.
d. Hasil dari suatu proses yang direkamkan ke simpanan data, data yang mengalir ini biasanya berbentuk suatu variabel.
e. Dari simpanan data dibaca oleh suatu proses, data yang mengalir ini biasanya berupa suatu *field* (item data).

Dengan demikian bentuk dari data yang mengalir dapat berupa :

1). Dokumen dasar atau formulir.
2). Dokumen hasi cetakan komputer
3). Laporan tercetak.
4). Tampilan dilayar monitor.
5). Variabel.
6). Parameter
7) *Field*

Bentuk dari data ini perlu dicatat di KD, karena dapat digunakan untuk mengkelompokan KD ke dalam kegunaanya, sewaktu perancangan sistem KD yang mencatat data yang mengalir dalam bentuk dokumen dasar atau formulir yang digunakan untuk merancang bentuk input sistem. KD yang mencatat data yang mengalir dalam bentuk laporan tercetak dan dokumen hasil cetakan

komputer akan digunakan untuk merancang *output* yang akan dihasilkan oleh sistem. KD yang mencatat data yang mengalir dalam bentuk tampilan layar monitor akan digunakan juga untuk merancang tampilan layar yang akan dihasilkan oleh sistem. KD yang mencatat data yang mengalir ke dalam bentuk parameter dan variabel akan digunakan untuk merancang proses dari program. KD yang mencatat data yang mengalir ke dalam bentuk dokumen, formulir, laporan, dokumen cetakan komputer, tampilan di layar monitor, variabel, dan *field* akan digunakan untuk merancang database.

4. Arus Data

   Arus data menunjukkan dari mana data yang mengalir dan kemana data akan menuju. Keterangan arus data ini perlu dicatat di KD supaya memudahkan mecari arus data di DAD (Jogiyanto, HM; 2005 : 726-727).

## II.5. Normalisasi

Normalisasi adalah proses pengkelompokkan attribute-attribute dan suatu relasi sehingga membentuk *Well- Structure Relation.* Normalisasi merupakan proses pengkelompokkan elemen data menjadi suatu tabel-tabel menunjukan entity dan relasinya. Normalisasi ditemukan pada tahun 1970 oleh E. F. CODD.

a. Bentuk tidak normal (*Unnormalized Form*)

   Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti suatu format tertentu. Dapat saja data tidak lengkap atau terduplikasi. Data dikumpulkan apa adanya dengan saat menginput.

   Contoh data :

Tabel II.1. Bentuk Tidak Normal (*Unnormalized Form*)

| No_Siswa | Nama | Pa | Kelas 1 | Kelas 2 | Kelas 3 |
|---|---|---|---|---|---|
| 22890100 | Shandy | Linda | 1234 | 1543 | 1543 |
| 22890101 | Susi | Riska | 1234 | 1775 | |

Sumber :Linda Marlinda (2004 : 122)

Siswa yang punya nomor siswa, nama, dan pa mengikuti 3 mata pelajaran/kelas. Di sini ada perulangan kelas 3 kali ini bukan bentuk tidak 1 NF.

b. Bentuk Normal Ke Satu (1 NF/ *Fisrt Normal Form*)

Suatu relasi 1 NF dan hanya jika sifat dan setiap relasi attributenya bersifat *atomic*. Atom adalah zat terkecil yang masih memiliki sifat induknya. Bila dipecah lagi maka ia tidak memiliki sifat induknya.

Ciri-ciri 1 NF :

1. Setiap data dibentuk kedalam *flat file* data terbentuk per satu *record* nilai dan field berupa "*atomic value*".
2. Tidak ada *set attribute* yang berulang atau bernilai ganda,
3. Tiap *field*
4. hanya satu pengertian.

Tabel II.2. Bentuk Normal Ke Satu (1 NF/ *Fisrt Normal Form*)

| No_Siswa | Nama | Pa | Kelas 1 |
|---|---|---|---|
| 22890100 | Shandy | Linda | 1234 |
| 22890100 | Shandy | Linda | 1543 |
| 22890101 | Susi | Riska | 1234 |
| 22890101 | Susi | Riska | 1775 |
| 22890101 | Susi | Riska | 1543 |

Sumber :Linda Marlinda (2004 : 122)

c. Bentuk Normal Ke Dua (2 NF/ *Second Normal Form*)

Bentuk normal kedua mempunyai syarat yaitu bentuk data telah memenuhi kriteria bentuk normal kesatu. *Attribute* bukan kunci haruslah bergantung

secara fungsi pada *primary key*. Jadi, untuk membentuk normal kedua haruslah sudah ditentukan kunci-kunci *field.* Kunci *field* haruslah unik dan dapat mewakili *attribute* lain yang menjadi anggotanya. Misal : dari contoh relasi siswa pada 1 NF terlihat bahwa *primary key* adalah nomor siswa. Nama siswa dan pa bergantung fungsi pada no_siswa, tetapi kode_kelas bukanlah fungsi dan siswa dipecah menjadi 2 relasi :

Relasi siswa :

Tabel II.3. Bentuk Normal Kedua Relasi Siswa (2 NF/ *Second Normal Form*)

| **No_Siswa** | **Nama** | **Pa** |
|---|---|---|
| 22890100 | Shandy | Linda |
| 22890101 | Susi | Riska |

Sumber :Linda Marlinda (2004 : 123)

Relasi ambi_Kelas

Tabel II.4. Bentuk Normal Kedua Relasi ambil_Kelas (2 NF/ *Second Normal Form*)

| **No_Siswa** | **Kode Kelas** |
|---|---|
| 22890100 | 1234 |
| 22890100 | 1543 |
| 22890101 | 1234 |
| 22890101 | 1775 |
| 22890101 | 1543 |

Sumber :Linda Marlinda (2004 : 123)

d. Bentuk Normal Ketiga (3 NF/*Third Normal Form*)

Untuk menjadi bentuk normal ketiga maka relasi dalam bentuk normal kedua dan semua *attribute* bukan primer tidak punya hubungan yang transistif. Dengan kata lain, setiap *attribute* bukan kunci haruslah bergantung hanya pada *primary key* dan *primary key* secara menyeluruh.

Contoh pada bentuk normal kedua diatas termasuk juga bentuk normal ketiga karena seluruh attribute yang ada bergantung penuh pada kunci primernya.

e. *Boyee-Cood Normal Form* (BCNF)

BNCF mempunyai paksaan lebih kuat dan bentuk normal ketiga untuk menjadi BCNF, relasi harus dalam bentuk normal kesatu dan setiap *attribute* harus bergantung fungsi pada *attribute superkey.*

Pada contoh di bawah ini terdapat relasi seminar dengan ketentuan sebagai berikut :

1. Kunci primer adalah no_siswa +seminar
2. Siswa bopleh mengambil satu atau dua seminar.
3. Setiap siswa dibimbing oleh salah satu di antara 2 instruktur seminar tersebut.
4. Setiap instruktur boleh hanya mengambil satu seminar saja.

Pada contoh ini no_siswa dan seminar menunjuk seorang instruktur :

Relasi seminar

Tabel II.5. *Boyee-Cood Normal Form* (BCNF)

| No_Siswa | Seminar | Instruktur |
|---|---|---|
| 22890100 | 2281 | Si doel |
| 22890101 | 2281 | Pak tile |
| 22890102 | 2291 | Mandra |
| 22890101 | 2291 | Basuki |
| 22890109 | 2291 | Basuki |

Sumber :Linda Marlinda (2004 : 124)

Bentuk relasi seminar adalah bentuk normal ketiga, tetapi tidak BCNF karena nomor seminar masih tergantung fungsi pada instruktur. Jika setiap instruktur dapat mengajar hanya pada satu seminar saja, maka seminar bergantung fungsi

pada satu *attribute* bukan *superkey* seperti disyaratkan oleh BCNF. Maka relasi seminar haruslah dipecah menjadi dua yaitu :

Tabel II.6. *Boyee-Cood Normal Form* (BCNF)

Relasi Pengajar

| Instruktur | Seminar |
|---|---|
| Si doel | 2281 |
| Pak tile | 2281 |
| Mandra | 2291 |
| Basuki | 2291 |

| No_Siswa | Instruktur |
|---|---|
| 22890100 | Si doel |
| 22890101 | Pak tile |
| 22890102 | Mandra |
| 22890101 | Basuki |
| 22890109 | Basuki |

Sumber :Linda Marlinda (2004 : 124)

f. Bentuk normal keempat (4NF)

   Relasi R adalah bentuk 4 NF jika dan hanya jika relasi tersebut juga termasuk BCNF dan semua ketergantungan *multivaluei* adalah juga ketergantungan fungsional.

g. Bentuk normal kelima (5 NF)

   Disebut juga PINF (*Projection Join Normal Form*) dan 4 NF dilakukan dengan menghilangakan ketergantungan *join* yang merupakan kunci kandidat (Linda Marlinda; 2004 : 122-125).

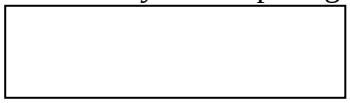
### II.5.1. ERD (*Entity Relationship Diagram*)

Merupakan suatu model untuk menjelaskan hubungan antar dua dalam basis data berdasarkan suatu persepsi bahwa *real word* terdiri dari *object-object* dasar yang memmpunyai hubungan atau antar *object-object* tersebut. Relasi antar *object* dengan menggunakan symbol-simbol grafis tertentu.

Model *entity relationship* adalah suatu penyajian dengan menggunakan *entity* dan *relationship*. Diperkenalkan pada tahun 1976 oleh P.P. Chen (Linda Marlinda; 2004 : 17).

### II.5.2. Komponen-komponen yang terdapat didalam *Entity Relationship* Model.

1. ***Entity***
   a. Adalah sesuatu yang dapat dibedakan dalam dunia nyata dimana informasi yang berkaitan dengannya dikumpulkan.
   b. *Entity* set adalah kumpulan *entity* yang sejenis.
   c. Symbol yang digunakan untuk *entity* adalah persegi panjang.
   d. *Entity* set dapat berupa :
   1. *Entity* yang bersifat fisik, yaitu *entity* yang dapat dilihat.
      Contohnya : rumah, kendaraan, mahasiswa, dosen, dan lain-lain.
   2. *Entity* yang bersifat konsep atau logic, yaitu *entity* yang tidak dapat dilihat.
      Contohnya : pekerjaan, perusahaan, rencana, mata kuliah, dan lain-lain.
   e. Simbol yang digunakan untuk *entity* adalah persegi panjang.

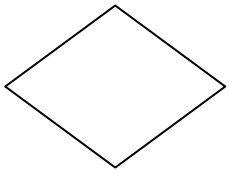


Gambar II.4. *Entity*
Sumber : Linda Marlinda (2004:17)

2. ***Relationship***
   a. Adalah hubungan yang terjadi antara satu atau lebih *entity*.
   b. *Relationship* tidak mempunyai keberadaan fisik, kecuali yang mewarisi hubungan antara *entity* tersebut.

c. *Relationship* set adalah kumpulan relationship yang sejenis.

d. Simbol yang digunakan adalah bentuk belah ketupat, *diamond* atau *rectangle.*



Gambar II.5. *Relationship*

Sumber : Linda Marlinda (2004:18)

3. **Attribute**

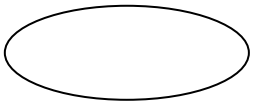
a. Adalah karakteristik dari *entity* atau *relationship* yang menyediakan penjelasan detail tentang atau *relationship* tersebut.

b. Attribute value (nilai atribute) adalah suatu data aktual atau informasi yang disimpan di suatu atribute di dalam suatu *entity* atau *relationship*.

c. Terdapat dua jenis atribute, yaitu :

1. *Indetifer (key)*, untuk menentukan suatu *entity* secara unik.
2. *Descriptor (nonkey attribute)*, untuk menentukan karakteristik dari suatu *entity* yang tidak unik.

d. Simbol yang digunakan adalah bentuk oval.



Gambar II.6. *Attribute*

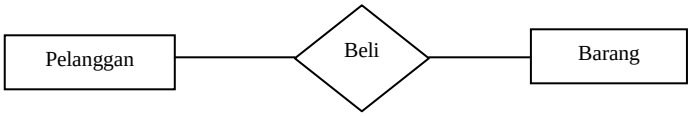
Sumber : Linda Marlinda (2004:18)

4. Indicator Tipe

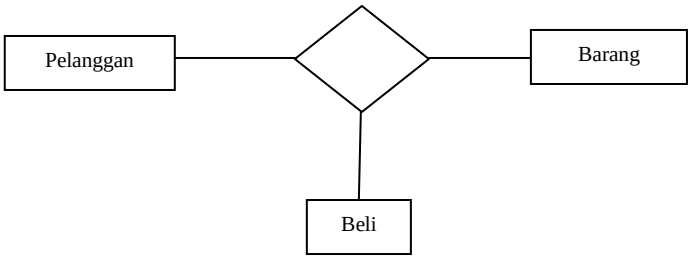
a. *Indicator tipe associative object*

Berfungsi sebagai suatu objek dan suatu *relationship*

Contoh :




Menjadi :



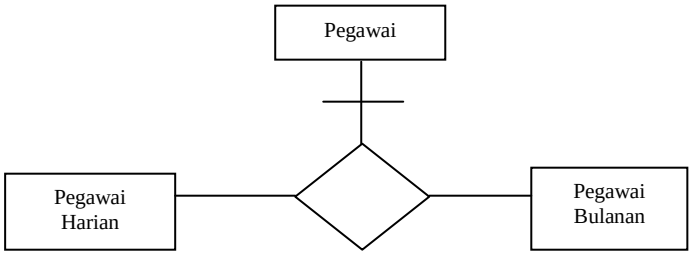

Gambar II.7. Indicator Tipe

Sumber : Linda Marlinda (2004:19)

b. Indicator tipe supertipe

Terdiri dari suatu object dan sub kategori atau lebih yang dihubungkan dengan dihubungkan dengan relationship yang tidak bernama (Linda Marlinda; 2004 : 17-19).

Contoh :




Gambar II.8. Indicator Tipe SuperTipe

Sumber : Linda Marlinda (2004:19)

## II.6. *Unified Modeling Language* (UML)

UML singkatan dari *Unified Modelling Langguage* yang berarti bahasa pemodelan standart. (Chonoles; 2003 : 6) mengatakan sebagai bahasa, berarti *UML* memiliki sintaks dan *semantic*. Ketika kita membuat model menggunakan konsep *UML* ada aturan –aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat harus berhubungan satu dengan lainnya harus mengikuti standart yang ada. *UML* bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. Ketika pelanggan memesan sesuatu dari sistem, bagaimana transaksinya? Bagaimana sistem mengatasi error yang terjadi? Bagaimana keamanan terhadap sistem yang ada kita buat? Dan sebagainya dapat dijawab dengan *UML*.

*UML* diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

*UML* telah diaplikasikan dalam investasi perbankan,lembaga kesehatan, departemen pertahanan, sistem terdistribusi, sistem pendukung alat kerja, retail, sales, dan supplier.

Blok pembangunan utama *UML* adalah diagram. Beberapa diagram ada yang rinci (jenis *timing diagram*) dan lainya ada yang bersifat umum (misalnya diagram kelas). Para pengembang sistem berorentasikan objek menggunakan bahasa model

untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka rancang. *UML* memungkinkan para anggota team untuk bekerja sama dalam mengaplikasikan beragam sistem. Intinya, *UML* merupakan alat komunikasi yang konsisten dalam mensuport para pengembang sistem saat ini. Sebagai perancang sistem mau tidak mau pasti menjumpai *UML*, baik kita sendiri yang membuat sekedar membaca diagram *UML* buatan orang lain (Prabowo Pudji Widodo, Herlawati; 2011 : 6-7).

### II.6.1. *Diagram-Diagram UML*

Beberapa literatur menyebutkan bahwa *UML* menyediakan Sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa yang digabung, misalnya diagram komunikasi, diagram urutan, dan diagram pewaktuan digabung menjadi diagram interaksi. Namun demikian model-model itu dapat dikelompokan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain :

1. Diagram Kelas. Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi, serta relasi-relasi diagram. Diagram ini umu dijumpai pada pemodelan sistem berorentasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas.
2. Diagram paket (*Package Diagram*) bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas merupakan bagian dari diagram komponen.
3. Diagram *Use Case* bersifat statis. Diagram ini memperlihatkan himpunan *use-case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini

terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.

4. Diagram interaksi dan *Sequence* (urutan). Bersifat dinamis. Diagram urutan adal;ah diagram interaksi yang menekankan pada pengiriman pesan dalam waktu tertentu.
5. Diagram komunikasi (*Communication Diagram*) bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi *UML* yang menekankan organisasi *structural* dari objek-objek yang menerima serta mengirim pesan.
6. Diagram *Statechart* (*Statechart Diagram)* bersifat dinamis. Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (*State)*, transisi kejadian serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka (*interface*), kelas, kolaborasi dan terutam penting pada pemodelan sistem-sistem yang reaktif.
7. Diagram aktivitas (*Activity Diagram*) bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan member tekanan pada aliran kendali antar objek.
8. Diagram komponen (*Component Diagram*) bersifat statis. Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan diagram kelas dimana komponen

dipetakan kedalam satu atau lebih kelas-kelas. Antarmuka-antarmuka serta kolaborasi-kolaborasi.

9. Diagram *Deployment* (*Deployment Diagram)* bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run time)*. Memuat simpul-simpul berserta komponen-komponen yang ada di dalamnya. Diagram *Deployment* berhubungan erat dengan diagram komponen dimana diagram ini memuat satu atau lebih komponen-komponen. Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing).*

Kesembilan diagram ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semuanya dibuat sesuai dengan kebutuhan. Pada *UML* dimungkinkan kita menggunakan diagram-diagram lainya misalnya *Data Flow Diagram, Entity Relationship Diagram* dan sebagainya (Prabowo Pudjo Widodo, Herlawati; 2011 : 10-12).

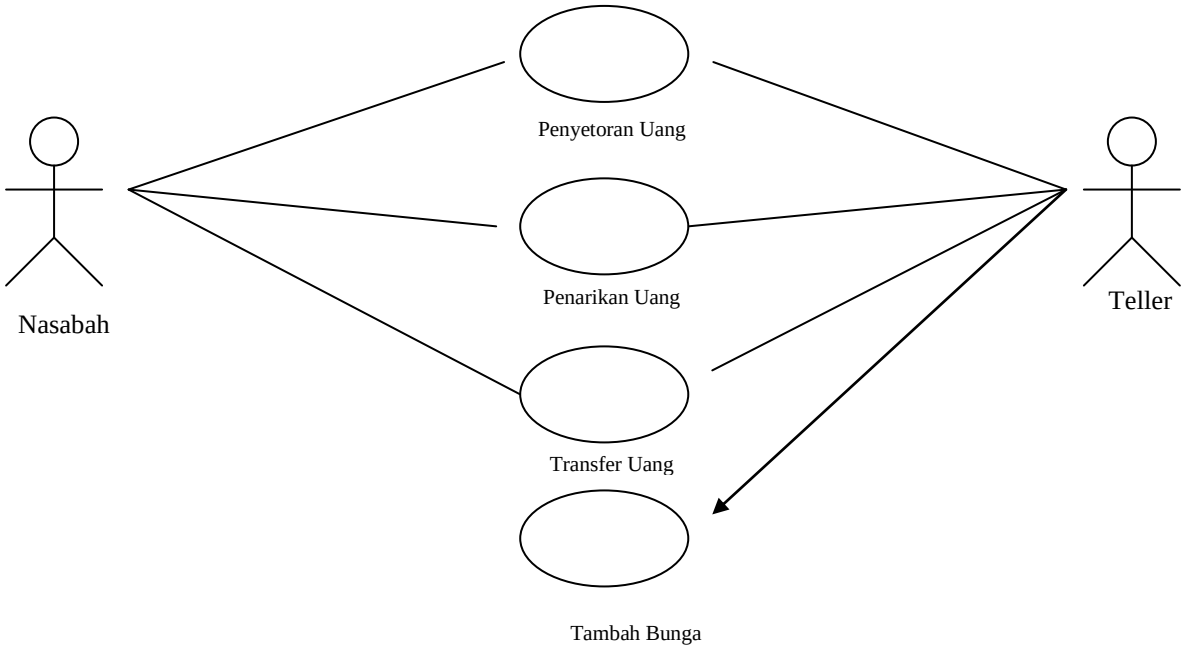
1. *Diagram Use Case (Use Case Diagram)*

   *Use Case* menggambarkan *external view* dari sistem yang akan kita buat modelnya. Menurut Pooley (2005:15) mengatakan bahwa model *use case* dapat dijabarkan dalam diagram, tetapi yang perlu diingat, diagram tidak identik dengan model karena model lebih luas dari diagram.

   komponen pembentuk diagram *use case* adalah :

   a. Aktor (*actor),* menggambarkan pihak-pihak yang berperan dalam sistem.
   b. *Use Case,* aktivitas/ sarana yang disiapkan oleh bisnis/sistem.
   c. Hubungan (*Link),* aktor mana saja yang terlibat dalam *use case* ini.

Gambar di bawah ini merupakan salah satu contoh bentuk diagram *use case* (Prabowo Pudji Widodo, Herlawati; 2011 : 16-17).

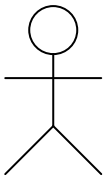



Gambar II.9. Diagram *Use Case*

Sumber : Prabowo Pudjo Widodo, Herlawati (2011:17)

**2. Aktor**

Menurut Chonoles (2003 :17) menyarankan sebelum mebuat use case dan menentukan aktornya, agar mengidentifikasi siapa saja pihak yang terlibat dalam sistem kita. Pihak yang terlibat biasanya dinamakan *stakeholder*.

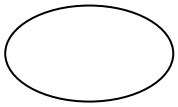


Gambar II.10. Aktor

Sumber : Prabowo Pudjo Widodo, Herlawati (2011:17)

**3. *Use Case***

Menurut Pilone (2005 : 21) *use case* menggambarkan fungsi tertentu dalam suatu sistem berupa komponen kejadian atau kelas. Sedangkan menurut Whitten (2004 : 258) mengartikan *use case* sebagai urutan langkah-langkah yang secara tindakan saling terkait (skenario) baik terotomatisasi maupun secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal. *Use case* digambarkan dalam bentuk *ellips/*oval



Gambar II.11. Simbol *Use Case*
Sumber : Prabowo Pudjo Widodo, Herlawati (2011:22)

*Use case* sangat menentukan karakteristik sistem yang kita buat, oleh karena itu Chonoles (2003:22-23) menawarkan cara untuk menghasilkan *use case* yang baik yakni :

a. Pilihlah Nama Yang Baik

*Use case* adalah sebuah *behaviour* (prilaku), jadi seharusnya dalam frase kata kerja. Untuk membuat namanya lebih detil tambahkan kata benda mengindikasikan dampak aksinya terhadap suatu kelas objek. Oleh karena itu diagram *use case* seharusnya berhubungan dengan diagram kelas.

b. Ilustrasikan Perilaku Dengan Lengkap**.**

*Use case* dimulai dari inisiasi oleh aktor primer dan berakhir pada aktor dan menghasilkan tujuan. Jangan membuat *use case* kecuali anda mengetahui tujuannya. Sebagai contoh memilih tempat tidur (*King Size,*

*Queen Size*, atau dobel) saat tamu memesan tidak dapat dijadikan *use case* karena merupakan bagian dari *use case* pemesanan kamar dan tidak dapat berdiri sendiri (tidak mungkin tamu memesan kamar tidur jenis king tapi tidak memesan kamar hotel).

c. Identifikasi Perilaku Dengan Lengkap**.**

Untuk mencapai tujuan dan menghasilkan nilai tertentu dari aktor, *use case* harus lengkap. Ketika memberi nama pada *use case*, pilihlah frasa kata kerja yang implikasinya hingga selesai. Misalnya gunakan frasa *reserve a room* (pemesanan kamar) dan jangan *reserving a room* (memesan kamar) karena memesan menggambarkan perilaku yang belum selesai.

d. Menyediakan *Use Case* Lawan (*Inverse)*

Kita biasanya membutukan *use case* yang membatalkan tujuan, misalnya pada u*se case* pemesanan kamar, dibutuhkan pula *use case* pembatalan pesanan kamar.

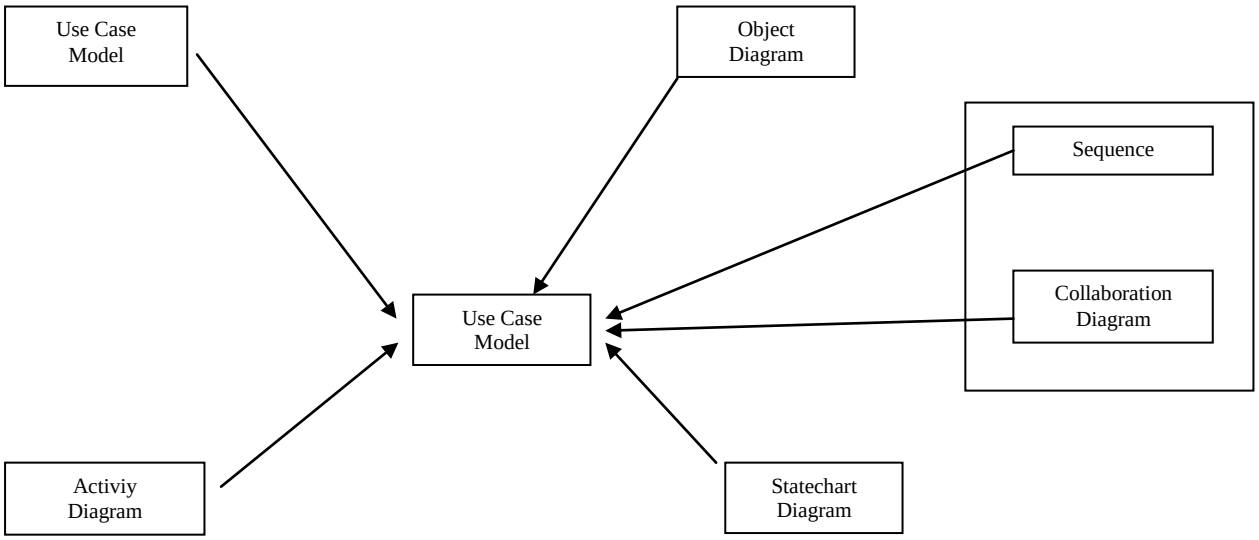
e. Batasi *Use Case* Hingga Satu Perilaku Saja.

Kadang kita cenderung membuat *use case* yang lebih dari satu tujuan aktivitas. Guna menghindari kerancuan, jagalah use case kita hanya fokus pada satu hal. Misalnya, penggunaan *use case check in* dan *check out* dalam satu *use case* menghasilkan ketidakfokusan, karena memiliki dua perilaku yang berbeda (Prabowo Pudji Widodo, Herlawati; 2011 : 22-23).

4. Diagram Kelas (*Class Diagram*)

Diagram kelas adalah inti dari proses pemodelan objek. Baik *forward engineering* maupun *reverse engineering* memanfaatkan diagram ini

*forward engineering* adalah proses perubahan model menjadi kode program sedangkan *reverse engineering* sebaliknya merubah kode program menjadi model (Probowo Pudji Widodo, Herlawati; 2011 : 37).




Gambar II.12. Hubungan Diagram Kelas Dengan Diagram *UML* lainya
Sumber : Prabowo Pudjo Widodo, Herlawati (2011 : 38)

5. Diagram Aktivitas (*Activity Diagram*)

Diagram aktivitas lebih memfokuskan diri pada eksekusi dan alur sistemdari pada bagaimana sistem dirakit. Diagram ini tidak hanya memodelkan software melainkan memodelkan bisnis juga. Diagram aktivitas menunjukan aktivitas sistem dalam kumpulan aksi-aksi. Ketika digunakan dalam pemodelan *software,* diagram aktivitas merepresentasikan pemanggilan suatu fungsi tertentu misalnya *call.* Sedangkan bila digunakan dalam pemodelan bisnis, diagram ini menggambarkan aktivitas yang dipicu oleh kejadian-kejadian diluar seperti

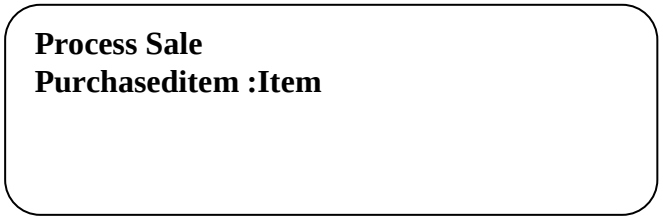
pemesanan atau kejadian-kejadian internal misalnya penggajian tiap jumat sore (Probowo Pudji Widodo, Herlawati ;2011 : 143-145).

Aktivitas merupakan kumpulan aksi-aksi. Aksi-aksi nelakukan langka sekali saja tidak boleh dipecah menjadi beberapa langkah-langkah lagi. Contoh aksinya yaitu :

a. Fungsi Matematika
b. Pemanggilan Perilaku
c. Pemrosesan Data

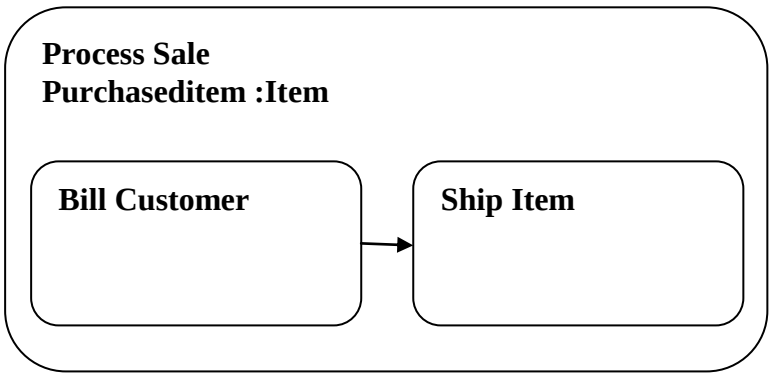
Ketika kita menggunakan diagram aktivitas untuk memodelkan perilaku suatu *classfier* dikatakan kontek dari aktivitas. Aktivitas dapat mengakses atribut dan operasi *classfier*, tiap objek yang terhubung dan parameter-parameter jika aktivitas memiliki hubungan dengan perilaku. Ketika digunakan dengan model proses bisnis, informasi itu biasanya disebut *process-relevant data*. Aktivitas diharapkan dapat digunakan ulang dalam suatu aplikasi, sedangkan aksi biasanya *specific* dan digunakan hanya untuk aktivitas tertentu.

Aktivitas digambarkan dengan persegi panjang tumpul. Namanya ditulis di kiri atas. Parameter yang terlibat dalam aktivitas ditulis dibawahnya.




Gambar II.13. Aktivitas serderhana tanpa rincian
Sumber : Prabowo Pudjo Widodo, Herlawati (2011:145)

Detail aktivitas dapat dimasukan di dalam kotak. Aksi diperlihatkan dengan symbol yang sama dengan aktivitas dan namanya diletakkan didalam persegi panjang.



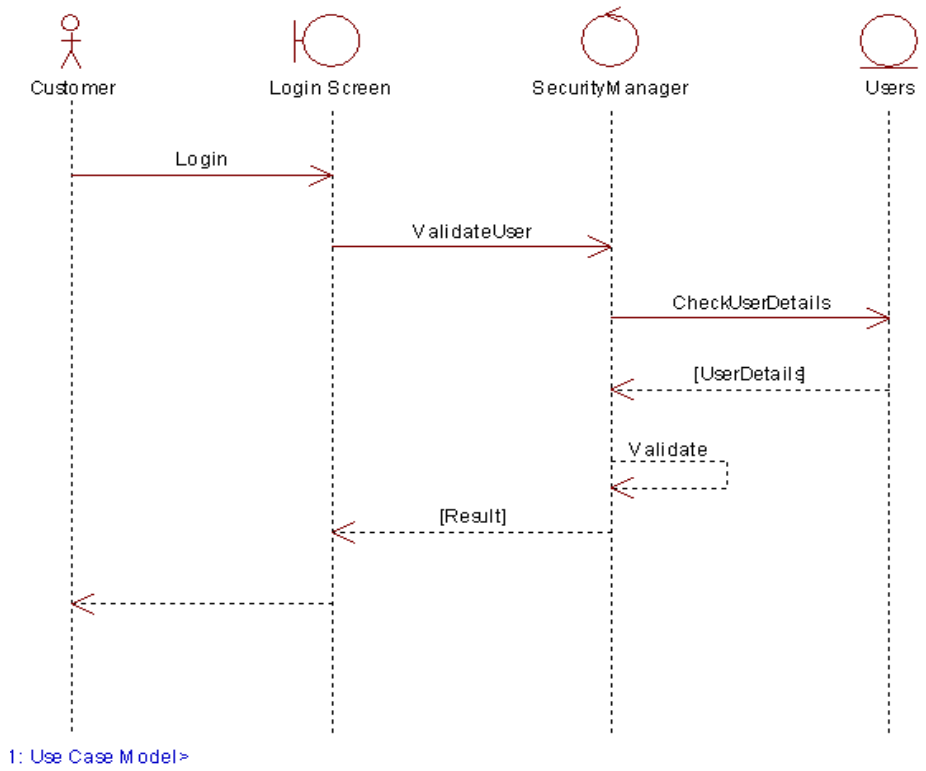

Gambar II.14. Aktivitas dengan detail rincian

Sumber : Prabowo Pudjo Widodo, Herlawati (2011:145)

**6. *Sequence Diagram***

Menurut Douglas (2004 : 174) menyebutkan ada tiga diagram primer UML dalam memodelkan scenario interaksi, yaitu diagram urutan (*sequence diagram)*, diagram waktu (*timing diagram)* dan diagram komunikasi (*communication diagram).*

Menurut Pilone (2005 : 174) menyatakan bahwa diagram yang paling banyak dipakai adalah diagram urutan. Gambar II.15. memperlihatkan contoh diagram urutan dengan notasi-notasinya yang akan dijelaskan nantinya (Prabowo Pudji Widodo, Herlawati; 2011 : 174-175)




Gambar II.15. Diagram Urutan

Sumber : Prabowo Pudjo Widodo, Herlawati (2011:175).

## II.7. Bahasa Pemograman *Microsoft Visual Studio* 2008

*Microsoft Visual Studio* 2008 merupakan kelanjutan dari *Microsoft Visual Studio* sebelumnya, yaitu *Visual Studio. Net* 2003 yang diproduksi oleh Microsoft. Pada bulan Februari 2002 *Microsoft* memproduksi teknologi. *Net Framework* versi 1.0, teknologi. *Net* ini didasarkan atas susunan berupa *Net Framework*, sehingga setiap produk baru yang terkait dengan teknologi. Net akan selalu berkembang mengikuti perkembangan. *Net Frameworknya*. Pada perkembangannya nantinya mungkin untuk membuat program dengan teknologi. *Net* memungkinkan para pengembang perangkat lunak akan dapat menggunakan lintas sistem operasi, yaitu dapat dikembangkan di sistem operasi windows juga

dapat dijalankan pada sistem operasi lain, misalkan pada sistem operasi *Linux*, seperti yang telah dilakukan pada pemograman *Java* oleh *Sun Microsystem*. Pada saat ini perusahaan-perusahaan sudah banyak meng*update* aplikasi lama yang dibuat *Microsoft Visual Basic* 6.0 ke teknologi. *Net* karena kelebihan-kelebihan yang ditawarkan, terutama memungkinkan pengembang perngkat lunak secara cepat mampu membuat program *robust*, serta berbasiskan integrasi ke internet yang dikenal dengan *XML Web Service* (Ketut Darmayuda; 2008 :1)

Untuk melihat tampilan visual studio 2008 dapat dilihat pada gambar II.16. sebagai berikut :

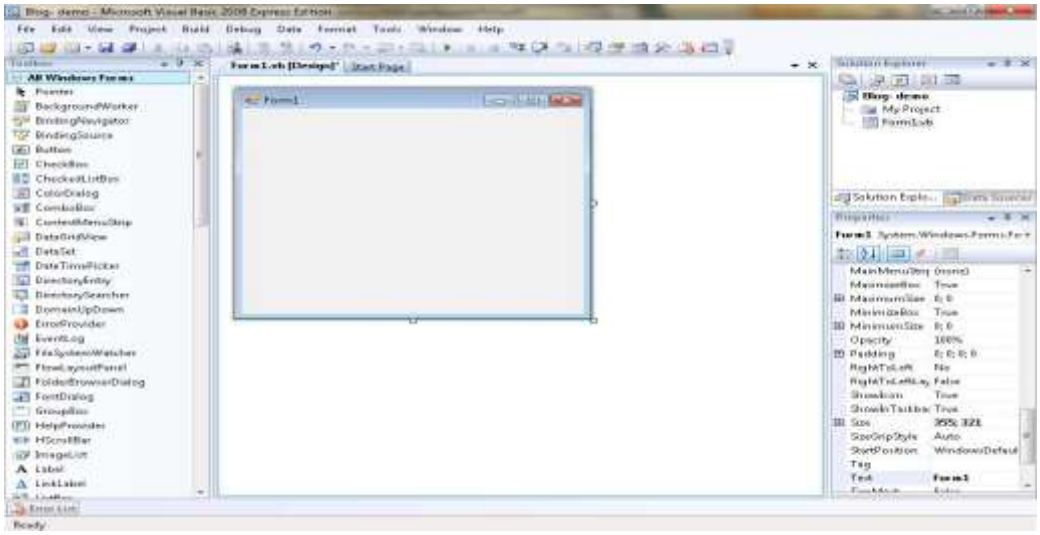

Gambar II.16. Tampilan Utama Visual Studio 2008

Sumber : Ketut Darmayuda (2008 : 12).

### II.7.1. Antar Muka Microsoft Visual Basic. Net IDE 2008.

Antarmuka atau lingkungan dari *Visual Basic. Net IDE* 2008 tidak jauh berbeda dengan *Visual Basic 6.0 IDE*, kelebihannya memiliki IDE (*Interface Development Envroitment*) yang lebih lengkap dan terorganisasi, sehingga mudah

bagi pengembang untuk mencari objek-objek atau komponen yang terdapat pada *toolbox* yang kita inginkan, untuk ditempatkan pada objek *form*, dengan *meng-klik sebuah objek* dan kemudian diletakkan diatas *form*. Berikut adalah tampilan lingkungan dari *Visual Basic. Net* 2008 dapat dilihat pada Gambar II.17. sebagai berikut :



Gambar II.17. Interface Microsoft Visual Basic. Net IDE 2008

Sumber : Ketut Darmayuda (2008 : 13)

### II.7.2. Lingkungan Kerja Pada Microsoft Visual Basic. Net 2008

1. *Tittle Bar*

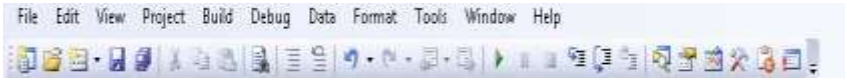
   *Tittle bar* berfungsi untuk menampilkan nama project yang aktif atau sedang dikembangkan. Adapun *tittle bar* dapat dilihat pada gambar II.18. sebagai berikut :


Gambar II.18. Tampilan *Tittle Bar Visual Studio* 2008

Sumber : Ketut Darmayuda (2008 : 14)

2. Menu Bar

Menu bar berfungsi untuk pengelolaan fasilitas yang dimiliki oleh *visual basic. Net* 2008, sedangkan *Tool Bar* berfungsi untuk melakukan perintah khusus secara cepat. Adapun menu bar dapat dilihat pada gambar II.19. sebagai berikut :

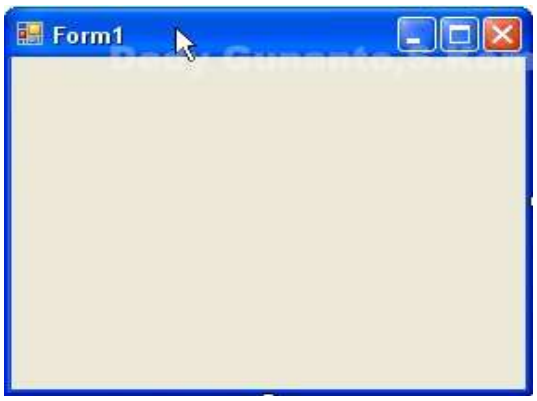



Gambar II.19. Tampilan Menu Bar Visual Studio 2008

Sumber : Ketut Darmayuda (2008 : 14)

3. *Form*

*Form* adalah objek utama yang berfungsi untuk meletakkan objek-objek yang terdapat pada toolbox yang digunakan dalam melakukan perancangan sebuah tampilan program aplikasi. Adapun *form* dapat dilihat pada gambar II.20. sebagai berikut :




Gambar II.20. Tampilan *Form Visual Studio* 2008

Sumber : Ketut Darmayuda (2008 : 14)

4. *ToolBox*

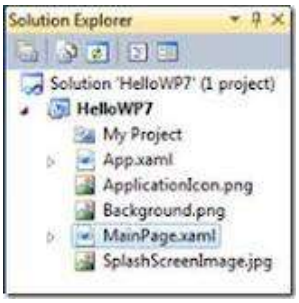
*ToolBox* berfungsi untuk menyediakan objek-objek atau komponen yang digunakan dalam merancang sebuah *form* pada program aplikasi. Adapun tampilan *toolbox* dapat dilihat pada gambar II.21. sebagai berikut :

Gambar II.21. Tampilan *ToolBox Visual Studio* 2008

Sumber : Ketut Darmayuda (2008 : 15)

5. *Solution Explorer*

*Solution Explorer* berfungsi untuk menampilkan *project* beserta *file-file* pendukung yang terdapat pada sebuah program aplikasi. Adapun tampilan *Solution Explorer* dapat dilihat pada gambar II.22. sebagai berikut :

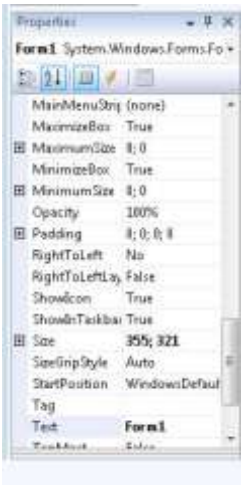



Gambar II.22. Tampilan *Solution Explorer Visual Studio* 2008

Sumber : Ketut Darmayuda (2008 : 16)

6. *Properties Windows*

*Properties windows* berfungsi untuk mengatur *properties-properties* pada objek (*setting object*) yang diletakkan pada sebuah *form*. Adapun tampilan *properties windows* dapat dilihat pada gambar II.23. sebagai berikut :




Gambar II.23. Tampilan *Properties Windows Visual Studio* 2008
Sumber : Ketut Darmayuda (2008 : 16)

## II.8. *Microsoft SQL Server 2008*

*SQL Server* adalah salah satu produk *Relational Database Management Sistem* (RDBMS) populer saat ini. Fungsi utamanya adalah sebagai *database server* yang mengatur semua proses penyimpanan data dan transaksi suatu aplikasi. *SQL Server Versi 2008* memiliki *feature-feature* lengkap untuk membangun aplikasi mulai skala kecil sampai dengan tingkat *enterprise*. Untuk lebih jelasnya dapat dilihat pada Gambar II.24 (Andi ; 2008 : 1).


Gambar II.24. *SQL Server 2008*

Sumber : Andi (2008 : 20)

### II.8.1. Tingkatan Data *SQL Server 2008*

Ada 3 tingkatan data *SQL Server* 2008 adalah sebagai berikut :

1. Level fisik *(Physcal Level)*

   Merupakan level yang paling rendah karena menggambarkan bagaimana data disimpan pada kondisi yang sebenarnya pada server.

2. Level Konseptual *(Conseptual Level)*

   Merupakan level yang menggambarkan data yang disimpan pada database dan dijelaskan secara keseluruhan antara data tersebut.

3. Level Pandangan *(View Level)*

   Merupakan level yang menggambarkan sebagian dari seluruh database sesuai dengan kebutuhan user (Andi ; 2008 :3-4).

### II.8.2. *SQL Server Configuration Manager*

Ada 3 tingkatan dalam *Sql Server Configuration Manager* adalah sebagai berikut :

1. *Restart Service*

   Kita bisa melakukan restart pada setiap *SQL Server Service* tersebut jika diperlukan

2. *Start Mode*

   Setiap *SQL Service* telah dikonfigurasikan secara *default* menjalankan *SQL Server Service* saat *server restart* atau baru dinyalakan.

3. *Network Configuration*

   Pada *SQL Server Network Configuration* terdapat beberapa protocol yang bisa digunakan untuk melakukan antar muka dengan *SQL Server* 2008 yaitu :

   a. *Shared Memory*

      Dengan *shared memory*, koneksi database bisa ditempatkan pada server lokal tanpa menggunakan network.

   b. *Named Pipes*

      Dengan *Named Pipes*, koneksi bisa dilakukan dengan *high speed* LAN.

   c. *TCP/IP*

      Dengan *TCP/IP*, koneksi *database* bisa dilakukan dengan koneksi *network* yang menggunakan *TCP/IP* (Andi; 2008 : 16-18).