BAB 2
TINJAUAN PUSTAKA

2.1 Profil Instansi

Profil dari perusahaan yang akan dibahas adalah mengenai profil dari Lembaga Ilmu Pengetahuan Indonesia (LIPI). Profil perusahaan meliputi sejarah perkembangan, visi dan misi, logo perusahaan, badan hukum, struktur organisasi serta tugas dan fungsi dari Lembaga Ilmu Pengetahuan Indonesia (LIPI).

2.1.1 Sejarah Lembaga Ilmu Penelitian Indonesia (LIPI)

Kegiatan ilmiah di Indonesia dimulai pada abad ke-16 oleh Jacob Bontius, yang mempelajari flora Indonesia dan Rompius dengan karyanya yang terkenal berjudul Herbarium Amboinese. Pada akhir abad ke-18 dibentuk Bataviaasch Genotschap van Wetenschappen. Dalam tahun 1817, C.G.L. Reinwardt mendirikan Kebun Raya Indonesia (S'land Plantentuin) di Bogor. Pada tahun 1928 Pemerintah Hindia Belanda membentuk Natuurwetenschappelijk Raad voor Nederlandsch Indie. Kemudian tahun 1948 diubah menjadi Organisatie voor Natuurwetenschappelijk onderzoek (Organisasi untuk Penyelidikan dalam Ilmu Pengetahuan Alam, yang dikenal dengan OPIPA). Badan ini menjalankan tugasnya hingga tahun 1956.

Pada tahun 1956, melalui UU no. 6 tahun 1956 pemerintah Indonesia membentuk Majelis Ilmu Pengetahuan Indonesia (MIPI) dengan tugas pokok :

1. Membimbing perkembangan ilmu pengetahuan dan teknologi.
2. Memberi pertimbangan kepada pemerintah dalam hal kebijaksanaan ilmu pengetahuan.

Kemudian pada tahun 1962 pemerintah membentuk Departemen Urusan Riset Nasional (DURENAS) dan menempatkan MIPI didalamnya dengan tugas tambahan : membangun dan mengasuh beberapa Lembaga Riset Nasional. Dan tahun 1966 pemerintah merubah status DURENAS menjadi Lembaga Riset Nasional (LEMRENAS).

Pada bulan Agustus 1967 pemerintah membubarkan LEMRENAS dan MIPI dengan SK Presiden RI no. 128 tahun 1967, kemudian berdasarkan Keputusan MPRS no. 18/B/1967 pemerintah membentuk Lembaga Ilmu Pengetahuan Indonesia (LIPI) dan menampung seluruh tugas LEMRENAS dan MIPI, dengan tugas pokok sebagai berikut :

1. Membimbing perkembangan ilmu pengetahuan dan teknologi yang berakar di Indonesia agar dapat dimanfaatkan bagi kesejahteraan rakyat Indonesia pada khususnya dan umat manusia pada umumnya.
2. Mencari kebenaran ilmiah dimana kebebasan ilmiah, kebebasan penelitian serta kebebasan mimbar diakui dan dijamin, sepanjang tidak bertentangan dengan Pancasila dan UUD 1945.
3. Mempersiapkan pembentukan Akademi Ilmu Pengetahuan Indonesia (sejak 1991 tugas pokok ini selanjutnya ditangani oleh Menteri Negara Riset dan Teknologi dengan Keppres no. 179 tahun 1991).

Sejalan dengan perkembangan kemampuan nasional dalam bidang ilmu pengetahuan dan teknologi, organisasi lembaga-lembaga ilmiah di Indonesia telah pula mengalami pertumbuhan dan perkembangan. Oleh sebab itu dipandang perlu untuk mengadakan peninjauan dan penyesuaian tugas pokok dan fungsi serta susunan organisasi LIPI sesuai dengan tahap dan arah perkembangan ilmu pengetahuan dan teknologi, maka Keppres no. 128 tahun 1967, tanggal 23 Agustus 1967 diubah dengan Keppres no. 43 tahun 1985, dan dalam rangka penyempurnaan lebih lanjut, tanggal 13 Januari 1986 ditetapkan Keppres no. 1 tahun 1986 tentang Lembaga Ilmu Pengetahuan Indonesia, dan terakhir dengan Keppres no. 103 tahun 2001.

2.1.2 Visi dan Misi

2.1.2.1 Visi

Menjadi lembaga ilmu pengetahuan berkelas dunia yang mendorong terwujudnya kehidupan bangsa yang adil, cerdas, kreatif, integratif dan dinamis yang didukung oleh ilmu pengetahuan dan teknologi yang humanis.

2.1.2.2 Misi

1. Menciptakan great science (ilmu pengetahuan berdampak penting) dan invensi yang dapat mendorong inovasi dalam rangka meningkatkan daya saing perekonomian nasional.
2. Mendorong peningkatan pemanfaatan pengetahuan dalam proses penciptaan *good governance* dalam rangka memantapkan NKRI.
3. Turut serta dalam proses pencerahan kehidupan masyarakat dan kebudayaan berdasarkan prinsip-prinsip ilmu pengetahuan dan kaidah etika keilmuan.
4. Memperkuat peran Indonesia (yang didukung ilmu pengetahuan) dalam pergaulan internasional.
5. Memperkuat infrastruktur kelembagaan (penguatan manajemen dan sistem).

2.1.3 Logo Lembaga Ilmu Pengetahuan Indonesia (LIPI)

Logo Lembaga Ilmu Pengetahuan Indonesia (LIPI) berdasarkan SK Ketua LIPI Nomor 37/KEP/C.3/1975 (Gambar 2.1) dengan penjelasan sebagai berikut:

a. Logo dan Penjelasan Logo

   Logo LIPI adalah huruf dan atau lambang yang mengandung suatu makna sebagai lambang lembaga/identitas yang melekat dalam mengikat batin dan kesatuan jiwa seluruh pegawai LIPI.

b. Bentuk Logo dan Jenis Warna

   Logo LIPI adalah berbentuk bulat dan merupakan satu kesatuan dari lambang dan huruf LIPI dibawahnya.

Jenis warna adalah “TURQUISE BLUE” dengan spesifikasi warna campuran sebagai berikut :



Gambar 2.1 Logo Lembaga Ilmu Pengetahuan Indonesia (LIPI)

Logo LIPI terdiri dari gambar dan tulisan LIPI sebagai satu kesatuan yang tidak dipisahkan. Penempatan Logo LIPI ditempatkan pada tempat yang layak dan baik bagi pandangan mata.

Penggunaan Logo LIPI dilakukan pada Kantor LIPI, Ruang Rapat, Kop Surat Dinas, Cap Instansi, Amplop Dinas, Pakaian Resmi, Jaket, Kartu Nama, dll.

Penanggung jawab penggunaan Logo LIPI antara lain adalah sebagai berikut :

1. Inspektorat LIPI
2. Kepala Satuan Kerja
3. Biro Umum dan Perlengkapan LIPI

Dan barang siapa yang tidak mengindahkan tata cara dalam penggunaan Logo LIPI, akan dikenakan sanksi menurut peraturan yang berlaku.

2.1.4 Tugas, Fungsi dan Kewenangan

Adapun tugas, fungsi dan kewenangan dari Lembaga Ilmu Penelitian Indonesia adalah :

1. Tugas

   Melaksanakan tugas pemerintahan di bidang ilmu pengetahuan sesuai dengan ketentuan peraturan perundang-undangan yang berlaku.

2. Fungsi
   a. Pengkajian dan penyusunan kebijakan nasional di bidang penelitian ilmu pengetahuan.
   b. Penyelenggaraan riset keilmuan yang bersifat mendasar.
   c. Penyelenggaraan riset inter dan multi disiplin terfokus.

d. Pemantauan, evaluasi kemajuan, dan penelaahan kecenderungan ilmu pengetahuan dan teknologi.
e. Koordinasi kegiatan fungsional dalam pelaksanaan tugas LIPI.
f. Pelancaran dan pembinaan terhadap kegiatan instansi pemerintah di bidang ilmu pengetahuan.
g. Penyelenggaraan pembinaan dan pelayanan administrasi umum di bidang perencanaan umum, ketatausahaan, organisasi dan tata laksana, kepegawaian, keuangan, kearsipan, persandian, perlengkapan dan rumah tangga.

3. Kewenangan

Untuk menjalankan fungsinya, LIPI mempunyai kewenangan :

a. Penyusunan rencana nasional secara makro di bidangnya.
b. Perumusan kebijakan di bidangnya untuk mendukung pembangunan secara makro.
c. Penetapan sistem informasi di bidangnya.
d. Kewenangan lain yang melekat dan telah dilaksanakan sesuai dengan ketentuan peraturan perundang-undangan yang berlaku, yaitu :
    1. Perumusan dan pelaksanaan kebijakan tertentu di bidang penelitian ilmu pengetahuan.
    2. Penetapan pedoman dan penyelenggaraan riset ilmu pengetahuan dasar.
    3. Penetapan pedoman etika ilmiah, kedudukan dan kriteria kelembagaan ilmiah.
    4. Pemberian ijin Peneliti Asing.
    5. Pemegang kewenangan ilmiah dalam keanekaragaman hayati.

2.1.5 Pusat Penelitian Informatika LIPI

Pusat Penelitian Informatika didirikan dengan surat Keputusan Kepala LIPI No 1151/M/2001 tanggal 5 Juni 2001. Semula bernama Pusat Penelitian dan Pengembangan Informatika dan Ilmu Pengetahuan Komputer didirikan pada tanggal 13 Januari 1986.

Berdirinya Pusat Penelitian Informatika tidak dapat dipisahkan dengan sejarah terbentuknya Lembaga Elektroteknika Nasional (LEN). Lembaga Elektroteknika Nasional sebagai lembaga penelitian dalam bidang elektronika diresmikan pada tanggal 10 Juni 1965. Bermula di kampus ITB dengan ruang seluas 30 meter persegi yang merupakan pinjaman dari ITB. Pada tahun 1967, berdasarkan Keputusan Presiden nomor 128 tahun 1967, Lembaga Elektroteknika Nasional (LEN) bernaung dibawah koordinasi Lembaga Ilmu Pengetahuan Indonesia. Pada tahun 1969, Prusahaan Negara Pertambangan Minyak dan Gas Bumi Nasional (Pertamina), memberikan pinjaman gedung seluas 1000 meter persegi di Jl. Sawunggaling No. 14 Bandung, sehingga ruangan dikomplek ITB ditinggalkan. Pada awal tahun 1978, komplek gedung perkantoran dan laboratorium LEN-LIPI sudah tersebar di 9 lokasi di kota Bandung, dengan luas lantai sekitar 2500 meter persegi.

Tahun 1983 LEN-LIPI pindah ke Jl. Soekarno-Hatta No.407 250A Bandung, serta gedung tersebut diresmikan pemakaiannya oleh Presiden Republik Indonesia, Bapak Soeharto pada tanggal 31 Mei 1983. Dengan dilakukannya reorganisasi di Lembaga Ilmu Pengetahuan Indonesia (LIPI) berdasarkan Keppres No. l tahun 1986 tanggal 13 Januari 1986, Lembaga Elektroteknika Nasional berkembang menjadi tiga Puslitbang dan satu Unit Pelaksana Teknis, yaitu:

1. Pusat Penelitian dan Pengembangan Telekomunikasi, Elektronika Strategis, Komponen dan Material (Puslitbang Telkoma-LIPI).
2. Pusat Penelitian dan Pengembangan Informatika dan Ilmu Pengetahuan Komputer (Puslitbang Inkom LIPI).
3. Pusat Penelitian dan Pengembangan Tenaga Listrik dan Mekatronika (Puslitbang Telimek-LIPI)
4. UPT Pusat Laboratorium Enjiniring Nasional (UPT Pusat LEN LIPI).

Dalam struktur organisasi LIPI, keempat Puslibang tersebut diatas berada dibawah koordinasi Deputi Bidang Ilmu Pengetahuan Teknik -LIPI.

Pada tanggal 8 Maret 1990 seluruh fasilitas UPT LEN-LIPI diserahkan ke BPIS. Kantor Puslitbang Inkom LIPI pindah ke Jl. Ranggamalela No. 11 Bandung, sedangkan laboratoriumnya menempati salah satu bagian gedung di

kompleks LIPI Jl. Cisitu 21/154D Bandung. Pada tahun 2001 dalam Reorganisasi baru Lembaga Ilmu Pengetahuan Indonesia, Pusat Penelitian dan Pengembangan Informatika dan Ilmu Pengetahuan Komputer berubah nama dan struktur organisasinya menjadi Pusat Penelitian Informatika LIPI.

2.1.5.1 Visi dan Misi

2.1.5.1.1 Visi

Pusat Penelitian Informatika mengabdi untuk kepentingan masyarakat dan menyelenggarakan kegiatan di bidang Informatika dan Ilmu Pengetahuan Komputer untuk pembangunan berkelanjutan.

Maknanya bahwa berbagai kegiatan penelitian yang dilakukan di bidang Informatika adalah untuk kepentingan masayarakat, antara lain untuk mencerdaskan kehidupan bangsa dengan menggunakan kemampuan teknologi informasi yang baru. Teknologi ini digunakan untuk pengumpulan, pengolahan, penyimpanan dan penyebarluasan data atau informasi kepada masyarakat dengan cepat, tepat waktu dan akurat. Sehingga akan terjadi percepatan peningkatan kemampuan masyarakat dalam pembangunan yang berkelanjutan.

2.1.5.1.2 Misi

Pusat Penelitian Informatika mengabdi untuk kepentingan masyarakat dan menyelenggarakan kegiatan di bidang Informatika dan Ilmu Pengetahuan Komputer untuk pembangunan berkelanjutan.

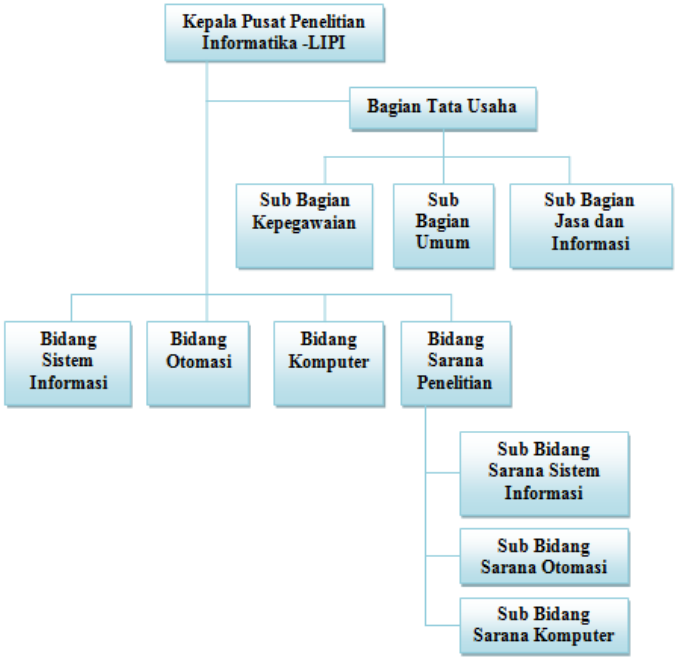
Maknanya bahwa berbagai kegiatan penelitian yang dilakukan di bidang Informatika adalah untuk kepentingan masyarakat, antara lain untuk mencerdaskan kehidupan bangsa dengan menggunakan kemampuan teknologi informasi yang baru. Teknologi ini digunakan untuk pengumpulan, pengolahan, penyimpanan dan penyebarluasan data atau informasi kepada masyarakat dengan cepat, tepat waktu dan akurat. Sehingga akan terjadi percepatan peningkatan kemampuan masyarakat dalam pembangunan yang berkelanjutan.

2.1.5.2 Struktur Organisasi Lembaga Ilmu Pengetahuan Indonesia (LIPI)

Struktur Organisasi Pusat Penelitian Informatika-LIPI, berdasarkan SK Kepala LIPI Nomor 1151/M/2001 tanggal 5 Juni 2001 tentang Organisasi dan Tata Kerja sebagaimana telah diubah dengan Keputusan Kepala LIPI Nomor 3212/M/2004 dimana struktur organisasi terdiri dari :

1. Kepala Pusat Penelitian Informatika LIPI
2. Bagian Tata Usaha, yang terdiri dari.
   a. Sub Bagian Kepegawaian
   b. Sub Bagian Umum
   c. Sub Bagian Jasa dan Informasi
3. Bidang Sistem Informasi
4. Bidang Otomasi
5. Bidang Komputer
6. Bidang Sarana Penelitian, yang terdiri dari.
   a. Sub Bidang Sarana Sistem Informasi
   b. Sub Bidang Sarana Otomasi
   c. Sub Bidang Sarana Komputer

Struktur organisasi dari Lembaga Ilmu Pengetahuan Indonesia (LIPI) dapat dilihat pada Gambar 2.2 berikut ini :



Gambar 2.2 Struktur Organisasi Lembaga Ilmu Pengetahuan Indonesia

2.2 Landasan Teori

2.2.1 Pengertian Data dan Informasi

Menurut Steven Alter, data merupakan fakta,gambar atau suara yang mungkin atau tidak berhubungan atau berguna bagi tugas tertentu.

Menurut McLeod, data terdiri dari fakta-fakta dan angka yang secara relatif tidak berarti bagi pemakai. Sedangkan informasi adalah data yang sudah diproses atau data yang memiliki arti.

Disini kita dapat melihat bahwa data merupakan suatu bentuk keterangan-keterangan yang belum diolah atau dimanipulasi sehingga belum begitu berarti bagi sebagian pemakai.

Menurut Raymond Mc.leod, informasi adalah data yang telah diolah menjadi bentuk yang memiliki arti bagi si penerima dan bermanfaat bagi pengambilan keputusan saat ini atau mendatang .

Menurut Jogiyanto HM., (1999: 692), “Informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan

lebih berarti bagi penerimanya yang menggambarkan suatu kejadian – kejadian (*event*) yang nyata (*fact*) yang digunakan untuk pengambilan keputusan".

Menurut George H. Bodnar, (2000: 1), "Informasi adalah data yang diolah sehingga dapat dijadikan dasar untuk mengambil keputusan yang tepat".

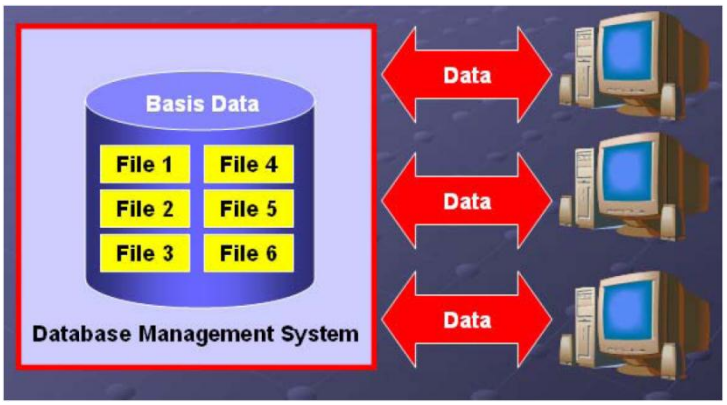
2.2.2 Basis Data dan Sistem Manajemen Basis Data (*Database and Database Management System*)

Menurut James A. O'Brien, *database* adalah suatu koleksi terintegrasi dimana secara logika berhubungan dengan *record* dari *file*.

Menurut Fatansyah, *database* adalah kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan [2].

Jadi *database* adalah tempat penyimpanan data yang saling berhubungan secara logika, sehingga bisa digunakan untuk mendapatkan suatu informasi yang diperlukan oleh suatu organisasi atau perusahaan. Sedangkan data yang diperoleh suatu organisasi atau perusahaan umumnya didapat dari kegiatan operasional sehari-hari atau hasil dari transaksi.

Sistem Manajemen Basis Data adalah program yang digunakan pengguna untuk membuat dan memelihara *database*. Sistem Manajemen Basis Data memfasilitasi untuk mendefinisikan, mengkonstruksi, dan memanipulasi *database* untuk berbagai aplikasi. Pendefinisian *database* meliputi spesifikasi tipe data, struktur, dan *constraint* untuk data yang disimpan dalam *database*. Pengkonstruksian *database* adalah proses penyimpanan data itu sendiri pada media penyimpanan. Pemanipulasian *database* meliputi fungsi memanggil *query database* untuk mendapatkan data yang spesifik, *update database*, dan meng-*generate* laporan dari data tersebut (Elmasri, 2000).



Gambar 2.3 ***Database Management System***

Keuntungan yang diperoleh menggunakan Sistem Manajemen Basis Data meliputi mengontrol redudansi, membatasi akses yang tidak berwenang, menyediakan penyimpanan yang persisten, menghasilkan *interface* (antar muka) banyak pengguna, menjaga integritas constraint, menyediakan *backup* dan *recovery*.

Dalam Sistem Manajemen Basis Data menyediakan perintah yang digunakan untuk mengelola dan mengorganisasikan data, yakni *Data Definition Language (DDL*) dan *Data Manipulation Language* (*DML*). *Data Definition Language* adalah bahasa untuk medefinisikan skema atau dan *database* fisik ke Sistem Manajemen Basis Data. *Data Manipulation Language* adalah bahasa untuk memanipulasi data yaitu pengambilan informasi yang telah disimpan, penyisipan informasi baru, penghapusan informasi, modifikasi informasi yang disimpan dalam *database*. Selanjutnya, *query* adalah statemen yang ditulis untuk mengambil informasi. Bagian dari DML yang menangani pengambilan informasi ini disebut bahasa *query*. Dari perkembangan model *database* maka muncul apa yang disebut dengan *data warehouse* [6].

2.2.3 *Data Mart*

Data mart adalah suatu bagian pada *data warehouse* yang mendukung pembuatan laporan dan analisis data pada suatu unit, bagian operasi pada suatu perusahaan. *Data mart* berisi informasi yang relevan bagi user yang ingin mengambil keputusan, sebuah perusahaan yang telah menginvestasikan baik uang dan waktunya dalam sebuah bisnis operasional yang telah mempunyai dan

menyimpan sebuah data dari pelanggan, penjualan dan inventori. Data tersebut merupakan asset dari suatu perusahaan yang perlu di manage untuk keperluan kompetitif perusahaan. Banyak perusahaan yang telah menerapkan DSS (*Decision Support system*) yang telah ada di suatu data mart, sebuah data mart terintegrasi secara operasional dan sejarah data untuk aplikasi pengambilan keputusan.

Ada empat tugas yang bisa dilakukan dengan adanya data mart menurut Williams, keempat tugas tersebut yaitu [5]:

1. Pembuatan laporan

   Pembuatan laporan merupakan salah satu kegunaan data warehouse yang paling umum dilakukan. Dengan menggunakan query sederhana didapatkan laporan perhari,perbulan, pertahun atau jangka waktu kapanpun yang diinginkan.

2. *On-Line Analytical Processing (OLAP)*

   Dengan adanya data warehouse,semua informasi baik detail maupun hasil summary yang dibutuhkan dalam proses analisa mudah didapat. *OLAP* mendayagunakan konsep data multi dimensi dan memungkinkan para pemakai menganalisa data sampai mendetail, tanpa mengetikkan satupun perintah *SQL*. Hal ini dimungkinkan karena pada konsep multi dimensi, maka data yang berupa fakta yang sama bisa dilihat dengan menggunakan fungsi yang berbeda. Fasilitas lain yang ada pada sofware OLAP adalah fasilitas *rool-up* dan *drill-down*. *Drill-down* adalah kemampuan untuk melihat detail dari suatu informasi dan roll-up adalah kebalikannya.

3. *Data mining*

   Data mining merupakan proses untuk menggali (*mining*) pengetahuan dan informasi baru dari data yang berjumlah banyak pada *data mart*, dengan menggunakan kecerdasan buatan (*Artificial Intelegence*), statistik dan matematika. *Data mining* merupakan teknologi yang diharapkan dapat menjembatani komunikasi antara data dan pemakainya.

   Beberapa solusi yang diberikan data mining antara lain :

   a. Menebak target pasar

Data mining dapat mengelompokkan (*clustering*) model-model pembeli dan melakukan klasifikasi terhadap setiap pembeli dan melakukan klasifikasi terhadap setiap pemebeli sesuai dengan karakteristik yang diinginkan.

b. Melihat pola beli dari waktu ke waktu

Data mining dapat digunakan untuk melihat pola beli dari waktu ke waktu.

c. *Cross-market analysis*

Data mining dapat dimanfaatkan untuk melihat hubungan antara satu produk dengan produk lainnya.

d. Profil pelanggan

Data mining bisa membantu pengguna untuk melihat profil pembeli sehingga dapat diketahui kelompok pembeli tertentu cenderung kepada suatu produk apa saja.

e. Informasi *summary*

Data mining dapat membuat laporan summary yang bersifat multi dimensi dan dilengkapi dengan informasi statistik lainnya.

4. Proses informasi *executive*

Data mart dapat membuat ringkasan informasi yang penting dengan tujuan membuat keputusan bisnis, tanpa harus menjelajahi keseluruhan data. Dengan menggunakan *data mart* segala laporan telah diringkas dan dapat pula mengetahui segala rinciannya secara lengkap, sehingga mempermudah proses pengambilan keputusan. Informasi dan data pada laporan *data mart* menjadi target informative bagi *user*.

2.2.3.1 Karakteristik *Data Mart*

Seperti yang telah dikemukakan terlebih dahulu pada sub bagian pengertian *data mart*, karakteristik yang harus dimiliki dalam sebuah *data mart* antara lain adalah *Subject Oriented, Integrated, Time Variant*, dan *Non Volatile*. Berikut ini dijelaskan lebih lanjut mengenai karakteristik *data mart* [1] :

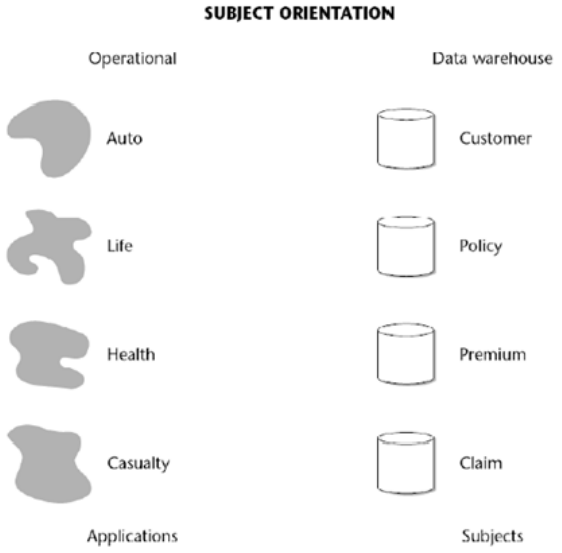
1. *Subject Oriented*

   Data mart berorientasi subjek artinya *data mart* dirancang untuk menganalisis data berdasarkan subjek-subjek tertentu dalam organisasi, bukan pada proses atau fungsi aplikasi tertentu. Jadi dengan kata lain, data yang disimpan adalah berorientasi kepada subjek bukan terhadap proses. Secara garis besar perbedaan antara data operasional dan data warehouse yaitu :

Tabel 2.1 Perbedaan Data Operasional dan *Data Mart*

Data Operasional	Data Mart
Dirancang berorientasi hanya pada aplikasi dan fungsi tertentu	Dirancang berdasar pada subjek-subjek tertentu (utama)
Fokusnya pada desain *database* dan proses	Fokusnya pada pemodelan data dan desain data
Berisi rincian atau detail data	Berisi data-data *history* yang akan dipakai dalam proses analisis
Relasi antar table berdasar aturan terkini (selalu mengikuti *rule* terbaru)	Banyak aturan bisnis dapat tersaji antara tabel-tabel

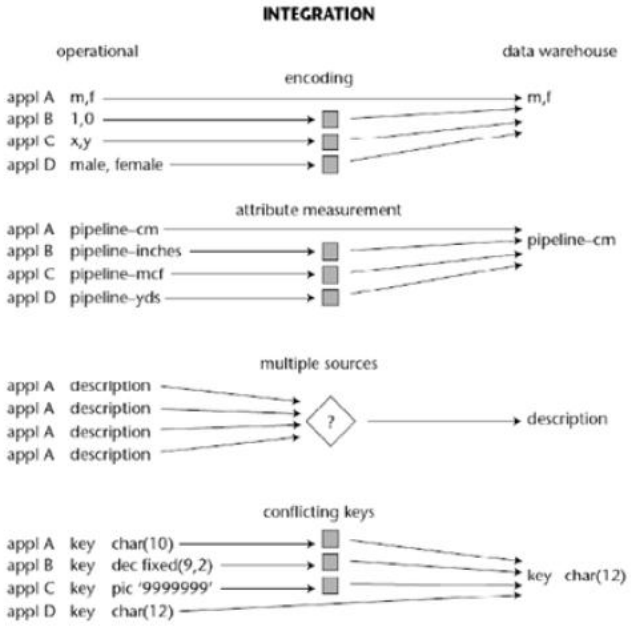
Karakteristik *subject orientation* dapat dilihat pada gambar berikut :



Gambar 2.4 ***Subject Orientation Data*** **[1]**

2. *Integrated*

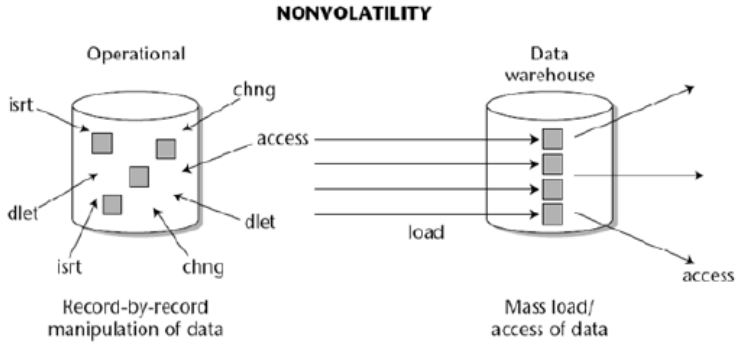
   Sumber data berasal dari sistem – sistem aplikasi perusahaan yang berbeda – beda. Sumber data sering tidak konsisten misalnya perbedaan format data. Integrasi sumber data harus dibuat konsisten dalam menampilkan data agar dapat menyatukan pandangan user terhadap data.



Gambar 2.5 ***Integration*** **[1]**

3. *Non Volatile*

Karakteristik ketiga dari *data mart* adalah *non volatile*, yaitu data pada *data mart* cenderung statis dan berupa *data historical* yang digunakan untuk analisis. Pada gambar 2.3 ini, menjelaskan bahwa "Data operasional pada tiap – tiap *record* yang ada, umum diakses dan manipulasi, perubahan maupun pembaruan data merupakan sebuah hal yang wajar dilakukan pada sistem ini. Namun, *data mart* juga dapat diakses, namun tidak terlalu sering, dan biasanya diambil dalam jumlah besar. Akan tetapi data yang ada tidak diperkenankan untuk diperbaharui maupun diubah, apabila terjadi perubahan atau pembaruan, maka data tersebut akan masuk sebagai data baru. Oleh karena itu, data yang tersimpan pada sebuah data mart merupakan data yang bersifat historis".



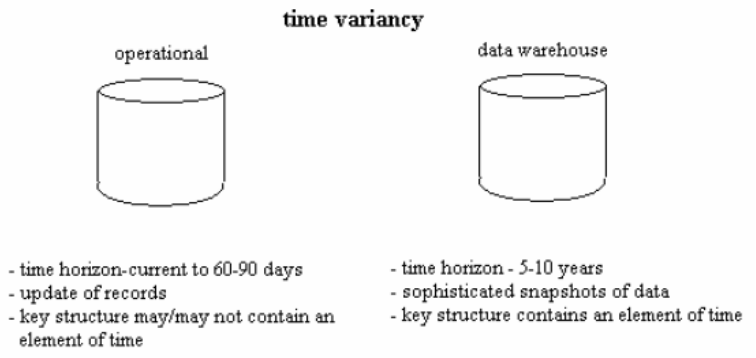
Gambar 2.6 *Non Volatile* [1]

4. *Time Variant*

Data mart dibandingkan dengan data operasional, memiliki perbedaan cakupan pada aspek *time variant*. *Time variant* yang membedakan antara data pada *data mart* dengan data operasional adalah data yang disimpan pada sistem operasional merupakan keadaan terkini pada perusahaan. Data transaksi lampau juga ada yang disimpan, namun pada intinya sistem operasi dirancang memang untuk mendukung operasi harian dari perusahaan sehingga data yang tersimpan harus mencerminkan keadaan terkini perusahaan. Namun pada *data mart*, data yang digunakan merupakan data untuk analisis dan pengambilan keputusan. Dari tujuan inilah data yang dimiliki oleh sebuah *data mart*

haruslah berupa data historis, bukan sekedar keadaan terkini dari perusahaan. Data disimpan sebagai gambaran keadaan perusahaan di masa lampau hingga saat ini. *Time variant* memiliki keuntungan sebagai berikut [3] :

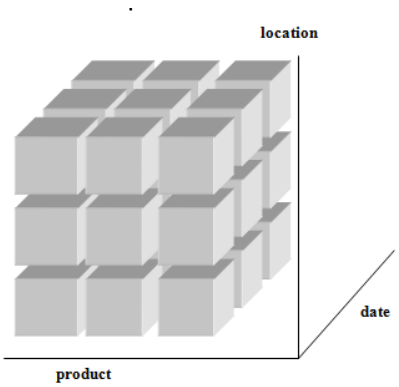
- Memungkinkan untuk menganalisis hal yang terjadi di masa lampau
- Menghubungkan informasi yang ada ke masa kini
- Memungkinkan adanya perkiraan atau ramalan mengenai kondisi masa yang akan datang.



Gambar 2.7 *Time Variant* [1]

2.2.3.2 Pemodelan Data Dimensional

Pemodelan data dimensional adalah representasi data dengan kubus multidimensional agar lebih mudah dibaca. Terdapat 2 macam aspek dalam pemodelan ini, yaitu ukuran (*measures*) dan dimensi (*dimension*). Ukuran adalah besaran data, sedangkan dimensi adalah konteks data atau parameter bisnis. Parameter ini dapat dilihat dari karakteristik seperti *who, what, when, where* dan *how* dari subjek data. Ukuran (*measures*) disimpan dalam tabel fakta (*fact table*) sedangkan dimensi disimpan dalam tabel dimensi (*dimension table*). Sebagai contoh misalnya untuk basis data mengenai total penjualan per bulan, maka pengukuran dapat dilakukan berdasarkan dimensi lokasi, waktu dan produk yang dijual. Representasi dalam bentuk kubus multidimensional untuk contoh ini dapat dilihat pada gambar berikut :



Gambar 2.8 Kubus Model Dimensional

Berikut penjelasan dari *fact table, dimension table dan measures* :

a. *Fact table*

Tabel fakta adalah pusat dari *table star join* dimana data dengan banyak kepentingan tersimpan [1]. Setiap *fact* biasanya merepresentasikan sebuah bisnis *item*, suatu transaksi bisnis, atau sebuah kejadian yang dapat digunakan dalam analisis bisnis atau proses bisnis.

b. *Dimension table*

Dimension table atau tabel dimensi adalah tempat dimana data tambahan yang berhubungan dengan tabel fakta ditempatkan pada sebuah tabel multidimensional [1]. Dalam dimensional model, semua data menunjukan *fact table* yang diasosiasikan dengan satu dan hanya satu member dari setiap *multiple dimensions.* Jadi dimensi menunjukan latar belakang kontekstual dari *fact.* Banyak proses analisis yang digunakan untuk menghitung dampak dari dimensi pada *fact.*

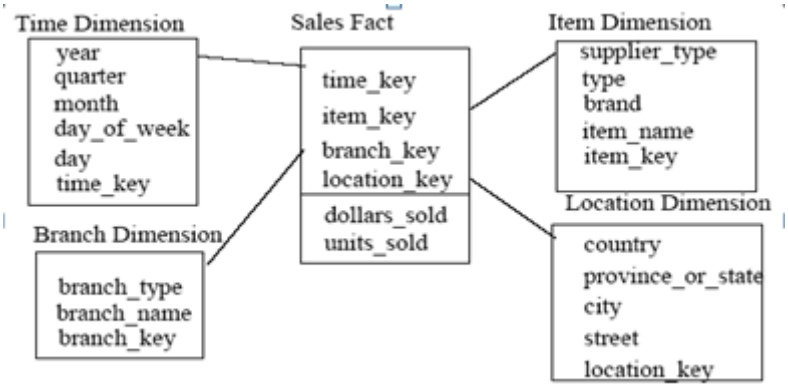
c. *Measures*

Suatu *measures* (ukuran) adalah suatu besaran (angka numerik) atribut dari sebuah *fact*, yang menunjukan *performance* atau *behavior* (tingkah laku) dari bisnis secara relatif pada suatu dimensi. Angka atau nomor yang ditunjukan disebut dengan *variable*. Sebagai contoh ukuran dari penjualan dalam bentuk uang, besarnya penjualan, jumlah pengadaan, biaya pengadaan, banyaknya transaksi dan lainnya. Suatu ukuran dijelaskan dengan kombinasi dari member dari suatu dimensi dan diletakkan dalam *fact.* Dalam *dimensional*

modeling, ada beberapa pendekatan yang digunakan untuk membuat *data mart*, yaitu [10]:

1. Skema bintang (*star schema*)

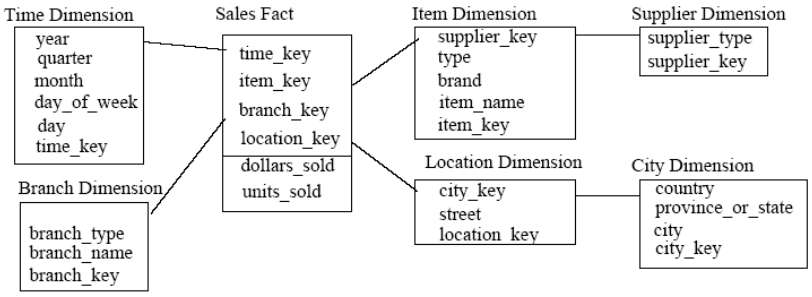
   Skema ini mengikuti bentuk bintang, dimana terdapat satu tabel fakta (*fact table*) di pusat bintang dengan beberapa tabel dimensi (*dimensional tables*) yang mengelilinginya. Semua tabel dimensi berhubungan dengan ke tabel fakta. Tabel fakta memiliki beberapa *key* yang merupakan kunci indeks individual dalam tabel dimensi. Berikut model skema bintang :



Gambar 2.9 Model Skema Bintang (*Star Schema*) [11]

2. Skema bola salju (*snowflake Schema*)

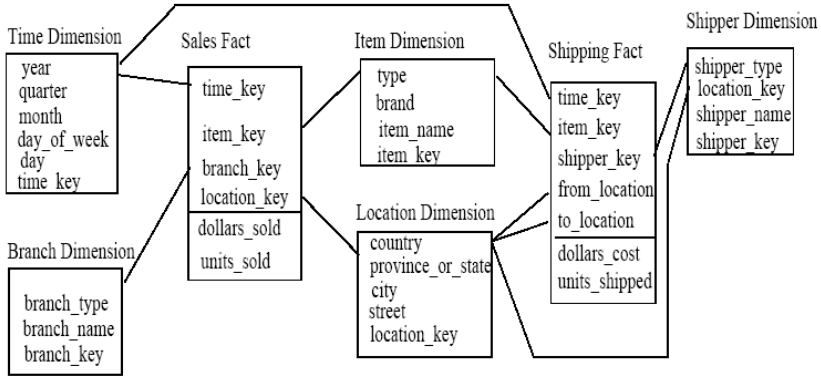
   Skema bola salju merupakan perluasan dari skema bintang dengan tambahan beberapa tabel dimensi yang tidak berhubungan secara langsung dengan tabel fakta. Tabel dimensi tersebut berhubungan dengan tabel dimensi yang lain. Berikut model skema bola salju :



Gambar 2.10 Model Skema Bola Salju (*Snowflake Schema*) [11]

3. *Fact constellations*

   Pada skema ini terdapat beberapa tabel fakta yang menggunakan satu atau beberapa tabel dimensi secara bersama-sama sehingga jika digambarkan akan terlihat seperti sekumpulan bintang. Skema ini juga dikenal dengan *galaxy schema*. Berikut model skema *constellations* :



Gambar 2.11 Model Skema *Constellations* [11]

2.2.4 ETL (*Extract, Transform, Loading*)

Menurut Inmon *Extract, Transform, Loading* (ETL) adalah "*the process of finding data, integrating it, and placing it in a data warehouse*".

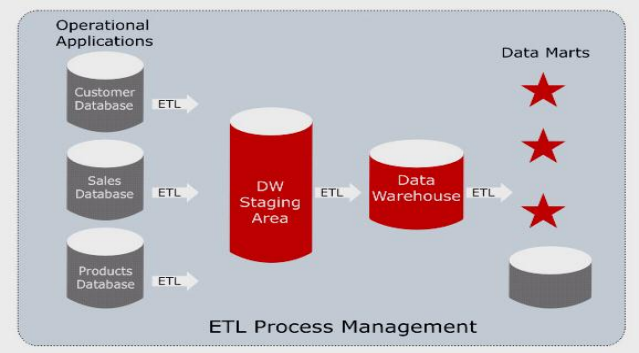
Proses ETL mengambil data dari *source systems* menggunakan *query*. ETL berkoneksi dengan *source system database* dan mengambil data dengan *query*. Setelah data hasil *query* diambil langkah selanjutnya dilakukan eksekusi proses ETL dan mengirimkannya ke *database data mart*.

ETL (*Extract, Transform, and Load*) adalah proses – proses dalam *data mart* yang meliputi:

a. Mengekstrak data dari sumber – sumber eksternal.
b. Mentransformasikan data ke bentuk yang sesuai dengan keperluan.
c. Memasukkan data ke target akhir, yaitu *data mart*.

ETL merupakan proses yang sangat penting, dengan ETL data dapat dimasukkan ke dalam *data mart*. ETL juga dapat digunakan untuk mengintegrasikan data dengan sistem yang sudah ada sebelumnya.

Tujuan ETL adalah mengumpulkan, menyaring, mengolah, dan menggabungkan data yang relevan dari berbagai sumber untuk disimpan ke dalam *data mart*. Hasil dari proses ETL adalah dihasilkannya data yang memenuhi kriteria *data mart* seperti data yang historis, terpadu, terangkum, statis, dan memiliki struktur yang dirancang untuk keperluan proses analisis. Manajemen proses ETL dapat dilihat pada gambar berikut :



Gambar 2.12 *ETL Process Management*

1. *Extract*

   Langkah pertama pada proses ETL adalah mengekstrak data dari sumber – sumber data. Kebanyakan proyek *data mart* menggabungkan data dari sumber – sumber yang berbeda. Sistem – sistem yang terpisah sangat mungkin menggunakan format data yang berbeda. Ekstraksi adalah mengubah data ke dalam suatu format yang berguna untuk proses transformasi.

   Pada hakekatnya proses ekstraksi adalah proses penguraian dari data yang diekstrak untuk mendapatkan struktur atau pola data yang diharapkan. Jika struktur atau pola data tidak sesuai dengan harapan maka data tidak dimasukkan ke dalam *data mart*.

2. *Transform*

   Tahapan transformasi menggunakan serangkaian aturan – aturan atau fungsi – fungsi ke dalam data yang telah diekstraksi, yang akan menentukan bagaimana data akan digunakan untuk analisis dan dapat melibatkan transformasi seperti penjumlahan data, *data encoding*, penggabungan data, pemisahan data, kalkulasi data, dan pembuatan *surrogate key*.

Data akan disimpan dalam bentuk *detail* dan ringkasan untuk menyediakan fleksibilitas dalam memenuhi kebutuhan informasi yang beragam pada pengguna. Data tersebut kemudian dihubungkan dengan *surrogate key* (sebuah *key* yang menghubungkan struktur *data mart* dan terpisah dari sistem sumber) dan agregasi untuk mempercepat proses analisis.

Output dari transformasi adalah data yang telah bersih dan konsisten dengan data yang tersimpan pada *data mart*, dan lebih jauh lagi adalah dalam bentuk yang telah siap untuk dianalisis oleh pengguna dari *data mart*.

3. *Loading*

Loading data ke dalam *data mart* dapat terjadi setelah semua transformasi dilakukan atau sebagai bagian dari proses transformasi. Ketika data dimasukkan ke dalam *data mart*, batasan – batasan tambahan yang didefinisikan di dalam skema *database* dan *trigger* diaktivasi ketika *loading* data akan dilakukan, yang juga akan memberikan kontribusi pada keseluruhan kualitas performa dari proses ETL. Jangka waktu proses *loading* bergantung pada kebutuhan organisasi.

2.2.5 Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak.

2.2.5.1 **Definisi UML**

UML (*Unified Modeling Language*) adalah sebuah bahasa untuk menetukan, visualisasi, kontruksi, dan mendokumentasikan *artifact* (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak. *Artifact* dapat berupa model, deskripsi atau perangkat lunak) dari system perangkat lunak, seperti pada pemodelan bisnis dan system non perangkat lunak lainnya [9].

UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya

digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan [9].

Bahasa Pemodelan UML lebih cocok untuk pembuatan perangkat lunak dalam bahasa pemrograman berorientasi objek (C+ , Java, VB.NET), namun demikian tetap dapat digunakan pada bahasa pemrograman prosedural.

Unified Modeling Language (UML) biasa digunakan untuk beberapa hal berikut ini, diantaranya :

1. Menggambarkan batasan sistem dan fungsi-fungsi sistem secara umum, dibuat dengan *use case* dan *actor*.
2. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.
3. Menggambarkan representasi struktur statik sebuah sistem dalam bentuk class diagrams.
4. Membuat model *behavior* "yang menggambarkan kebiasaan atau sifat sebuah sistem" dengan *state transition diagrams*.
5. Menyatakan arsitektur implementasi fisik menggunakan *component* and *development diagrams*.
6. Menyampaikan atau memperluas fungsionalitas dengan *stereotypes*.

UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan bahasa pemodelan visual yang memungkinkan pengembang sistem membuat *blue print* atas visinya dalam bentuk yang baku.

UML berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui sejumlah elemen grafis yang bisa dikombinasikan menjadi diagram. UML mempunyai banyak diagram yang dapat mengakomodasi berbagai sudut pandang dari suatu perangkat lunak yang akan dibangun. Diagram-diagram tersebut digunakan untuk hal-hal sebagai berikut :

1. Mengkomunikasikan ide.
2. Melahirkan ide-ide baru dan peluang-peluang baru.
3. Menguji ide dan membuat prediksi
4. Memahami struktur dan relasi-relasinya

2.2.5.2 Konsep Pemodelan Menggunakan Unified Modeling Language (UML)

Pemodelan menggunakan *Unified Modeling Language* (UML) merupakan metode pemodelan berorientasi objek dan berbasis visual. Karenanya pemodelan menggunakan UML merupakan pemodelan objek yang fokus pada pendefinisian struktur statis dan model sistem informasi yang dinamis dari pada mendefinisikan data dan model proses yang tujuannya adalah pengembangan tradisional.

UML menawarkan diagram yang dikelompokan menjadi lima perspektif berbeda untuk memodelkan suatu sistem. Seperti satu *set blue print* yang digunakan untuk membangun sebuah rumah. Jadi, hanya dibutuhkan sebuah UML agar dapat dibangun sebuah sistem.

2.2.5.3 Diagram Dasar dalam *Unified Modeling Language* (UML)

Diagram dasar Unified Modelling Language (UML) serta tujuannya dijelaskan seperti berikut :

a. Model *Use Case Diagram*

Use Case Diagram secara grafis menggambarkan interaksi antara sistem, sistem eksternal, dan pengguna. Dengan kata lain *Use Case diagram* secara grafis mendeskripsikan siapa yang akan menggunakan sistem dan dalam cara apa pengguna (*user*) mengharapkan interaksi dengan sistem itu. *Use Case* secara naratif digunakan untuk secara tekstual menggambarkan sekuensi langkah-langkah dari setiap interaksi.

b. Diagram Struktur Statis

UML menawarkan dua macam diagram untuk memodelkan struktur statis sistem informasi, yaitu:

1. *Class Diagram*: menggambarkan struktur objek sistem. Diagram ini menunjukan *class object* yang menyusun sistem dan juga hubungan antara *class object* tersebut.
2. *Object Diagram*: serupa dengan *class diagram*, tetapi *object diagram* memodelkan *instance object actual* dengan menunjukan nilai-nilai saat ini dari *atribut instance*. *Object Diagram* menyajikan “snapshot/potret” tentang objek sistem pada point waktu tertentu. Diagram ini tidak

digunakan sesering *Class Diagram*, tetapi saat digunakan dapat membantu seorang *developer* memahami struktur sistem secara lebih baik.

c. Diagram Interaksi

Diagram interaksi memodelkan sebuah interaksi, terdiri dari satu set objek, hubungan-hubungannya, dan pesan yang terkirim di antara objek. Model diagram ini memodelkan *behavior* (kelakuan) sistem yang dinamis dan UML memiliki dua diagram untuk tujuan ini, yaitu:

1. Diagram rangkaian/*Sequence Diagram*: secara grafis menggambarkan bagaimana objek berinteraksi dengan satu sama lain melalui pesan pada sekuensi sebuah *use case* atau operasi. Diagram ini mengilustrasikan bagaimana pesan terkirim dan diterima di antara objek dan dalam sekuensi atau timing apa.
2. Diagram kolaborasi/*Collaboration Diagram*: serupa dengan diagram rangkaian/sekuensi, tetapi tidak fokus pada *timing* atau sekuensi pesan. Diagram ini justru menggambarkan interaksi (atau kolaborasi) antara objek dalam sebuah format jaringan.
3. Diagram rangkaian maupun diagram kolaborasi merupakan *isomorphic* artinya kita dapat mengubah dari satu diagram ke diagram lain.

d. Diagram State/*State Diagram*

UML memiliki sebuah diagram untuk memodelkan *behavior* objek khusus yang kompleks (*statecahrt*) dan sebuah diagram untuk memodelkan *behavior* dari sebuah *use case* atau sebuah metode, yaitu:

1. Diagram *statechar*t: digunakan untuk memodelkan *behavior* objek khusus yang dinamis. Diagram ini mengilustrasikan siklus hidup objek-berbagai keadaan yang dapat diasumsikan oleh objek dan *event-event* (kejadian) yang menyebabkan objek beralih dari satu *state* ke *state* lain.
2. Diagram aktivitas/*Activity Diagram*: secara grafis digunakan untuk menggambarkan rangkaian aliran aktivitas baik proses bisnis maupun *use case*. *Activity diagram* dapat juga digunakan untuk memodelkan *action*

yang akan dilakukan saat sebuah operasi dieksekusi, dan memodelkan hasil dari action tersebut.

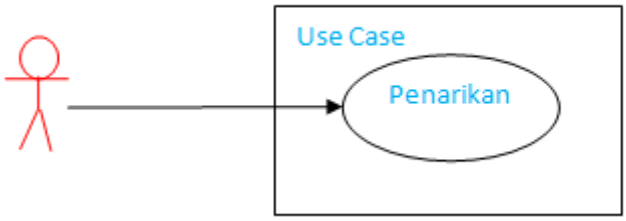
e. Diagram Implementasi

Diagram implementasi juga memodelkan struktur sistem informasi, yaitu sebagai berikut:

1. Diagram komponen/*Component Diagram*: digunakan untuk menggambarkan organisasi dan ketergantungan komponen-komponen pernagkat lunak sistem. Komponen diagram dapat digunakan untuk menunjukan bagaimana kode pemrograman dibagi menjadi modul-modul (atau komponen).
2. Diagram penguraian/*Deployment*: digunakan untuk mendeskripsikan arsitektur fisik dalam istilah ”node” untuk *hardware* dan *software* dalam sistem. Diagram ini menggambarkan konfigurasi komponen-komponen *software real-time*, prosesor, dan peralatan yang membentuk arsitektur sistem.

2.2.5.4 Use Case Diagram dan Aktor

Use Case diagram adalah model fungsional sebuah sistem yang menggunakan *actor* dan *use case*. *Use Case* adalah layanan (*services)* atau fungsi-fungsi yang disediakan oleh sistem untuk pengguna-penggunanya (Henderi et al, 2008). *Use Case* adalah suatu pola atau gambaran yang menunjukan kelakukan atau kebiasaan sistem. Setiap *Use Case* adalah suatu urut-urutan (*sequence*) transaksi yang saling berhubungan dan dilakukan oleh sebuah *actor* dan sistem dalam bentuk sebuah dialog (Henderi, 2007). *Use Case Diagram* dibuat untuk memvisualisasikan/ menggambarkan hubungan antara *Actor* dan *Use Case*. *Use Case diagram* mempresentasikan kegunaan atau fungsi-fungsi sistem dari perspektif pengguna.



Gambar 2.13 Contoh Aktifitas Aktor dan Use Case

Sebuah aktor mencirikan suatu bagian *outside* user atau susunan yang berkaitan dengan *user* yang berinteraksi dengan sistem [Rumbaugh, Booch, dan Jacobson 1999]. Dalam model *use case*, aktor merupakan satu-satunya kesatuan eksternal yang berinteraksi dengan sistem.

Terdapat beberapa variasi bagaimana aktor dibentuk [Fowler dan Scott 1999]. Sebuah aktor sering kali merupakan manusia (human user). Pada sejumlah sistem informasi, manusia adalah satu-satunya aktor. Dan mungkin saja dalam sistem informasi, seorang aktor bisa saja menjadi suatu sistem eksternal. Pada aplikasi real-time dan distribusi, sebuah aktor bisa saja menjadi satu perangkat eksternal I/O atau sebuah alat pengatur waktu. Perangkat eksternal I/O dan pengatur waktu aktor secara khusus lazimnya berada dalam *real-time* yang tersimpan dalam sistem (*real-time embedded systems*), sistem berinteraksi dengan lingkungan eksternal melalui sensor dan aktuator.

Primary actor (aktor utama) memprakarsai sebuah *use case*. Jadi, suatu *primary actor* memegang peran sebagai proaktif dan yang memulai aksi dalam sistem. Aktor lainnya yang berperan sebagai *secondary actor* bisa saja terlibat dalam *use case* dengan menerima *output* dan memberikan input. Setidaknya satu *actor* harus mendapatkan nilai dari *use case*. Biasanya adalah *primary actor* (aktor utama). Bagaimanapun, dalam *real-time embedded systems*, *primary actor* dapat berperan sebagai perangkat eksternal I/O atau pengatur waktu, penerima utama dari *use case* bisa menjadi *secondary human actor* yang menerima sejumlah informasi dari sistem.

2.2.5.5 *Class Diagram*

Class adalah kumpulan objek-objek dengan dan yang mempunyai struktur umum, *behavior* umum, relasi umum, dan semantic/kata yang umum. *Class-class* ditentukan/ditemukan dengan cara memeriksa objek-objek dalam *sequence diagram* dan *collaboration diagr*am. Sebuah *class* digambarkan seperti sebuah bujur sangkar dengan tiga bagian ruangan. *Class* sebaiknya diberi nama menggunakan kata benda sesuai dengan domain/bagian/kelompoknya (Whitten L. Jeffery et al, 2004).

Class Diagram adalah diagram yang menunjukan *class-class* yang ada dari sebuah sistem dan hubungannya secara logika. *Class diagram* menggambarkan struktur statis dari sebuah sistem. Karena itu *class diagram* merupakan tulang punggung atau kekuatan dasar dari hampir setiap metode berorientasi objek termasuk UML (Henderi, 2008). Sementara menurut (Whitten L. Jeffery et al 2004:432) class diagram adalah gambar grafis mengenai struktur objek statis dari suatu sistem, menunjukan *class-class* objek yang menyusun sebuah sistem dan juga hubungan antara class objek tersebut.

Elemen-eleman *class diagram* dalam pemodelan UML terdiri dari: Class-class, struktur class, sifat class (*class behavior*), perkumpulan/gabungan (*association*), pengumpulan/kesatuan (*agregation*), ketergantungan (*dependency*), relasi-relasi turunannya, keberagaman dan indikator navigasi, dan *role name* (peranan/tugas nama).

2.2.5.6 *Activity Diagram*

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

Activity diagram merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-trigger oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar subsistem)

secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

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

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

2.2.5.7 *Sequence Diagram*

Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi diantara obyek-obyek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh obyek-obyek yang melakukan suatu tugas atau aksi tertentu. Obyek-obyek tersebut kemudian diurutkan dari kiri ke kanan, aktor yang menginisiasi interaksi biasanya ditaruh di paling kiri dari diagram.

Pada diagram ini, dimensi vertikal merepresentasikan waktu. Bagian paling atas dari diagram menjadi titik awal dan waktu berjalan ke bawah sampai dengan bagian dasar dari diagram. Garis Vertical, disebut lifeline, dilekatkan pada setiap obyek atau aktor. Kemudian, *lifeline* tersebut digambarkan menjadi kotak ketika obyek melakukan suatu operasi , kotak tersebut disebut *activation bo*x. Obyek dikatakan mempunyai *live activation* pada saat tersebut.

2.2.5.8 *State 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.