

BAB II

LANDASAN TEORI

2.1. Sistem Informasi

Pengertian sistem informasi tidak bisa dilepaskan dari pengertian sistem dan informasi. Definisi dari sistem adalah sekelompok dua atau lebih komponen-komponen yang saling berkaitan (*interrelated*) atau subsistem-subsistem yang bersatu untuk mencapai tujuan yang sama (*common purpose*). Sedangkan definisi dari informasi adalah data yang diambil kembali, diolah, atau sebaliknya digunakan sebagai dasar untuk peramalan atau pengambilan keputusan. Sumber dari informasi adalah data. Data adalah fakta dan angka yang tidak sedang digunakan pada proses keputusan, dan biasanya berbentuk catatan historis yang dicatatkan dan diarsipkan tanpa maksud untuk segera diambil kembali untuk pengambilan keputusan. Secara lugas sistem informasi didefinisikan sebagai kumpulan orang, prosedur, *hardware*, *software* yang saling berinteraksi untuk memberikan suatu pelayanan informasi bagi user.

Sistem informasi memiliki tiga fungsi dasar :

1. Menerima data (*input*)
2. Mengubah data menjadi informasi (proses)
3. Untuk memproduksi dan mengkomunikasikan informasi ke dalam *timely fashion* bagi user untuk membuat keputusan (*output*). Sebagai contoh, banyak bank dan institusi keuangan yang menggunakan sistem informasi untuk membantu menentukan apakah nasabah diperbolehkan untuk melakukan pinjam.

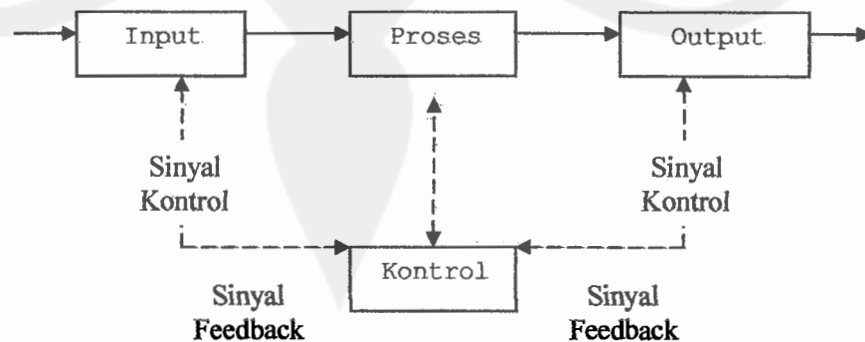
2.1.1. Konsep Dasar Sistem Informasi

Informasi merupakan sebuah sumber daya penting dalam sebuah organisasi. Informasi menjadi dasar, energi, atau mesin penggerak bagi organisasi penggunaannya. Dengan informasi yang tepat, sebuah organisasi dapat meningkatkan kinerja dan kualitas hasil kerjanya.

Definisi dari informasi adalah kumpulan data yang sudah diolah sehingga dapat dipergunakan sesuai dengan kebutuhan atau keperluan penggunaan informasi tersebut. Namun, informasi tidak akan bisa membantu sebuah organisasi penggunaannya jika tidak ditangani dengan baik. Penanganan informasi membutuhkan sebuah sistem yang mampu mengolah data-data menjadi informasi yang berguna bagi organisasi. Sistem sendiri dapat dikatakan sebagai kumpulan dari obyek atau elemen yang berinteraksi satu sama lain untuk mencapai tujuan tertentu. Menurut Turban, Rainer, Potter (2001) :

“Sistem informasi mengumpulkan, memproses, menyimpan, menganalisa, dan mengolah informasi untuk suatu tujuan tertentu.”

Sama seperti sistem lainnya, sistem informasi terdiri atas input (data, instruksi), proses, output (laporan, perhitungan) serta *feedback* dan kontrol. Sistem informasi memproses input dan menghasilkan output yang dikirimkan kepada user atau kepada sistem lainnya.



Gambar 2.1. Blok Pembangunan Sistem Informasi (Turban, 2001)

1. Blok Input

Blok Input adalah blok yang menerima semua bahan masukan berupa data yang diperlukan oleh proses dalam sistem informasi. Data yang didapatkan dari transaksi, kejadian, atau permintaan biasanya berupa teks, angka, simbol khusus, gambar. Penangkapan data dilakukan oleh unit-unit masukan yang berbentuk peralatan yang bertugas menangkap data dan mengubahnya ke dalam bentuk digital sehingga siap diproses oleh komputer.

2. Blok Proses

Blok proses melaksanakan semua proses transformasi yang akan mengubah masukan menjadi keluaran yang terjadi di dalam sistem informasi.

3. Blok Output

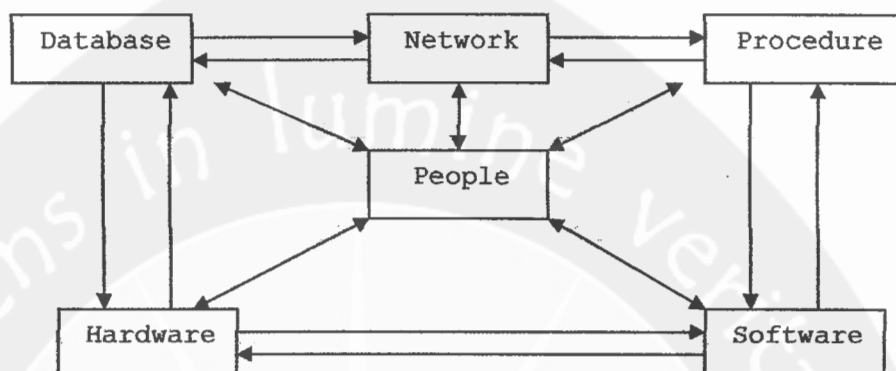
Blok Keluaran adalah blok yang bertugas menyajikan bentuk hasil sistem berupa informasi yang akan dikirim ke luar sistem.

4. Blok Kontrol & Feedback

Konsep sistem akan menjadi lebih efektif bila ditambahkan komponen tambahan berupa blok kontrol dan *feedback*. *Feedback* merupakan data mengenai kinerja sistem sedang kontrol merupakan pengawasan dan evaluasi dari *feedback* sehingga dapat dinilai apakah sistem itu sudah berjalan dengan baik sehingga dapat mencapai tujuan atau tidak. Jika sistem tidak berjalan dengan baik maka fungsi kontrol dapat membuat penyesuaian pada blok input dan komponen pemroses sistem sehingga dapat menghasilkan output yang sesuai dengan harapan.

2.1.2. Komponen Sistem Informasi

Komponen sistem informasi terdiri atas enam komponen dasar yang terdiri atas *hardware*, *software*, basis data, jaringan, prosedur, dan pengguna. Komponen tersebut digambarkan sebagai berikut :



Gambar 2.2. Enam Komponen Dasar Sistem Informasi (Turban, 2001)

1. Hardware

Hardware adalah sekumpulan piranti seperti *processor*, monitor, *keyboard* dan *printer* yang menerima semua bahan masukan berupa data dan informasi, memproses bahan masukan tersebut kemudian menampilkan data hasil pengolahan.

2. Software

Software adalah sekumpulan program komputer yang memungkinkan *hardware* untuk memproses data. *Hardware* komputer bekerja dengan efektif dengan adanya instruksi-instruksi yang kita berikan. Instruksi-instruksi tersebut tersimpan di dalam *software* komputer. *Software* tidak hanya mengatur *hardware* komputer untuk menggunakan sumber daya internal tapi juga mengatur sumber daya eksternal sehingga dapat digunakan secara bersamaan secara efektif.

3. Basis data

Basis Data adalah himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah. Perkembangan teknologi basis data sendiri tidak terlepas dari perkembangan perangkat keras dan perangkat lunak komputer. Perkembangan teknologi jaringan komputer dan komunikasi data adalah salah satu penyumbang kemajuan penerapan basis data, yang kemudian melahirkan sistem basis data terdistribusi. Perkembangan dunia perangkat lunak, seperti kecerdasan buatan, sistem pakar, dan pemrograman berorientasi obyek, juga sangat mempengaruhi perkembangan basis data.

4. Jaringan

Jaringan atau *Network* adalah sistem penghubung yang mengijinkan terjadinya pembagian data dan sumber daya antara komputer-komputer yang berbeda.

5. Prosedur

Prosedur adalah strategi, ketentuan, metode dan peraturan yang digunakan untuk mengatur penggunaan sistem informasi.

6. Pengguna

Pengguna atau *user* merupakan blok penting dalam sistem, termasuk orang yang menggunakan sistem informasi tersebut atau menggunakan keluaran dari sistem informasi tersebut.

2.1.3. Kapabilitas Sistem Informasi

Menurut Turban, Rainer, Potter (2001), Sistem Informasi yang berkualitas memiliki kemampuan :

1. Menyediakan proses transaksi yang cepat dan akurat

Informasi tersebut harus bebas dari kesalahan-kesalahan. Informasi harus secara jelas, karena ketidakakuratan informasi akan mengakibatkan keputusan yang tidak tepat. Suatu informasi yang sudah terlambat tidak akan mempunyai nilai lagi, karena informasi merupakan landasan didalam pengambilan keputusan.

2. Menyediakan kapasitas yang besar serta akses yang cepat

Sistem informasi harus menyediakan tidak hanya kapasitas yang besar untuk data-data tapi juga dapat menyediakan akses yang cepat terhadap data-data tersebut.

3. Menyediakan komunikasi yang cepat

Sistem informasi harus dapat menyediakan komunikasi yang cepat baik antar komputer, antar manusia serta antar manusia dan komputer baik berupa data, suara, gambar, dokumen dan video.

4. Mengurangi *Overload* Informasi

Sistem informasi harus didesain untuk mengurangi *overload* informasi agar dapat digunakan untuk mengambil keputusan secara efektif dan efisien.

5. Batasan Sistem Informasi

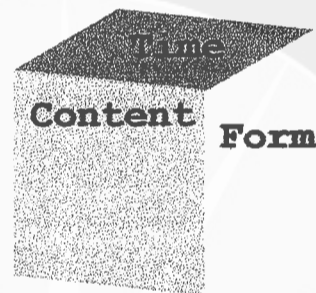
Suatu Sistem informasi diharapkan dapat menentukan batasan antar organisasi yang terdapat didalam sistem dengan organisasi yang terdapat diluar sistem tersebut. Batasan di dalam sistem informasi dapat membantu pengambilan keputusan pada area fungsional, proses dan komunikasi.

6. Menyediakan Bantuan Dalam Pengambilan Keputusan

Setiap orang dalam mengambil suatu tindakan atau keputusan memerlukan informasi yang berbeda-beda, sehingga yang disediakan harus sesuai dengan orang-orang yang betul-betul membutuhkan informasi tersebut.

2.1.4. Kualitas Sistem Informasi

Efektifitas blok keluaran pada konsep sistem dapat dilihat pada kualitas informasi yang disajikan.



Gambar 2.3. Kualitas Informasi

Informasi yang dihasilkan dikatakan berkualitas jika memenuhi 3 dimensi, yaitu :

1. *Time*

- a. *Timeliness*, informasi tersedia pada saat dibutuhkan.
- b. *Currency*, informasi yang tersedia harus merupakan informasi terbaru.
- c. *Frequency*, informasi dapat tersedia sesering mungkin.
- d. *Time period*, informasi yang tersedia dapat berupa informasi mengenai masa lampau, saat ini maupun yang akan datang.

2. Content

- a. *Accuracy*, informasi yang tersedia harus bebas dari kesalahan.
- b. *Relevance*, informasi yang tersedia harus berhubungan dengan penerima atau keadaan tertentu.
- c. *Completeness*, semua informasi yang diperlukan harus tersedia.
- d. *Conciseness*, hanya informasi yang diperlukan yang harus tersedia.
- e. *Scope*, informasi dapat memiliki batasan yang sempit atau luas serta fokus internal atau fokus eksternal.
- f. *Performance*, informasi dapat menunjukkan kinerja dengan cara mengukur aktivitas yang dilakukan, kemajuan yang dicapai.

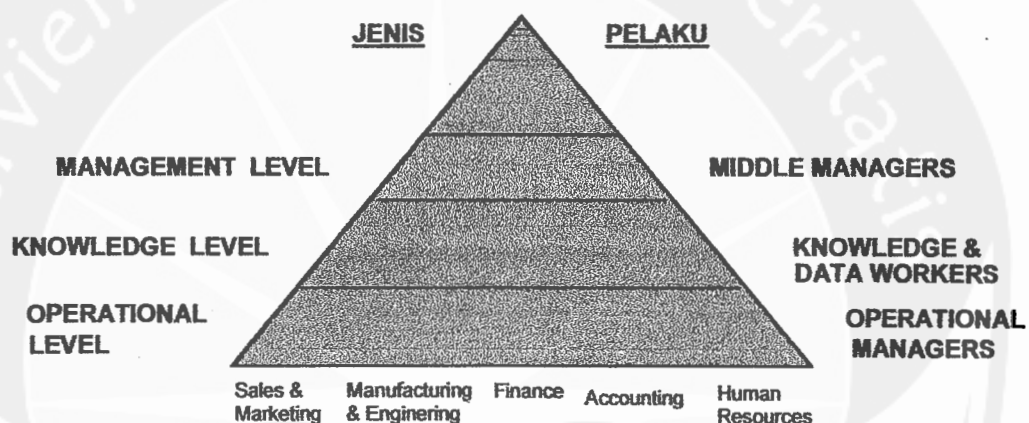
3. Form

- a. *Clarity*, informasi harus disediakan dalam bentuk yang mudah dipahami.
- b. *Detail*, informasi dapat menyediakan detail tertentu.
- c. *Order*, informasi dapat diatur menurut suatu aturan tertentu.
- d. *Presentation*, informasi dapat ditampilkan dalam bentuk suara, angka, grafik ataupun bentuk lainnya.
- e. *Media*, informasi dapat disediakan dalam bentuk dokumen, video atau media lainnya.

2.1.5 Sistem Informasi Skala Enterprise

Sistem informasi enterprise adalah sistem informasi yang digunakan pada skala enterprise atau perusahaan. Karena sifatnya yang enterprise maka inti yang ingin dicapai dari sistem informasi tersebut adalah

terbentuknya sebuah kualitas yang tinggi dari pelayanan (*service*), mengatasi data perusahaan yang volumenya sangat besar, dan mampu *men-support* sebuah organisasi atau perusahaan yang besar. Sistem informasi skala enterprise ini harus mampu untuk mengintegrasikan semua aspek yang terkait dengan *bussiness process* dari perusahaan yang bersangkutan, dan mampu membentuk sebuah sistem yang tersentralisasi sehingga semua departemen dalam perusahaan tersebut dapat mendapatkan informasi sesuai dengan kepentingannya masing-masing.



Gambar 2.4. Level Sistem Informasi Enterprise

Gambar di atas merupakan empat jenis sistem informasi beserta masing-masing pelakunya. Keempat jenis tersebut tergabung dalam tingkat-tingkat sesuai dengan jumlah pelaku yang terlibat di dalam masing-masing jenisnya. Berikut penjelasan singkat keempat jenis tersebut:

1. Sistem Level Operasional

Level yang paling banyak memerlukan sumber daya manusia karena sifatnya yang melakukan transaksi dan monitoring secara harian. Level ini berguna untuk mendukung kerja manajer operasional.

2. Sistem Level Knowledge

Merupakan level yang mendukung *knowledge* dan *data worker* untuk melakukan perancangan produk dan distribusi informasi. Informasi mengenai *knowledge* tersebut didapat dari level operasional.

3. Sistem Level Management

Level management ini mendukung *monitoring*, *controlling*, *decision-making*, dan aktivitas administrasi yang dilakukan oleh *middle manager* atau biasanya manajer kantor-kantor cabang perusahaan. Kegiatan management ini didasarkan pada *knowledge* yang didapat pada level sebelumnya.

4. Sistem Level Strategic

Level strategic merupakan level yang tertinggi, yang didalamnya terdapat aktivitas perencanaan untuk jangka panjang oleh *senior manager*. Disini biasanya muncul strategi-strategi perusahaan dikeluarkan untuk menjaga eksistensi dan berkompetisi dengan perusahaan lain.

2.2. Web Based Information System

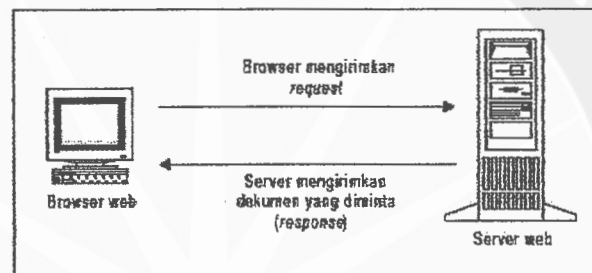
Saat ini komputer dan piranti pendukungnya telah masuk dalam setiap aspek kehidupan dan pekerjaan. Komputer yang ada sekarang memiliki kemampuan yang lebih dari sekedar perhitungan matematika biasa.

Interconnected Network yang biasanya sering disebut dengan Internet adalah sebuah sistem komunikasi global yang menghubungkan komputer-komputer dan bermacam jaringan komputer di seluruh dunia. Komputer dan jaringan dengan berbagai *platform* yang mempunyai perbedaan dan ciri khas masing-masing bertukar informasi dengan sebuah protokol *standard* yang dikenal dengan nama *TCP/IP*.

Web adalah fasilitas *hypertext* untuk menampilkan data berupa teks, gambar, bunyi, animasi, dan data multimedia lainnya, yang mana data tersebut saling berhubungan satu sama lainnya.

2.2.1. Web Server

Web server adalah suatu perangkat lunak yang mengatur halaman *web* dan membuat halaman-halaman *web* tersebut dapat diakses di klien, yaitu melalui jaringan lokal atau melalui jaringan *Internet*. Ada banyak *web server* yang tersedia diantaranya *Apache*, *IIS (Internet Information Service)*, dan *IPlanet's Enterprise server*.



Gambar 2.5 Konsep Dasar Browser dan Server Web

2.2.2. Web Browser

Web browser digunakan untuk menjelajah situs *web* lewat layanan *HTTP*. Untuk mengakses layanan *WWW (World Wide Web)* dari sebuah komputer digunakan program *web client* yang disebut *web browser* atau *browser* saja. Jenis-jenis *browser* yang biasa digunakan adalah *Internet Explorer*, *Netscape*, *NCSA Mosaic*, *Arena*, dan masih banyak lainnya.

2.2.3. Web Statis

Web statis merupakan suatu halaman yang berisi skrip HTML editor dan disimpan sebagai file *.htm* atau *.HTML*. Disebut statis karena halaman tersebut dari waktu ke waktu isinya tidak berubah. Karena halaman web statis ini tidak memerlukan pemrosesan di server, pembuatannya dapat dilakukan menggunakan editor HTML dan hasilnya dapat dilihat pada web browser.

2.2.4. Web Dinamis

Pembuatan halaman web dinamis dapat dilakukan dengan dua cara yaitu secara *client side* atau secara *server side*. Penggunaan *client side* dan *server side* tidak saling bertentangan melainkan saling melengkapi. Seorang web developer harus dapat menentukan bagian mana yang diletakkan secara *client side* dan mana yang diletakkan secara *server side*.

2.2.4.1. Web Dinamis Client Side

Di dalam model *client side* ini, terdapat modul-modul atau *plug-in* yang ditambahkan ke suatu browser untuk menciptakan halaman web dinamis. Pada umumnya skrip HTML dikirim ke browser bersama dengan file yang berisi sekumpulan instruksi, dan file tersebut mengacu pada isi halaman HTML tersebut. Akan tetapi, juga merupakan hal yang umum jika sekumpulan instruksi tersebut berada menjadi satu dengan file HTML. Kemudian, browser menggunakan sekumpulan instruksi tersebut untuk menghasilkan skrip HTML ketika terdapat request pada halaman tersebut. Singkatnya, halaman tersebut dihasilkan secara dinamis pada saat terdapat request.

Kelemahan pada *client side* :

1. Butuh waktu untuk men-*download file* instruksi terutama jika *file* instruksi tersebut berada secara terpisah dan jumlahnya lebih dari satu.
2. Tidak semua *browser* dapat menterjemahkan dan setiap *browser* mempunyai cara berbeda untuk menterjemahkannya.
3. Merupakan suatu masalah untuk membuat skrip bersifat *client side* yang membutuhkan sumber daya dari *server* misalnya *database* di *server*.
4. Setiap skrip *client side* dapat dilihat oleh semua orang, hal ini bisa jadi tidak diinginkan oleh pembuatnya.

Walaupun model *client side* memiliki beberapa kelemahan seperti yang telah disebutkan di atas, akan tetapi model *client side* tetap dibutuhkan oleh seorang *web developer*. *Client side* sangat berguna untuk mengadakan interaksi dengan *user* dalam frekuensi yang cukup tinggi dan data yang diperlukan relatif sedikit dan telah tersedia sebelumnya.

2.2.4.2. Web Dinamis Server Side

Pada model *server side*, skrip *HTML* yang di dalamnya juga terdapat sekumpulan instruksi, dikirimkan ke *web server*. Seperti halnya *client side*, sekumpulan instruksi tersebut digunakan untuk menghasilkan skrip *HTML* ketika ada *request* terhadap halaman tersebut. Perbedaannya adalah kumpulan instruksi pada *client side* diproses di *client* atau di komputer *user* yang me-*request*-nya, sedangkan pada model *server side* kumpulan instruksi tersebut diproses di komputer *server*.

2.3 Pengertian CRM (*Customer Relationship Management*)

Customer Relationship Management (CRM) adalah salah satu bentuk aplikasi Teknologi Informasi (TI) di departemen penjualan atau marketing suatu perusahaan. Tetapi ide atau nafas dari CRM sendiri tidak lain dan tidak bukan adalah pelayanan, contoh nyatanya bisa ditanyakan ke penjaga warung "nasi kucing" dekat rumah yang mampu menjaga pelanggannya selama bertahun-tahun.

Setelah era *digital* yang selalu menggunakan "e-" didepan semua kata seperti *e-commerce*, *e-book*, *e-sales*, *e-mail*, dan sebagainya. *Customer Relationship Management* atau CRM merupakan akronim atau singkatan yang paling populer dikalangan orang-orang *sales* dan *marketing*. Kalau diterjemahkan ke dalam bahasa Indonesia kira-kira adalah Manajemen Hubungan Pelanggan (MHP).

Pelanggan atau *customer*, kata pertama, di kamus bahasa Inggris artinya adalah seseorang yang berulang kali atau teratur melakukan pembelian kepada seorang pedagang. Jadi pelanggan adalah orangnya (dalam definisi ini tidak disinggung tentang kepuasan, harga, dan aspek-aspek lain). Hubungan atau *relationship*, kata kedua, adalah bentuk *komunikasi* dua arah antara pembeli dan penjual. Manajemen, kata terakhir, artinya pengelolaan (secara luas tanpa perlu menjabarkan detail bagaimana mengelola sesuatu). Jadi definisi diatas kalau digabungkan kira-kira menjadi pengelolaan hubungan dua arah antara suatu perusahaan dengan orang yang menjadi pelanggan di perusahaan tersebut. (Mandeep Khera, 2000)

CRM dalam perkembangannya juga bisa didefinisikan sebagai berikut:

- a. *CRM* adalah sebuah istilah industri TI untuk metodologi, strategi, perangkat lunak (*software*) dan atau aplikasi berbasis *web* lainnya yang mampu membantu sebuah perusahaan (*enterprise*, jika skalanya besar) untuk mengelola hubungannya dengan para pelanggan.
- b. *CRM* adalah usaha sebuah perusahaan untuk berkonsentrasi menjaga pelanggan (supaya tidak lari ke pesaing) dengan mengumpulkan segala bentuk interaksi pelanggan baik itu lewat telepon, *email*, masukan di situs atau hasil pembicaraan dengan *staff sales* dan *marketing*.
- c. *CRM* adalah sebuah strategi bisnis menyeluruh dari suatu perusahaan yang memungkinkan perusahaan tersebut secara efektif bisa mengelola hubungan dengan para pelanggan.

Dari penjelasan di atas, memang cukup sulit untuk menemukan apa definisi *CRM* yang tepat. Hal ini dikarenakan luasnya cakupan *CRM* terhadap aktivitas sales dan *marketing* yang pada akhirnya menjadi bagian dari manajemen pengetahuan (*knowledge management*) dari perusahaan itu sendiri. Kemudian akan timbul pertanyaan mengenai apa hubungan antara Teknologi Informasi dengan *Customer Relationship Management*? Untuk menjawab hal tersebut, perlu ditinjau ulang keberadaan TI terhadap *CRM*. TI merupakan bagian terpenting (tapi bukan yang utama) dari *CRM* karena tanpa kemampuan teknologi komputer yang handal untuk mengolah besarnya informasi yang berhasil dikumpulkan, *CRM* akan lumpuh dan tidak ada artinya.

2.3.1 Sejarah CRM

Dengan adanya internet, tembok penghalang yang membatasi pelaku bisnis baru untuk melakukan penetrasi pasar menjadi sangat kabur, bahkan mungkin dapat dikatakan tidak ada. Sekarang seorang pelaku bisnis baru bisa melakukan bisnisnya dengan rekanan dari luar kota, negara bahkan lintas benua.

Perbedaan teknologi sudah tidak penting lagi karena para "pemain baru" bisa mendapatkan teknologi yang sama itu dengan mudah. Jadi jika perusahaan besar masih mengandalkan produknya (*product focus*) niscaya mereka bakal tersusul oleh "pemain-pemain baru" tersebut. Hal ini dikarenakan faktor pembeda produk pemain lama dan pemain baru sudah tidak terlalu signifikan.

Karena hal inilah perusahaan-perusahaan besar (*enterprise*) mencoba mengganti arah bisnis mereka fokus ke pelanggan (*customer focus*). Fakta mengatakan bahwa biaya untuk mendapatkan pelanggan baru bisa 10 kali biaya untuk menjaga pelanggan yang sudah ada. Sementara itu pelanggan juga memiliki kecenderungan untuk tidak hanya 'transaksional', beli barang - dapat barang - selesai, tetapi ingin lebih, beli barang - memilih service - dapat barang dan service - beli lagi. Sehingga pelanggan menginginkan hubungan dua arah dengan perusahaan-perusahaan yang mampu memberikan pelayanan tambahan tanpa terasa "mahal" (Simon Harper, 2000).

Selain kecenderungan naiknya kompetisi global diatas dimana produk-produk yang ada sangat sulit dibedakan, kecenderungan lain yang melahirkan CRM adalah majunya teknologi manajemen data yang memungkinkan disimpannya jutaan data dalam satu sistem tunggal. Dua puluh tahun yang lalu, informasi tentang pelanggan akan

disimpan dalam dua puluh sistem yang berbeda dan tidak ada teknologi yang mampu mengelola informasi kompleks tentang pelanggan. Data yang ada bisa dari telepon, email, faksimili, tulisan tangan staf sales dan marketing, kartu nama, dan masih banyak lagi.

CRM berawal sekitar tahun 1990-an sebagai langkah logis setelah mulai booming-nya *Enterprise Resource Planning (ERP)*. Satu faktor lagi yang melahirkan CRM adalah penelitian marketing oleh PIMS yang menyimpulkan bahwa "pelanggan yang menggerutu tidak puas akan bercerita kepada sekitar 7-10 orang temannya sedangkan pelanggan yang puas akan merekomendasikan perusahaan bersangkutan ke 3-4 teman mereka" (Mandeep Khera, 2000).

2.3.2 Implementasi CRM

Untuk mengimplementasikan sebuah strategi CRM, diperlukan paling tidak 3 (tiga) faktor kunci, yaitu:

1. Orang-orang yang profesional (kualifikasi memadai).
2. Proses yang didesain dengan baik.
3. Teknologi yang memadai (*leading-edge technology*).

Tenaga yang profesional tidak saja mengerti bagaimana menghadapi pelanggan tetapi juga mengerti cara menggunakan teknologi (untuk CRM). Apapun tanpa desain yang baik akan gagal, begitu juga CRM. Perusahaan pengguna CRM harus sudah mengetahui tujuan (*business objectives*) dan tuntutan bisnis (*business requirements*) yang diinginkan dari implementasi CRM ini (Cliff Findlay, 2000).

Berikut adalah elemen-elemen yang sebaiknya dimiliki oleh teknologi CRM:

- a. **Aturan-aturan Bisnis:** tergantung dari kompleksitas transaksi, aturan-aturan bisnis harus dibuat untuk

memastikan bahwa transaksi dengan pelanggan dilakukan dengan efisien. Misalnya pelanggan dengan pembelian besar yang mendatangkan keuntungan besar harus dilayani oleh staf penjualan senior dan berpengalaman.

- b. **Penggudangan Data (*data warehousing*)**: konsolidasi dari informasi tentang pelanggan harus dilakukan dalam satu sistem terpadu. Hasil analisa harus mampu menampilkan petunjuk-petunjuk tertentu tentang pelanggan sehingga staf penjualan dan marketing mampu melakukan kampanye terfokus terhadap grup pelanggan tertentu. Nantinya gudang data (*data warehouse*) ini juga harus mampu menaikkan volume penjualan dengan *cross-selling* atau *up-selling*.
- c. **Situs (*web*)**: CRM juga harus memiliki kemampuan swalayan. Hanya aplikasi berbasis situs (*web based*) yang bisa mendukung ini. Pelanggan bisa melakukan transaksi sendiri, tahu berapa yang harus dibayar, dan sebagainya.
- d. **Pelaporan (*reporting*)**: teknologi CRM harus mampu menghasilkan laporan yang akurat dan komprehen, nantinya berguna untuk menganalisa kelakuan pelanggan.
- e. **Meja Bantu (*helpdesk*)**: teknologi yang mampu mengintegrasikan informasi pelanggan ke aplikasi meja bantu akan menunjukkan ke pelanggan seberapa serius sebuah *enterprise*. (Sambasivan, 2000).

Pertanyaan mendasar yang muncul paling awal adalah apakah sebuah perusahaan perlu CRM? Dari penjabaran dan analisa di atas, jawabannya, perlu! Karena dengan tren bisnis yang ada, tujuan utama sekarang adalah meningkatkan loyalitas pelanggan ke perusahaan kita. Dengan loyalitas pelanggan yang meningkat, kelangsungan bisnis masa depan bisa berlanjut.

Perusahaan-perusahaan yang menerapkan CRM mendapatkan pengertian yang lebih baik tentang pelanggan mereka dan kebutuhan mereka. Dikombinasikan dengan penggudangan data, bank data dan meja bantu atau pusat panggilan (*call center*), CRM membantu perusahaan untuk mengumpulkan dan menganalisa informasi tentang sejarah pelanggan, apa-apa yang menjadi kesenangannya, apa saja keluhannya dan bahkan data lain untuk memperkirakan apa yang pelanggan akan beli di masa datang. Banyak perusahaan menawarkan solusi-solusi CRM yang "tinggal pakai" yang memiliki tingkat fungsionalitas yang beragam. Bisnis proses yang terintegrasi dan konsolidasi data yang sudah ada saat implementasi CRM yang bakal menjamin suksesnya CRM itu sendiri.

2.3.3. Manfaat dan Tantangan CRM

Potensi manfaat bisnis dari manajemen hubungan pelanggan sangat banyak. Contohnya, CRM memungkinkan sebuah perusahaan untuk mengidentifikasi serta berfokus pada para pelanggan terbaik mereka (yaitu mereka yang paling menguntungkan bagi perusahaan) agar mereka dapat dipertahankan sebagai pelanggan seumur hidup untuk layanan yang lebih besar dan menguntungkan. Manajemen hubungan pelanggan memungkinkan penyesuaian dan personalisasi *real-time* atas berbagai produk dan jasa berdasarkan pada keinginan, kebutuhan, kebiasaan membeli serta siklus hidup para pelanggan. CRM juga dapat menelusuri saat ketika pelanggan menghubungi perusahaan, darimana pun titik hubungannya. Sistem CRM juga memungkinkan perusahaan untuk memberi pengalaman yang konsisten dan layanan serta dukungan superior bagi pelanggan, di semua titik kontak yang dipilih oleh

pelanggan. Semua manfaat ini akan memberi nilai bisnis srrraregis bagi perusahaan dan nilai pelanggan yang lebih besar bagi para pelanggannya.

Dapat disimpulkan bahwa *CRM* membantu perusahaan untuk mengembangkan produk baru berdasarkan pengetahuan yang lengkap tentang keinginan pelanggan, dinamika pasar dan pesaing dengan cara:

- a. Menjaga pelanggan yang sudah ada.
- b. Menarik pelanggan baru.
- c. *Cross Selling* : menjual produk lain yang mungkin dibutuhkan pelanggan berdasarkan pembeliannya.
- d. *Upgrading*: menawarkan status pelanggan yang lebih tinggi (gold card vs. silver card).
- e. Identifikasi kebiasaan pelanggan untuk menghindari penipuan.
- f. Mengurangi resiko operasional karena data pelanggan tersimpan dalam satu sistem.
- g. Respon yang lebih cepat ke pelanggan.
- h. Meningkatkan efisiensi karena otomasi proses.
- i. Meningkatkan kemampuan melihat dan mendapatkan peluang.

2.3.4 Kegagalan CRM

Manfaat bisnis dari manajemen hubungan pelanggan tidak dijamin dan, malahan, terbukti menjebak bagi banyak perusahaan. Berbagai survei yang dilakukan kelompok peneliti industri meliputi sebuah laporan yang menyatakan bahwa lebih dari 50 persen proyek *CRM* tidak memberi hasil yang dijanjikan. Di dalam laporan lainnya 20 persen dari perusahaan yang disurvei melaporkan bahwa implementasi *CRM* benar-benar telah merusak hubungan lama mereka dengan pelanggan. Selain itu, di dalam survei mengenai kepuasan

manajemen senior atas 25 amat manajemen, CRM diletakkan pada urutan hampir terbawah dari segi kepuasan walaupun 72 persen dari mereka berharap dapat mengimplememasikan CRM waktu dekat.

Apa alasan untuk tingginya tingkat kegagalan atau ketidakpuasan dengan usaha yang berkaitan dengan CRM? Penelitian menunjukkan bahwa alasan sangatlah umum, yaitu kurangnya pemahaman dan persiapan. Dengan kata lain, sering kali para manajer perusahaan bergantung pada aplikasi baru terkenal dari teknologi informasi (seperti CRM) untuk mengatasi masalah bisnis tanpa mengembangkan terlebih dahulu perubahan proses bisnis dan program manajemen perubahan yang dibutuhkan. Contohnya, dalam banyak kasus, proyek CRM yang gagal diimplementasikan tanpa partisipasi dari pihak-pihak yang berkepentingan dengan perusahaan (*stakeholder*) yang terlibat dengan proyek tersebut. Oleh karena itu, karyawan dan pelanggan tidak siap untuk berbagai proses atau tantangan baru yang merupakan bagian dari implementasi baru CRM.

2.3.5. Metode Perancangan CRM

Metode yang digunakan dalam penyusunan Tugas Akhir dengan topik membangun Aplikasi CRM ini adalah:

1. Metode Studi Literatur dan studi perangkat lunak

Metode ini dilakukan dengan cara membaca dan memahami buku-buku referensi, perangkat lunak yang sudah ada, dan media-media lain yang relevan dengan *Object Relational Database* dan *Customer Relationship Management*.

2. Analisis

Menganalisis data dan *business rule* yang ada untuk melakukan analisis dan perancangan *CRM*.

3. Wawancara

Melakukan wawancara dengan calon pengguna (*user*) dalam hal *future-fiture* dan desain *interface* yang diinginkan.

4. Pembangunan aplikasi *CRM* yang meliputi:

- a) Pengambilan informasi yang diperlukan dari data yang didapat dan hasil wawancara mengenai *business rule*.
- b) Pembangunan database dengan konsep *Object Relational Database* berdasarkan data dan *business rule* yang didapat.
- c) Desain *interface* aplikasi *CRM* berbasis desktop.
- d) Pengimplementasian (*code*) aplikasi *CRM*.

5. Pengujian

Menguji kehandalan aplikasi *CRM* dalam membantu melakukan manajemen dan perencanaan strategi pemasaran.

2.3.6 Fokus Bisnis Dalam *CRM*

Mengelola berbagai hubungan dengan pelanggan melibatkan dua tujuan yang saling berkaitan : pertama, memberi organisasi dan semua karyawannya yang berhadapan dengan pelanggan, satu pandangan lengkap tentang setiap pelanggan di setiap hal di lintas semua saluran, dan kedua, memberi pelanggan satu pandangan lengkap tentang perusahaan dan saluran-salurannya yang luas. Ini adalah sebab mengapa perusahaan berpaling ke manajemen hubungan pelanggan untuk membantu mereka menjadi bisnis yang berfokus pada pelanggan. *CRM* menggunakan teknologi informasi untuk membuat sistem lintas fungsi perusahaan

yang mengintegrasikan dan mengotomasikan banyak proses layanan pada pelanggan dalam penjualan, pemasaran, dan layanan pelanggan yang berinteraksi dengan pelanggan perusahaan. Sistem CRM ini juga menciptakan kerangka kerja TI software serta database yang dijalankan melalui Web, yang mengintegrasikan proses-proses ini dengan operasi bisnis perusahaan lainnya, dan mendukung kerja sama antara perusahaan dengan para pelanggan serta mitranya.

Sistem CRM meliputi sekelompok modul software yang memberikan berbagai alat, yang membantu perusahaan dan para karyawannya memberikan layanan yang cepat, dapat diandalkan, dan konsisten ke para pelanggannya. Siebel Systems, Oracle, PeopleSoft, SAP AG, dan Epiphany adalah beberapa penjual utama software CRM. Terdapat beberapa komponen utama aplikasi CRM, yaitu manajemen kontak dan rekening, penjualan, pemasaran dan pemenuhan pesanan, layanan dan dukungan untuk pelanggan, serta program retensi dan loyalitas.

2.3.6.1. Manajemen Kontak dan Rekening

Software CRM membantu para praktisi penjualan, pemasaran, dan layanan untuk menangkap serta menelusuri data yang relevan mengenai setiap kontrak yang sudah lewat atau yang direncanakan dengan para pelanggan atau calon pelanggan, seperti juga kegiatan siklus hidup dan bisnis lainnya dari pelanggan. Informasi ditangkap dari semua titik persinggungan, seperti telepon, faks, email, situs Web perusahaan, toko ritel, kios, dan kontak person. Sistem CRM menyimpan data tersebut dalam database umum untuk pelanggan yang mengintegrasikan semua informasi rekening pelanggan serta membuatnya tersedia di

seluruh perusahaan melalui internet, intranet, atau hubungan jaringan lainnya untuk aplikasi penjualan, pemasaran, layanan, dan aplikasi CRM lainnya.



Gambar 2.6 Hubungan terintegrasi antara sebuah perusahaan dengan mitranya (Turban, 2001)

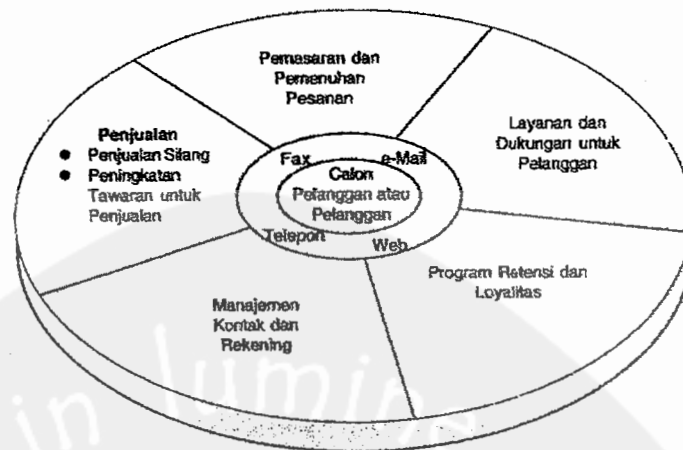
2.3.6.2. Penjualan

Sistem CRM memberikan para staff penjualan alat software dan sumber data perusahaan yang mereka butuhkan untuk mendukung serta mengelola aktivitas penjualan mereka, dan mengoptimalkan penjualan silang dan peningkatan tawaran untuk penjualan, contohnya meliputi prospek penjualan dan informasi produk, konfigurasi produk, dan kemampuan pembuatan daftar harga penjualan, CRM juga memberi mereka akses *real-time* ke satu tampilan umum alas pelanggan, hingga memungkinkan mereka untuk memeriksa semua aspek dari status rekening pelanggan dan sejarahnya, sebelum menjadwalkan panggilan telepon untuk

penjualan mereka. Contohnya, sistem CRM akan memperingatkan staf penjualan sebuah bank untuk menelepon nasabah yang melakukan penyimpanan besar, agar dapat menjual layanan kredit utama atau investasi. Atau, sistem tersebut akan memperingatkan seorang tenaga penjualan atas layanan yang belum dipenuhi, masalah pengiriman atau pembayaran, yang dapat diatasi melalui hubungan personal dengan pelanggan.

2.3.6.3. Pemasaran dan Pemenuhan Pesanan

Sistem CRM membantu para praktisi pemasaran menyelesaikan kampanye pemasaran langsung dengan mengotomatisasi tugas-tugas seperti pengkualifikasian pemasaran pada sasaran, dan penjadwalan serta penelusuran pengiriman surat pemasaran langsung. Kemudian, *software CRM* akan membantu para praktisi pemasaran untuk menangkap dan mengelola data respons pelanggan dan calon pelanggan di database CRM, serta menganalisis nilai pelanggan dan nilai bisnis dari kampanye pemasaran langsung perusahaan. CRM juga membantu dalam pemenuhan respons calon pelanggan serta pelanggan dengan secara cepat menjadwalkan kontak penjualan serta memberi informasi yang tepat atas produk dan jasa bagi mereka, sementara sambil menangkap informasi yang relevan untuk database CRM.



Gambar 2.7 Kelompok Aplikasi Utama Dalam CRM (Turban, 2001)

2.3.6.4. Layanan dan Dukungan Untuk Pelanggan

Sistem CRM memberi para staf penjualan alat *software* dan akses *real-time* ke *database* umum pelanggan yang dapat dibagi bersama dengan para praktisi penjualan dan pemasaran. CRM membantu para manajer layanan pelanggan membuat, menetapkan, dan mengelola berbagai permintaan atas layanan dari pelanggan. *Software call center* mengirimkan semua panggilan ke para staf dukungan untuk pelanggan berdasarkan keahlian serta otoritas mereka untuk menangani permintaan layanan tertentu. *Software help desk* membantu para staf layanan untuk pelanggan membantu para pelanggan yang memiliki masalah dengan suatu produk atau jasa, dengan memberi data layanan dan saran yang relevan untuk mengatasi masalah tersebut. Layanan mandiri berbasis *web* memungkinkan para pelanggan mengakses dengan mudah informasi pendukung pribadi di situs *web* perusahaan, dengan tetap memberi mereka pilihan untuk menerima bantuan lebih jauh secara *online* atau melalui telepon dari personel layanan pelanggan.

2.3.6.5. Program Retensi dan Loyalitas

- a. Biaya enam kali lebih besar untuk menjual ke seorang pelanggan baru daripada menjual ke pelanggan lama.
- b. Pelanggan yang tidak puas biasanya akan memberitahukan tujuh hingga sepuluh orang lainnya mengenai pengalamannya, sedangkan pelanggan yang puas biasanya akan memberi tahukan pengalaman mereka ke 3 atau 4 orang yang lain.
- c. Perusahaan dapat meningkatkan labanya hingga 85 persen dengan meningkatkan retensi terhadap pelanggannya sebesar 5 persen setahun.
- d. Kepastian untuk menjual produk ke pelanggan baru adalah 15 persen, sedangkan kepastian untuk menjual produk ke pelanggan lama adalah 50 persen.
- e. 70 persen pelanggan yang mengeluh akan melakukan bisnis dengan perusahaan itu lagi jika perusahaan tersebut dengan cepat menangani kekacauan layanannya.

Inilah mengapa meningkatkan dan mengoptimalkan retensi serta loyalitas pelanggan adalah strategi bisnis utama dan tujuan utama dari manajemen hubungan dengan pelanggan. Sistem CRM mencoba untuk membantu perusahaan mengidentifikasi, memberi penghargaan, dan memasarkan kepada para pelanggan perusahaan yang paling menguntungkan dan loyal. *SoftWare* analitis CRM meliputi alat penambangan data (*data mining*) dan *software* analitis pemasaran lainnya, sementara *database* CRM dapat berisi gudang data pelanggan, dan *data mart* CRM. Alat-alat ini digunakan untuk mengidentifikasi para pelanggan yang menguntungkan dan loyal serta mengarahkan dan mengevaluasi program pemasaran ke sasaran, dan pemasaran relasi ke para pelanggan tersebut.

2.4. ORDBMS (Object-Relational Database Management System)

Pada RDBMS (Relational Database Management System), data direpresentasikan ke dalam bentuk tabel dengan kolom dan baris. Baris pada tabel direpresentasikan sebagai nilai dan kolom sebagai atribut. Suatu tabel dapat berhubungan dengan tabel lainnya dengan menggunakan salah satu ataupun banyak kolom sebagai penghubungnya.

Ada beberapa masalah yang dihadapi para pengembang yang menggunakan RDBMS. Pengembang dapat menggunakan tipe data yang telah tersedia di dalam RDBMS, namun para pengembang tidak dapat menambahkan objek nyata seperti murid atau alamat sebagai suatu tipe data. Pada bahasa Pemrograman Berorientasi Objek/*Object Oriented Programming* pengembang dapat memodelkan objek nyata dengan cara merepresentasikannya ke dalam bentuk *class*.

RDBMS cocok untuk aplikasi yang bersifat memproses suatu transaksi, seperti aplikasi *accounting* dimana tipe data yang digunakan sederhana dan jumlahnya sedikit. Namun, RDBMS menawarkan fitur yang terbatas untuk pengembangan aplikasi sejenis CAD (*Computer Aided Design*) dimana tipe data yang digunakan banyak dan sifatnya lebih kompleks. Aplikasi semacam itu membutuhkan struktur bersarang yang tidak disediakan dalam fitur RDBMS.

Salah satu keuntungan dari bahasa pemrograman berorientasi objek seperti C++ ataupun JAVA adalah kemudahan untuk menciptakan komponen yang sifatnya *reusable*. Empat fitur kunci OOP yakni penamaan abstrak (*namely abstraction*), pembungkusan (*encapsulation*), pewarisan (*inheritance*), dan polimorfisme (*polymorphism*)

memudahkan user menciptakan aplikasi yang sifatnya *scalable* dan efisien. Fitur-fitur tersebut memungkinkan kita untuk menggabungkan data dan operasi/method yang digunakan untuk memanipulasi data tersebut menjadi satu unit.

Untuk *men-support* penyimpanan dan pengambilan data di basis data dari aplikasi berbasis objek, programmer harus mengintegrasikan aplikasi dengan DBMS. Programmer dapat mengintegrasikan RDBMS dengan aplikasi yang kita buat sebagai komponen tambahan atau *network service*. Namun merupakan hal yang sulit untuk mengintegrasikan aplikasi berorientasi objek dengan RDBMS karena secara fundamental *object oriented* dan *relational model* berbeda.

Oleh karena itu dikembangkan suatu model basis data yang menggabungkan fitur RDBMS dan fitur dalam bahasa pemrograman berorientasi objek yang disebut *Object Relational Database Management System*. Dalam hal ini beberapa vendor basis data telah mengintegrasikan konsep ORDBMS, dan salah satunya adalah Oracle yang memunculkan fitur ORDBMS dalam produknya Oracle 9i.

2.4.1. Fitur ORDBMS

ORDBMS merupakan generasi berikutnya dari DBMS yang mengkombinasikan fitur *Object Oriented Programming* ke dalam RDBMS. Keuntungan dari menggunakan ORDBMS adalah konversi data antara format RDBMS dan format *object oriented* disediakan bagi para penggunanya. Oleh karena itu, programmer dapat mengakses basis data melalui bahasa pemrograman berorientasi objek. Berikut ini fitur-fitur dalam ORDBMS :

1. Extensibility

Fitur ini memungkinkan *programmer* untuk mendefinisikan tipe data baru. Dengan kata lain user dapat mengelola data yang sifatnya kompleks seperti *image*, *audio* dan *video*.

2. Encapsulation

Fitur ini memungkinkan *programmer* menciptakan suatu atribut dan *method* di dalam suatu *object type*, dimana jika ingin mengakses atribut atau fungsi itu *programmer* harus membuat instance dari *object type* tersebut. *Object type* membungkus struktur data bersama dengan *method* untuk memanipulasi data.

3. Inheritance

Dengan fitur ini suatu tabel atau tipe data dapat mewarisi properti yang dimiliki oleh tabel atau tipe data lain. Properti yang ikut diwariskan dapat berupa : struktur, index, fungsi atau prosedur, *constraint*, dan *trigger*.

4. Polymorphism

Dengan fitur ini *programmer* dapat membuat banyak fungsi yang memiliki nama sama namun dengan paramater yang berbeda.

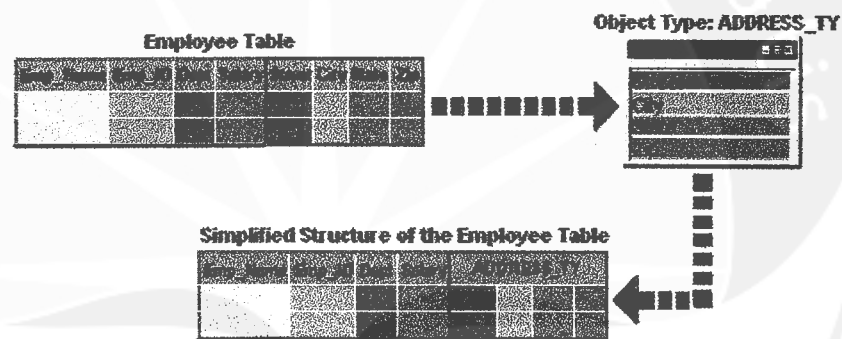
2.4.2. Oracle 9i ORDBMS

Oracle 9i telah mendukung fitur-fitur untuk mendukung konsep *Object Relational Database Management System*. Dalam menciptakan suatu basis data pada Oracle 9i *programmer* dapat menggunakan tipe data yang telah ada dan dapat menyimpan data dalam bentuk *relational table* seperti *DBMS* lainnya, namun *programmer* juga dapat menggunakan fitur *object oriented* dalam *ORDBMS* yaitu

dengan menciptakan *object type*, *object table*, *object view*, serta fitur-fitur lainnya.

2.4.2.1. Object Types

Object type merupakan tipe data bentukan *user* yang terdiri dari struktur data dan fungsi/*method* yang didefinisikan jika diperlukan perubahan atau modifikasi atribut di dalam *object type* tersebut. *Object type* di dalam ORDBMS hampir sama dengan *class* di dalam pemrograman berorientasi objek. Sama dengan *class*, *object type* juga mengurangi kompleksitas dengan memungkinkan *programmer* merepresentasikan data yang kompleks sebagai suatu entitas tunggal.



Gambar 2.8 Object Type di ORDBMS

Sebagai contoh, terdapat tabel *employee* dengan atribut seperti gambar 2.3. Atribut *street*, *city*, *state*, dan *zip* mewakili alamat dari seorang *employee*. Untuk menyederhanakan struktur *table employee* maka dapat *object type* *Address_Ty* yang mengandung atribut *street*, *city*, *state*, dan *zip*. Hal ini memungkinkan *programmer* untuk merepresentasikan alamat *employee* sebagai satu entitas. Variabel dalam *object type* dapat dimasukkan nilai yang

disebut sebagai *instance* dari *object type* yang juga merupakan suatu objek.

Object type terdiri dari dua bagian yaitu atribut dan metode. Atribut adalah komponen struktural yang dapat bertipe data yang telah ada atau tipe data bentukan. Metode adalah fungsi atau prosedur dalam *object type* yang memungkinkan aplikasi untuk menjalankan operasi terhadap atribut di dalam *object type* tersebut.

Keuntungan dari penggunaan *object type*, yaitu:

1. *Encapsulation*

Dalam *ORDBMS* fungsi untuk memanipulasi data tersimpan secara terpisah di dalam basis data. Dengan menggunakan *object type programmer* dapat membungkus atribut dengan metode untuk memanipulasi datanya sebagai satu kesatuan.

2. *Reusability*

Object type yang telah dibuat dapat digunakan berulang kali tanpa perlu didefinisikan ulang.

3. *Network Performance*

Object type memungkinkan *programmer* merepresentasikan data yang kompleks sebagai satu entitas. Dengan ini *programmer* dapat mengambil data yang saling berhubungan sebagai satu unit, caranya adalah dengan menggunakan *object references*. Hal ini menyebabkan koneksi ke basis data lebih cepat.

Tipe *method* dalam *object type* :

1. *Constructor Methods*

Peranannya hampir sama dengan konstruktor kelas pada pemrograman berorientasi obyek yakni pendefinisian atribut pada saat *instance* suatu objek.

2. Member Methods

Merupakan *method* yang didefinisikan oleh *user* untuk mengambil atau memanipulasi data atribut. Jenisnya ada tiga yaitu *Simple methods*, *Map methods*, dan *Order methods (comparing functions)*.

3. Static Methods

Method yang didefinisikan untuk *object type*, tidak dapat dipanggil oleh instance dari *object type* tersebut.

2.4.2.2. Nesting of Object Types

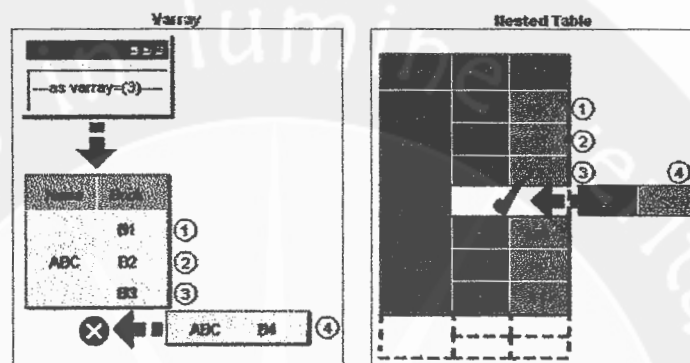
Object type dapat bersifat bersarang, yaitu di dalam *object type* terdapat *object type* yang lain. Sebagai contoh *programmer* akan menyimpan data *employee* di dalam *object type* *Emp_Ty*, dimana *employee* memiliki alamat yang disimpan dalam *object type* *Address_ty*. Maka *object type* *Address_Ty* digunakan di dalam *object type* *Emp_Ty*. *Object type* *Address_Ty* ini juga dapat digunakan sebagai nested type di dalam *object type* yang lain. Hal ini menunjukkan salah satu sifat *Object-Oriented Programming*, yaitu *reusable* dari suatu *object type*.

2.4.2.3. Object Table

Object table merupakan tabel yang dibentuk dari satu atau beberapa *object type* yang menyimpan data dalam bentuk *object*. Seorang *programmer* dapat membentuk *object table* dimana setiap baris diperlakukan sebagai satu *object* yang disebut *row object*. Dan apabila salah satu kolom di dalam *object table* menggunakan *object type* sebagai tipe datanya maka kolom tersebut disebut *column object*.

2.4.2.4. Collection Types

Di dalam Oracle 9i ORDBMS terdapat tipe *collection* yang memungkinkan penyimpanan banyak record data sebagai bagian dari satu baris tunggal di dalam suatu tabel atau *object type*. Hal ini mengurangi kebutuhan akan banyak tabel dan pendefinisian *constraint key*.



Gambar 2.9 Collection Type di ORDBSM

Tipe *collection* dalam Oracle 9i ORDBMS ada dua yaitu :

1. Varrays

Tipe *collection* yang digunakan apabila telah diketahui jumlah maksimum data *collection* yang dapat dimasukkan ke dalam satu baris data. Jumlah maksimum baris yang dapat dimasukkan harus didefinisikan pada saat pembuatan *varray* tersebut.

2. Nested Tables

Tabel bersarang, dimana di dalam tabel terdapat tabel yang direpresentasikan sebagai sebuah kolom. Dengan menggunakan *nested table*, maka jumlah maksimum baris data yang dapat dimasukkan sebagai *collection* tidak terbatas.

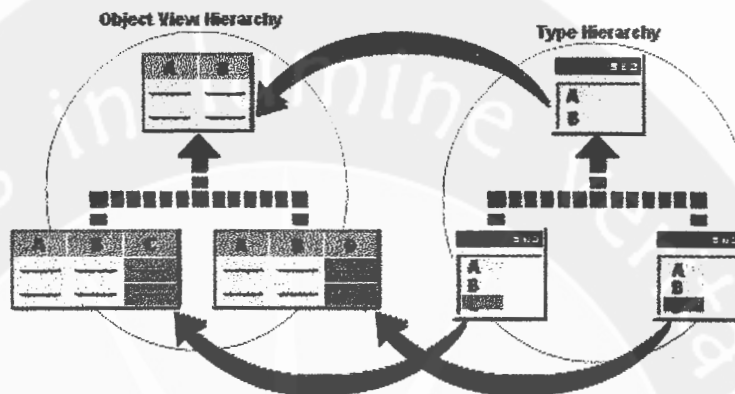
2.4.2.5. Object View

Untuk menggunakan data yang disimpan dalam bentuk *relational table* dalam bahasa pemrograman berorientasi objek, dapat diciptakan tipe data bentukan/*object type* dan *object table* yang dibuat berdasarkan tipe data tersebut. Kemudian dapat dikembangkan suatu aplikasi berorientasi objek berdasarkan objek-objek yang telah dibuat tersebut. Namun ada saat dimana seorang pengembang harus mengimplementasikan konsep *object oriented programming* dalam suatu aplikasi yang dikembangkan menggunakan *relational table*. Dalam kasus seperti ini dapat digunakan *object view* untuk mendefinisikan suatu objek yang menggunakan data pada *relational table* tanpa harus merubah aplikasi secara keseluruhan.

Object view dapat dikatakan sebuah *virtual object table* dimana tiap barisnya direpresentasikan sebagai sebuah objek. Data dari *object view* diambil dari *relational table* ataupun *object table*. Dengan menggunakan *object view*, programmer dapat mengambil data tertentu dari satu atau lebih *object table* yang saling berhubungan. Dengan menggunakan *object view* juga data-data tertentu dapat ditampilkan tanpa harus memperlihatkan semua data dalam suatu tabel secara keseluruhan, sebagai contoh dapat dibuat suatu *object view* yang memungkinkan user untuk melihat data spesifik tentang detail *employee* dan menyembunyikan data yang sifatnya rahasia seperti umur dan gaji *employee*. Menampilkan spesifik data dengan menggunakan *object view* juga meningkatkan *network performance* karena setiap baris diambil sebagai satu unit.

2.4.2.6. Object Hierarchy

Di dalam Oracle 9i ORDBMS terdapat fitur yang *object hierarchy* yang mewakili salah satu sifat *object oriented programming* yaitu *inheritance*. Dua fitur tersebut yaitu *object type hierarchy* dan *object view hierarchy*.



Gambar 2.10 Object View dan Object Type di ORDBMS

2.4.2.6.1. Object Type Hierarchy

Salah satu fitur teknologi *object relational* di Oracle 9i adalah *type hierarchy*. Hirarki tipe merupakan pohon keluarga dari kumpulan *object type*. Pohon ini mengandung *parent object type* yang disebut *supertype* dan satu atau lebih *child object types* yang disebut *subtypes*.

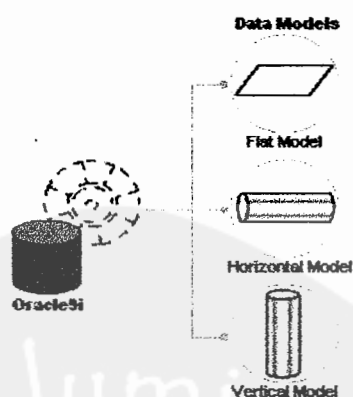
Suatu *type hierarchy* memungkinkan diciptakannya beberapa *object types* yang sifatnya diturunkan dari satu *object type*. Sebagai contoh dibuat *object type* `Employee_Ty` sebagai *supertype* dan *object type* `Government_Employee_Ty` dan `Private_Employee_Ty` sebagai *subtypes* dari `Employee_Ty`. Dalam suatu *type hierarchy*, *supertype* dan *subtypes* saling berhubungan satu sama lain dengan sifat *inheritance*. Hal ini berarti *subtypes* secara otomatis mewarisi semua atribut dan metode yang dimiliki oleh *supertype*.

Sebagai contoh jika *Employee_Ty* memiliki atribut *name* dan *address* serta metode *getName()*, maka *Private_Employee_Ty* sebagai subtype dari *Employee_Ty* juga memiliki atribut *name* dan *addrees* serta metode *getName()*. Semua perubahan terhadap atribut dan metode di *supertype* secara otomatis ikut mempengaruhi *subtypes*-nya.

2.4.2.6.2. Object View Hierarchy

Konsep dari *Object View Hierarchy* hampir sama dengan *object type hierarchy*, dimana *object type hierarchy* merupakan kumpulan dari *object view*. Tiap *object view* merepresentasikan *object type* yang berbeda-beda dalam *type hierarchy*. Dengan menggunakan *object view hierarchy*, dimungkinkan eksekusi *query* dari *object view* yang berbeda dengan spesialisasi dan level yang berbeda juga tingkatannya.

Object view hierarchy dapat dibuat berdasarkan setiap tingkatan *type hierarchy*. Basis dari *object view hierarchy* tidak harus root dari suatu *type hierarchy*. Di dalam *object view hierarchy* juga tidak diharuskan untuk memasukkan semua *subtypes* dari suatu *type hierarchy*. Ini berarti *object view hierarchy* juga bersifat dinamis layaknya *object view* ataupun *relational view*.



Gambar 2.11 Data Models Oracle 9i

Suatu object view hierarchy dapat dibentuk berdasarkan tiga model data dalam Oracle 9i. Tiga model data ini merepresentasikan cara berbeda yang masing-masingnya dapat digunakan untuk menciptakan object view hierarchy. Tiga data model tersebut yaitu:

1. *Flat Model*

Pada model ini satu tabel digunakan untuk membuat semua view yang digunakan dalam object view hierarchy.

Kelemahan dari model ini yaitu

- a. Satu tabel tunggal tidak dapat mengandung lebih dari 1000 kolom.
- b. Nilai null harus didefinisikan pada atribut tiap tabel tidak dari object type-nya.

2. *Horizontal Model*

Berbeda dari flat model, pada model ini jumlah tabel yang digunakan untuk membuat view dalam object view hierarchy adalah satu untuk masing-masing view.

Kelemahan dari model ini yaitu tidak dapat digunakannya perintah `SELECT * FROM` untuk mengakses semua data dari tabel.

3. Vertical Model

Hampir sama dengan *horizontal model*, pada model ini tiap *view* yang dibuat berdasarkan tabel yang berbeda. Namun tabel pada model ini hanya mengandung atribut yang sifatnya unik terhadap *subtype*-nya. Kelemahan dari model ini adalah selama proses *instance* dari suatu *object type*, satu *object identifier* harus didefinisikan untuk setiap *subtype* yang dipisahkan dari basis *object view hierarchy*.

