BAB II

TINJAUAN PUSTAKA

2.1 Sistem Informasi Restoran

2.1.1 Konsep Dasar Sistem Informasi

Definisi sistem informasi oleh Robert A. Leitch dan K. Roscoe Davis yang dikutip [Jog01] sebagai berikut:

> "Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan."

2.1.2 Konsep Dasar Sistem

Menurut [Jog01] terdapat dua kelompok pendekatan di dalam mendefinisikan sistem, yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya.

Definisi pendekatan sistem yang lebih menekankan pada prosedur menurut Jerry FitzGerald, Ardra F. FitzGerald, Warren D. Stallings, Jr yang dituliskan oleh [Jog01] "Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu." Pendekatan sistem yang merupakan jaringan kerja dari prosedur lebih menekankan urut-urutan operasi di dalam sistem. [Jog01] menuliskan definisi prosedur (*procedure*) didefinisikan oleh Richard F. Neuschel:

> "Suatu prosedur adalah suatu urut-urutan operasi klerikal (tulis-menulis), biasanya melibatkan beberapa orang di dalam satu atau lebih departemen,

> yang diterapkan untuk menjamin penanganan yang seragam dari transaksi-transaksi bisnis yang terjadi."

Lebih lanjut Jerry FitzGerald, Ardra F. FitzGerald dan Warren D. Stallings, Jr, mendefinisikan prosedur sebagai berikut:

> "Suatu prosedur adalah urut-urutan yang tepat dari tahapan-tahapan instruksi yang menerangkan apa (*what*) yang harus dikerjakan, siapa (*who*) yang mengerjakannya, kapan (*when*) dikerjakan dan bagaimana (*how*) mengerjakannya."

Pendekatan sistem yang lebih menekankan pada elemen atau komponennya mendefinisikan sistem sebagai berikut "Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu" [Jog01.]

Kedua kelompok definisi ini adalah benar dan tidak bertentangan, yang berbeda adalah cara pendekatannya.

2.1.3 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, yaitu mempunyai komponen-komponen (*components*), batas sistem (*boundary*), lingkungan luar sistem (*environments*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolah (*process*) dan sasaran (*objectives*) atau tujuan (*goal*) [Jog01].

2.1.3.1 Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem

2.1.3.2 Batas Sistem

Batas sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkup luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

2.1.3.3 Lingkungan Luar Sistem

Lingkungan luar (*environment*) dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup dari sistem.

2.1.3.4 Penghubung Sistem

Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Keluaran (*output*) dari satu subsistem akan menjadi masukan (*input*) untuk subsistem yang lainnya dengan melalui

penghubung. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

2.1.3.5 Masukan Sistem

Masukan (*input*) adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran. Sebagai contoh di dalam sistem komputer, program adalah *maintenance input* yang digunakan untuk mengoperasikan komputer dan data adalah *signal input* untuk diolah menjadi informasi.

2.1.3.6 Keluaran Sistem

Keluaran (*output*) adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supra sistem. Misalnya untuk sistem komputer, panas yang dihasilkan adalah keluaran yang tidak berguna dan merupakan sisa hasil pembuangan, sedang informasi adalah keluaran yang dibutuhkan.

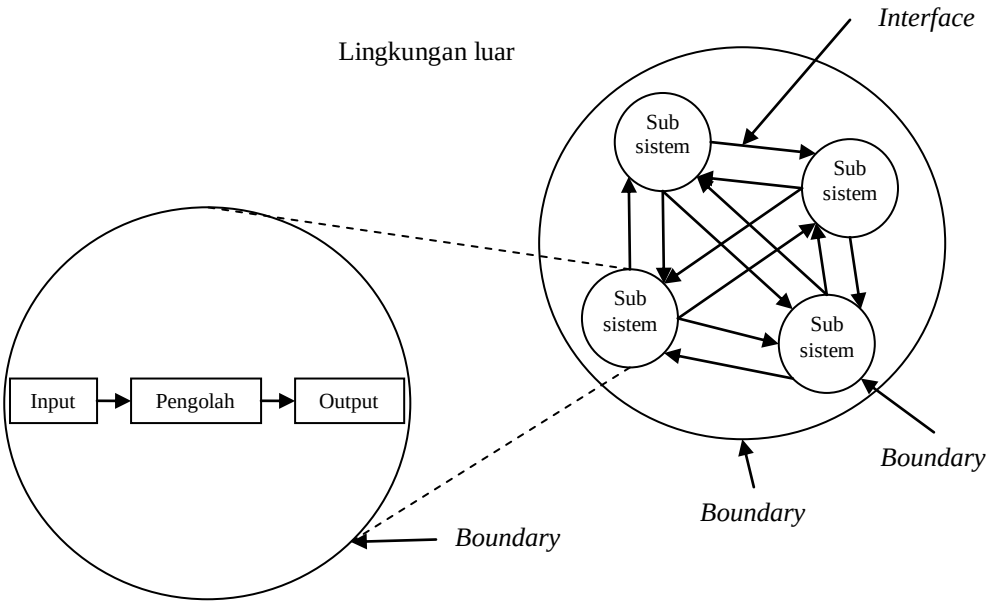
2.1.3.7 Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran. Suatu sistem produksi akan

mengolah masukan berupa bahan baku dan bahan-bahan yang lain menjadi keluaran berupa barang jadi.

2.1.3.8 Sasaran Sistem

Suatu sistem pastimempunyai tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya. Berikut adalah gambar karakteristik suatu sistem yang dogambarkan oleh [Jog01].



Gambar 2.1. Karakteristik Suatu Sistem

(Sumber: Analisis & Desain Sistem Informasi, Jogiyanto HM., 2001 Hal. 6)

2.1.4 Klasifikasi Sistem

Menurut [Jog01] Sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya adalah sebagai berikut ini:

1. Sistem diklasifikasikan sebagai sistem abstrak (*abstracy system*) dan sistem fisik (*physical system*).

   Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Sistem fisik merupakan sistem yang ada secara fisik.

2. Sistem diklasifikasikan sebagai sistem alamiah (*natural system*) dan sistem buatan manusia (*human made system*).

   Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Sistem buatan manusia adalah sistem yang dirancang oleh manusia.

3. Sistem diklasifikasikan sebagai sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*).

   Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur *probabilitas*.

4. Sistem diklasifikasikan sebagai sistem tertutup (*closed system*) dan sistem terbuka (*open system*).

   Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem terbuka adalah

sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya.

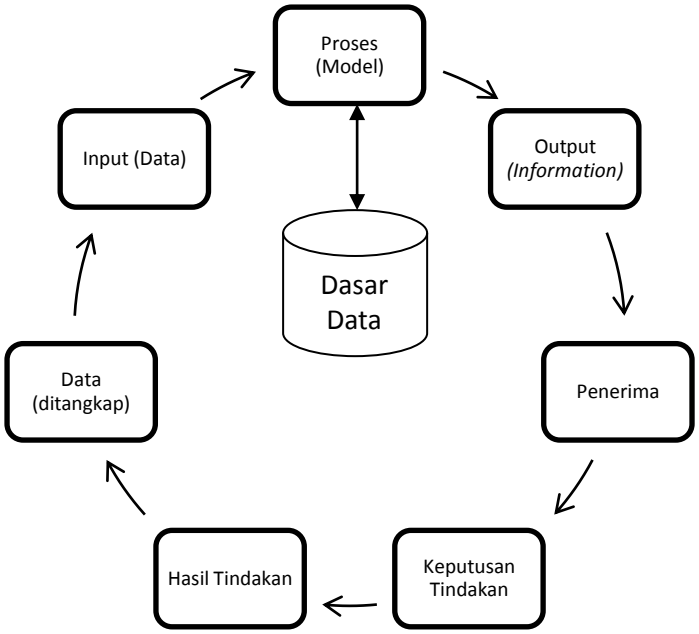
2.1.5 Konsep Dasar Informasi

Informasi menurut [Jog01] adalah "data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya." Sumber dari informasi adalah data. Data merupakan kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata.

2.1.6 Siklus Informasi

Data merupakan bentuk yang masih mentah yang belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut. Data diolah melalui suatu model untuk dihasilkan informasi [Jog01].

Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai input, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus. Siklus ini oleh John Burch yang dikutip oleh [Jog01] disebut dengan siklus informasi (*information cycle*). Siklus ini disebut juga dengan siklus pengolahan data (*data processing cycles*). Berikut adalah gambar siklus informasi yang digambarkan oleh [Jog01].

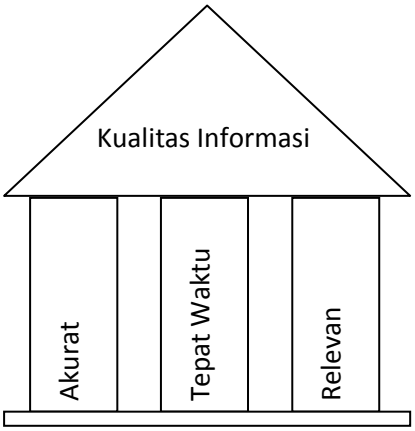


Gambar 2.2. Siklus Informasi

(Sumber: Analisis & Desain Sistem Informasi, Jogiyanto HM., 2001 Hal. 9)

2.1.7 Kualitas Informasi

Kualitas dari sebuah informasi (*quality of information*) menurut [Jog01] tergantung dari tiga hal, yaitu informasi harus akurat (*accurate*), tepat pada waktunya (*timeliness*) dan relevan (*relevance*). John Burch dan Gary Grudnitski yang dikutip oleh [Jog01] menggambarkan kualitas informasi dengan bentuk bangunan yang ditunjang oleh tiga buah pilar.



Gambar 2.3. Pilar Kualitas Informasi

(Sumber: Analisis & Desain Sistem Informasi, Jogiyanto HM., 2001 Hal. 10)

Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya.

Tepat pada waktunya, berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi.

Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda.

2.1.8 Nilai Informasi

Nilai dari informasi (*value of information*) [Jog01] ditentukan dari dua hal, yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya.

2.2 Pengertian Restoran

Restoran adalah suatu tempat atau bangunan yang diorganisasi secara komersial, yang menyelenggarakan pelayanan dengan baik kepada semua tamunya baik berupa makan maupun minum [Mwa05].

Tujuan operasi restoran adalah untuk mencari untung sebagaimana tercantum dalam definisi Prof. Vanco Christian dari *School Hotel Administration* yang dikutip oleh [Mwa05]. Selain bertujuan bisnis atau mencari untung, membuat puas para tamu pun merupakan tujuan operasi restoran yang utama.

2.3 Basisdata (*Database*)

Definisi basisdata menurut [Bha04] adalah kumpulan data (elementer) yang secara logik berkaitan dalam merepresentasikan fenomena/fakta secara terstruktur dalam domain tertentu untuk mendukung aplikasi pada sistem tertentu. Basisdata adalah kumpulan data yang saling berhubungan yang merefleksikan fakta-fakta yang terdapat di organisasi. Basisdata mendeskripsikan *state* organisasi/perusahaan/sistem. Saat satu kejadian muncul di dunia nyata mengubah *state* organisasi/perusahaan/sistem maka satu perubahan pun harus dilakukan terhadap data yang disimpan di basisdata. Basisdata merupakan komponen utama sistem informasi karena semua informasi untuk pengambilan keputusan berasal dari data di basisdata. Pengelolaan basisdata yang buruk dapat mengakibatkan ketidaktersediaan data penting yang digunakan untuk menghasilkan informasi yang diperlukan dalam pengambilan keputusan.

2.3.1 Sistem Manajemen Basisdata (*Database Management System*)

Sistem manajemen basisdata atau DBMS (*Database management system*) menurut [Bha04] adalah perangkat lunak untuk mendefinisikan, menciptakan, mengelola dan mengendalikan pengaksesan basisdata. Fungsi sistem manajemen basisdata saat ini yang paling penting adalah menyediakan basis untuk sistem informasi manajemen.

2.3.2 Tujuan DBMS

Tujuan utama DBMS [Bha04] adalah menyediakan lingkungan yang nyaman dan efisien untuk penyimpanan dan pengambilan data dari basisdata. DBMS berperan memberi abstraksi data tingkat tinggi ke pemakai. DBMS adalah perangkat lunak yang dirancang untuk membantu pengelolaan dan pemakaian koleksi data yang besar. Pemakai berurusan dengan abstraksi tingkat tinggi yang lebih akrab, nyaman dan fleksibelbukan rincian fisik penyimpanan data.

[Bha04] juga menuliskan tujuan lain DBMS yaitu:

1. Menghindari redundansi dan inkonsistensi data.
2. Menghindari kesulitan pengaksesan data.
3. Menghindari isolasi data.
4. Menghindari terjadinya anomali pengaksesan konkuren.
5. Menghindari masalah-masalah keamanan.
6. Menghindari masalah-masalah integritas.

Pada kebanyakan aplikasi, basisdata di bawah kendali DBMS yang disediakan *vendor* spesialis DBMS. Saat aplikasi hendak melakukan operasi pada

basisdata. Aplikasi memberi/mengirim pesan permintaan (*request*) ke perangkat lunak DBMS. Operasi-operasi (dasar) yang biasa dilakukan pemakai terhadap basisdata (diasumsikan pada basisdata relasional) hanya empat:

1. Menambah informasi, pada basisdata relasional dilakukan dengan menambah baris di tabel (dapat dipandang sebagai operasi C - *Create*).
2. Mengekstrak informasi, pada basisdata relasional dilakukan dengan mengekstrak baris-baris yang berasal dari satu atau beberapa tabel (dapat dipandang sebagai operasi R - *Read*).
3. Memodifikasi data yang tersimpan, pada basisdata relasional dilakukan dengan memodifikasi satu atau beberapa baris di tabel (dapat dipandang sebagai operasi U - *Update*).
4. Menghapus data tertentu, pada basisdata relasional dilakukan dengan menghapus baris tertentu di tabel (dapat dipandang sebagai operasi D - *Delete*).

Untuk mempermudah penghapalan operasi-operasi dasar pada basisdata, kumpulan operasi terhadap basis data dapat disingkat dengan CRUD (*Create Read Update Delete*). Jadi tugas pokok DBMS yang utama adalah mengurusi empat operasi dasar itu. Agar operasi CRUD berjalan secara cepat dan lancar pada basisdata bervolume besar maka basisdata perlu dirancang memenuhi kriteria-kriteria basisdata yang bagus seperti tidak mempunyai redundansi sehingga tidak ada anomali pembaruan yang dapat menyesatkan.

2.3.3 Bentuk Normal

Tingkat redundansi yang terjadi dinyatakan sebagai bentuk normal (*norm form*). Relasi secara berurutan diproses menjadi bentuk normal lebih tinggi yang memiliki tingkat redundansi lebih rendah. Normalisasi adalah pemrosesan relasi-relasi menjadi bentuk normal lebih tinggi. Dengan demikian, tujuan proses normalisasi adalah mengkonversi relasi menjadi bentuk normal yan lebih tinggi [Bha04].

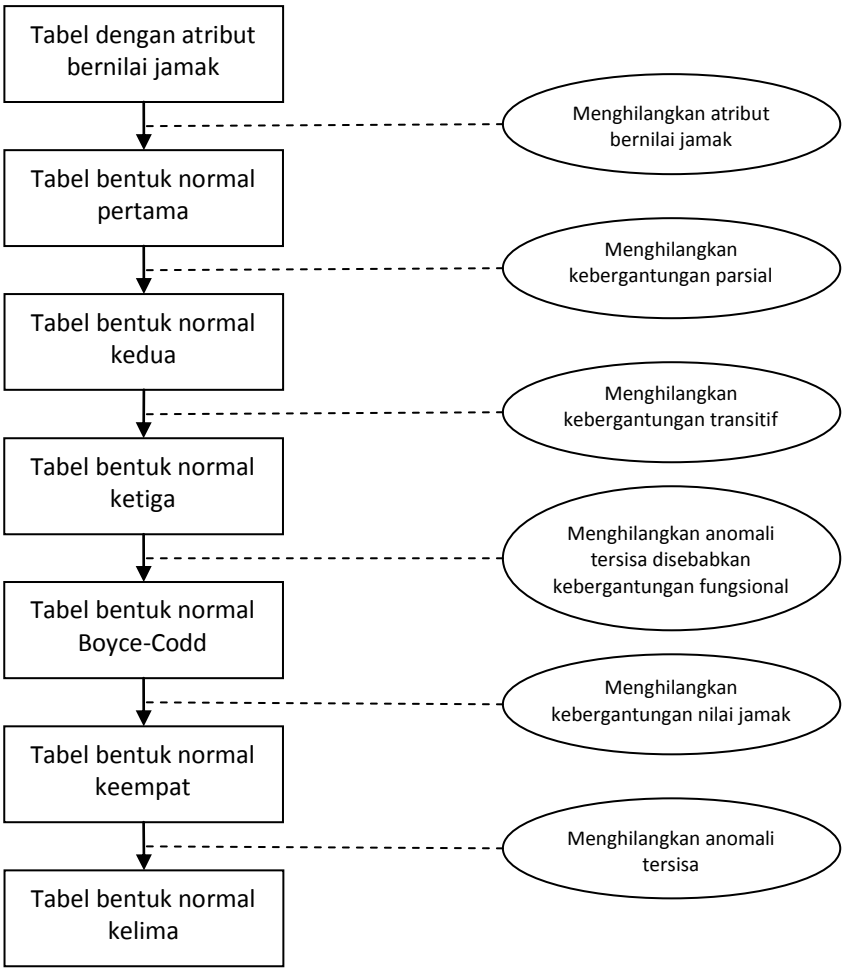
[Bha04] juga menuliskan beragam tingkat berntuk normal, yaitu:

1. Bentuk normal pertama (1NF).
2. Bentuk normal kedua (2NF).
3. Bentuk normal ketiga (3NF).
4. Bentuk normal Boyce-Codd (BCNF).
5. Bentuk normal keempat (4NF).
6. Bentuk normal kelima (5NF).

[Bha04] menuliskan bahwa Codd mendifinisikan bentuk normal pertama, kedua dan ketiga di makalah (Codd, 1970). Bentuk normal ketiga kemudian diperbaiki sehingga mempunyai bentuk normal yang lebih kuat yaitu BCNF (Codd, 1974). Fagin memperkenalkan bentuk normal keempat (Fagin, 1977) kemudian Fagin juga memperkenalkan bentuk normal kelima (Fagin, 1979).

[Bha04] menuliskan kriteria dalam proses normalisasi adalah kebergantungan fungsional (*functional dependency*), kebergantungan banyak nilai dan kebergantungan *join* (*join dependency*). Proses normalisasi membentuk relasi-relasi bentuk normal menggunakan dekomposisi yang memecah relasi menjadi relasi-relasi berbentuk normal lebih tinggi. Namun kita belum tentu perlu bentuk

normal tertinggi, BCNF adalah telah memadai pada banyak aplikasi. Berikut adalah gambar langkah-langkah normalisasi yang diambil dari [Bha04].



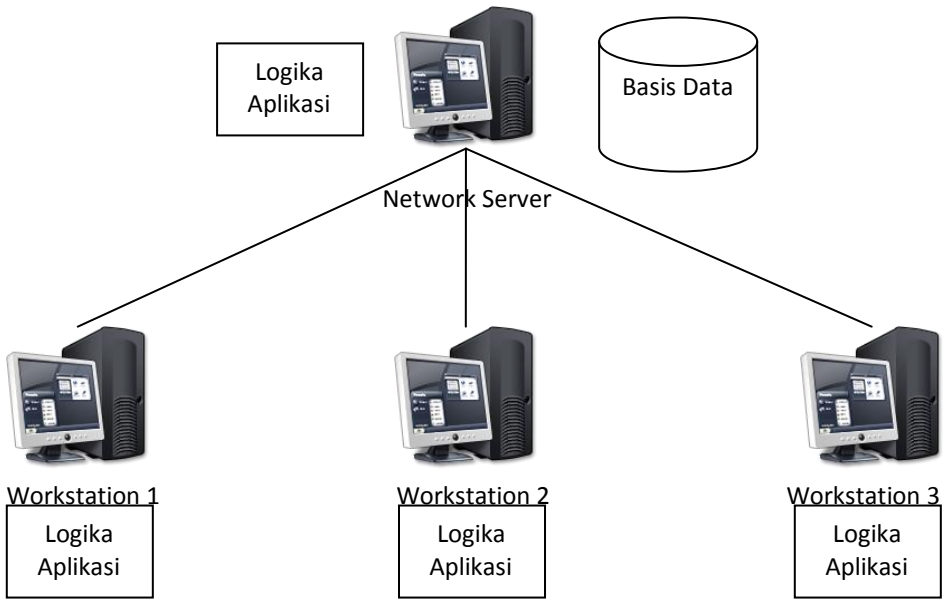
Gambar 2.4. Langkah-langkah Normalisasi

(Sumber: Sistem Manajemen BasisData, Bambang H., Ir., MT, 2001 Hal. 6)

2.4 Arsitektur *Client-Server*

Arsitektur *Client-Server* [Anu02] adalah suatu cara untuk meningkatkan kinerja konfigurasi *file-server* yang menurun karena faktor skalabilitas (penambahan *workstation* dalam jumlah yang signifikan). Pada arsitektur ini, 2 aplikasi yang terpisah, beroperasi secara mandiri dan bekerjasama untuk menyelesaikan suatu pekerjaan. Suatu aplikasi yang cocok dengan arsitektur ini

adalah DBMS (*Database Management System)* berbasis SQL (*Structured Query Language)*.



Gambar 2.5. Arsitektur *Client-Server*

(Sumber: Analisis dan Perancangan Sistem Informasi, Adi Nugroho, ST., MMSI., 2002 Hal. 158)

Konfigurasi *client-server* mencakup 2 entitas: *client* dan *server*. *Client* meminta sesuatu pada *server* kemudian server melakukan sesuatu pekerjaan yang diminta oleh *client*. Permintaan dapat berupa *query* SQL yang dikirimkan pada mesin basis data SQL. Kemudian, mesin basis data SQL memroses *query* dan hasilnya (berupa *resultset*) dikembalikan pada *client* yang meminta.

Sistem *client-server* memungkinkan satu aplikasi tunggal dibagi-bagi antar *workstation* maupun *server*. Dari sini muncullah istilah/konsep *deployment*. Para penegmbang sistem saat ini harus mulai memikirkan konsep *deployment* (pemecahan aplikasi) demi peningkatan kinerja aplikasi yang dia kembangkan.

2.5 *Object Oriented Programming* (OOP)

OOP (*Object Oriented Programming)* atau Pemrograman Berorientasi Objek [Anu02] adalah suatu cara baru dalam berpikir serta berlogika dalam menghadapi masalah-masalah yang akan dicoba atasi dengan bantuan komputer. OOP, tidak seperti pendahulunya, (pemrograman terstruktur), mencoba melihat permasalahannya lewat pengamatan dunia nyata dimana setiap objek adalah entitas tunggal yang memiliki kombinasi struktur data dan fungsi tertentu. Ini kontras dengan pemrograman terstruktur dimana struktur data dan fungsi didefinisikan secara terpisah dan tidak berhubungan erat.

Filosofi OOP menciptakan sinergi yang luar biasa sepanjang siklus pengembangan perangkat lunak (perencanaan, analisis, perancangan, serta implementasi) sehingga dapat diterapkan pada perancangan sistem secara umum; menyangkut perangkat lunak, perangkat keras, serta sistem informasi secara keseluruhan.

2.6 Visual Studio .Net 2005

Visual Studio .Net 2005 [Ham05] merupakan salah satu produk pengembangan aplikasi yang diproduksi oleh Microsoft. Visual Studio .Net 2005 dapat digunakan untuk pengembangan aplikasi web ASP .NET, XML Web Service, aplikasi desktop dan juga aplikasi mobile. Dalam Visual Studio .Net 2005 terdapat beberapa *tool* yang dapat dipilih untuk pengembangan aplikasi. Tool-tool tersebut, antara lain Visual Basic, Visual C#, Visual C++ dan Visual J#. Tool-tool pada Visual Studio .Net 2005 tersebut menggunakan IDE (*Integrated*

Development Environment) yang sama sehingga dapat saling berbagi pakai fasilitas dalam pengembangan aplikasi.

2.7 Microsoft SQL Server 2005

Microsoft SQL Server 2005 [Wbu06] ialah perangkat lunak *Relational Database Management System* (RDBMS) yang andal. Didesain untuk mendukung proses transaksi yang besar (seperti *order entry* yang *on line*, inventory, akuntansi, atau manufaktur). SQL server 2005 memiliki fasilitas tambahan yang menyebabkannya memiliki kemampuan penuh dalam *e-Commerce*, antara lain *reporting* dan *analysis services*.

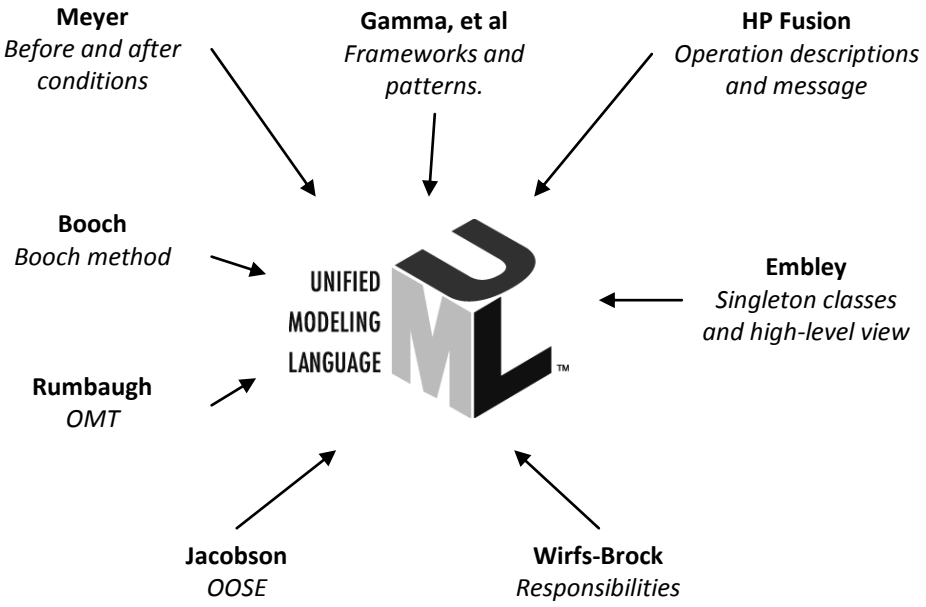
2.8 UML (*Unified Modelling Language*)

UML [Mun05] adalah salah satu alat bantu di dunia pengembangan sistem yang berorientasi obyek. UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru (*blueprint*) atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, OMT (*Object Modeling Technique*) dan OOSE (*Object Oriented Software Engineering*) [Mun05]. Metode Booch dari Grady Booch sangat terkenal dengan nama metode *Design Object Oriented*. Metode ini menjadikan proses analisis dan desain ke dalam empat tahapan iteratif, yaitu: identifikasi kelas-kelas dan obyek-obyek, identifikasi semantik dari hubungan obyek dan kelas tersebut,

perincian interface dan implementasi. Keunggulan metode Booch adalah pada detil dan kayanya dengan notasi dan elemen. Pemodelan OMT yang dikembangkan oleh Rumbaugh didasarkan pada analisis terstruktur dan pemodelan *entity-relationship*. Tahapan utama dalam metodologi ini adalah analisis, desain sistem, desain obyek dan implementasi. Keunggulan metode ini adalah dalam penotasian yang mendukung semua konsep OO (*Object Oriented*). Metode OOSE dari Jacobson lebih memberi penekanan pada *use case*. OOSE memiliki tiga tahapan yaitu membuat model requirement dan analisis, desain dan implementasi, dan model pengujian (*test model*). Keunggulan metode ini adalah mudah dipelajari karena memiliki notasi yang sederhana namun mencakup seluruh tahapan dalam rekayasa perangkat lunak.

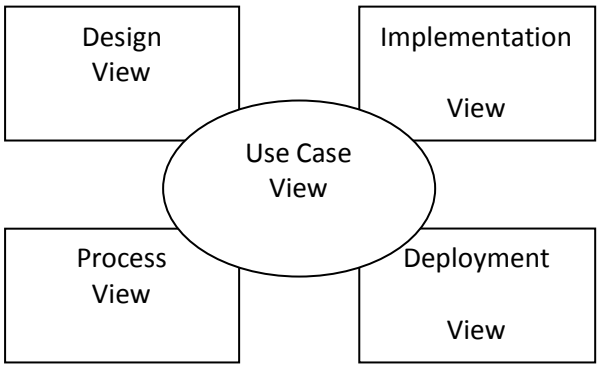
Dengan UML, metode Booch, OMT dan OOSE digabungkan dengan membuang elemen-elemen yang tidak praktis ditambah dengan elemen-elemen dari metode lain yang lebih efektif dan elemen-elemen baru yang belum ada pada metode terdahulu sehingga UML lebih ekspresif dan seragam daripada metode lainnya. Gambar berikut adalah unsur-unsur yang membentuk UML.



Gambar 2.6. Unsur-unsur Pembentuk UML.

(Sumber: Pemodelan Visual dengan UML, Munawar, 2005. Hal. 18)

UML dibangun atas model 4+1 view. Model ini didasarkan pada fakta bahwa struktur sebuah sistem dideskripsikan dalam lima view dimana salah satu diantaranya use case view. Use case view ini memegang peran khusus untuk mengintegrasikan content ke view yang lain.



Gambar 2.7. Model 4+1 View

(Sumber: Pemodelan Visual dengan UML, Munawar, 2005. Hal. 20)

Kelima view tersebut tidak berhubungan dengan diagram yang dideskripsikan di UML. Setiap view berhubungan dengan perspektif tertentu dimana sistem akan diuji. View yang berbeda akan menekankan pada aspek yang berbeda dari sistem yang mewakili ketertarikan sekelompok stakeholder tertentu. Penjelasan lengkap tentang sistem bisa dibentuk dengan menggabungkan informasi-informasi yang ada pada kelima view tersebut.

Use Case View mendefinisikan perilaku eksternal sistem. Hal ini menjadi daya tarik bagi *end user*, analisis dan tester. Pandangan ini mendefinisikan kebutuhan sistem karena mengandung semua view yang lain yang mendeskripsikan aspek-aspek tertentu dari rancangan sistem. Itulah sebabnya use case view menjadi pusat peran dan sering dikatakan yang mendrive proses pengembangan perangkat lunak.

Design View mendeskripsikan struktur logika yang mendukung fungsi-fungsi yang dibutuhkan di use case. Design view iniberisi definisi komponen program, class-class utama bersama-sama dengan spesifikasi data, perilaku dan interaksinya. Informasi yang terkandung di view inimenjadi perhatian para programmer karena menjelaskan secara detil bagaimana fungsionalitas sistem akan diimplementasikan.

Implementation View menjelaskan komponen-komponen fisik dari sistem yang akan dibangun. Hal ini berbeda dengan komponen logic yang dideskripsikan pada design view. Termasuk di sini diantaranya file exe, library dan database. Informasi yang ada di view ini relevan dengan aktifitas-aktifitas seperti manajemen konfigurasi dan integrasi sistem.

Process View berhubungan dengan hal-hal yang berkaitan dengan *concurrency* di dalam sistem.

Deployment View menjelaskan bagaimana komponen-komponen fisik didistribusikan ke lingkungan fisik seperti jaringan komputer dimana sistem akan dijalankan. Kedua view ini menunjukkan kebutuhan non fungsional dari sistem seperti toleransi kesalahan dan hal-hal yang berhubungan dengan kinerja.

2.9 Sistem Operasi Windows

Microsoft Windows atau lebih dikenal dengan sebutan Windows adalah keluarga sistem operasi komputer pribadi yang dikembangkan oleh Microsoft yang menggunakan antarmuka dengan pengguna berbasis grafik (*graphical user interface*).

Sistem operasi Windows telah berevolusi dari MS-DOS, sebuah sistem operasi yang berbasis modus teks dan command-line. Windows versi pertama, Windows Graphic Environment 1.0 pertama kali diperkenalkan pada 10 November 1983, tetapi baru keluar pasar pada bulan November tahun 1985 yang dibuat untuk memenuhi kebutuhan komputer dengan tampilan bergambar. Windows 1.0 merupakan perangkat lunak 16-bit tambahan (bukan merupakan sistem operasi) yang berjalan di atas MS-DOS (dan beberapa varian dari MS-DOS), sehingga ia tidak akan dapat berjalan tanpa adanya sistem operasi DOS. Versi 2.x, versi 3.x juga sama. Beberapa versi terakhir dari Windows (dimulai dari versi 4.0 dan Windows NT 3.1) merupakan sistem operasi mandiri yang tidak lagi bergantung kepada sistem operasi MS-DOS. Microsoft Windows kemudian bisa

berkembang dan dapat menguasai penggunaan sistem operasi hingga mencapai 90% [http://id.wikipedia.org/wiki/Windows].

2.10 Metode Pengembangan Sistem

Pengembangan sistem *(system development)* dapat berarti menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah ada [Jog01].

Metodologi adalah kesatuan metode-metode, prosedur-prosedur, konsep-konsep pekerjaan, aturan-aturan dan postulat-postulat yang digunakan oleh suatu ilmu pengetahuan, seni atau disiplin lainnya.

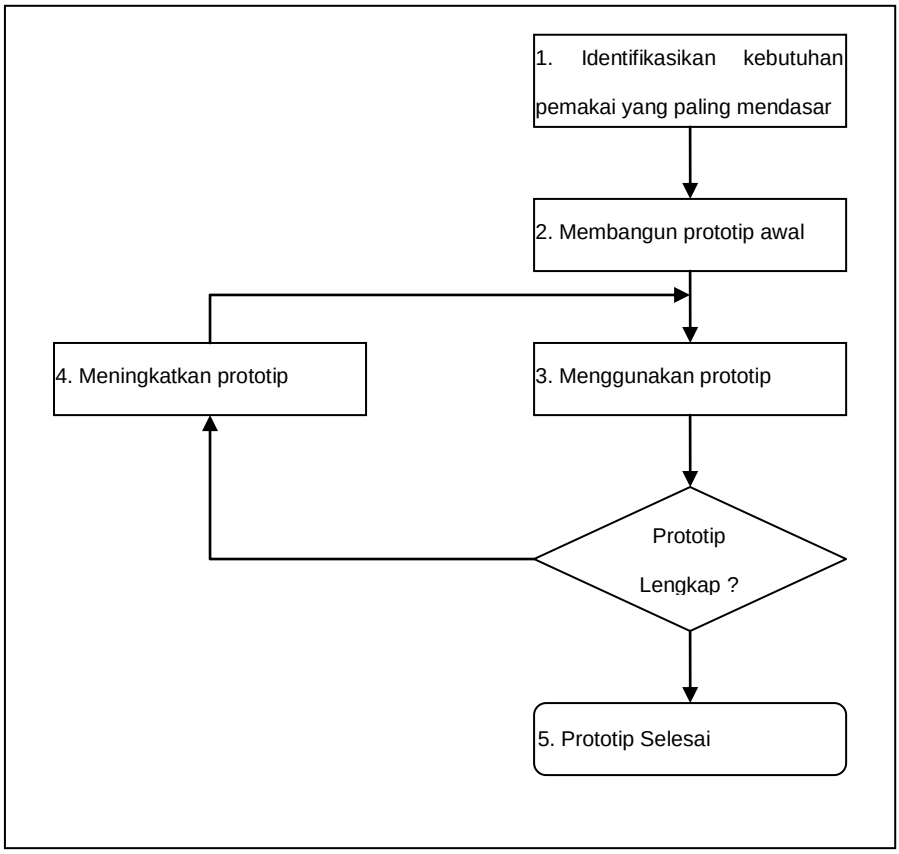
Metodologi pengembangan sistem adalah metode-metode, prosedur-prosedur, konsep-konsep pekerjaan, aturan-aturan dan postulat-postulat yang akan digunakan untuk mengembangkan suatu sistem informasi. Penulis menggunakan *prototyping* sebagai metodologi pengembangan sistem.

Suatu prototip *(prototype)* [Jog03] adalah bentuk dasar atau model awal dari suatu sistem atau bagian dari suatu sistem. Setelah dioperasikan, prototip ditingkatkan terus sesuai dengan kebutuhan pemakai sistem yang juga meningkat.

Prototyping adalah proses pengembangan suatu prototip secara cepat untuk digunakan terlebih dahulu dan ditingkatkan terus-menerus sampai didapatkan sistem yang utuh. Proses membangun sistem ini yaitu dengan membuat prototip atau model awal, mencobanya, meningkatkannya dan mencobanya lagi dan meningkatkannya dan seterusnya sampai didapatkan sistem yang lengkap. Proses ini disebut proses iteratif dari pengembangan sistem.

Tahapan-tahapan yang dilakukan didalam pengembangan sistem menggunakan model prototip adalah sebagai berikut.

1. Identifikasi kebutuhan pemakai paling mendasar. Pembuat sistem dapat mewawancarai pemakai sistem tentang kebutuhan pemakai sistem yang paling minimal terlebih dahulu.
2. Membangun prototip. Prototip dibangun oleh pembuat sistem dengan cepat. Hal ini dimungkinkan karena pembuat sistem hanya membangun bagian yang paling mendasar dulu dari keseluruhan sistem yang paling dibutuhkan terlebih dahulu oleh pemakai sistem. Hal lainnya yang memungkinkan pembuat sistem membangun prototip dengan cepat adalah dengan menggunakan alat-alat bantu generasi terbaru seperti misalnya DBMS dan CASE.
3. Menggunakan prototip. Pemakai sistem dianjurkan untuk menggunakan prototip sehingga dapat menilai kekurangan-kekurangan dari prototip sehingga dapat memberikan masukan-masukan kepada pembuat sistem.
4. Merevisi dan meningkatkan prototip. Pembuat sistem memperbaiki prototip berdasarkan keinginan dari pemakai sistem atau berdasarkan pengalamannya untuk membuat sistem sejenis yang baik. Jika prototip belum lengkap, maka proses iterasi diulang lagi mulai dari langkah 3.
5. Jika prototip lengkap menjadi sistem yang dikehendaki, proses iterasi dihentikan.



Gambar 2.8. Membangun Prototip

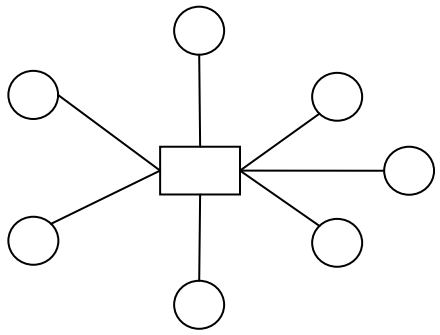
(Sumber: Sistem Teknologi Informasi, Jogiyanto HM., 2003. Hal. 527)

2.11 Topologi Jaringan

Jaringan (*Network*) [Jog03] adalah jaringan dari sistem komunikasi data yang melibatkan sebuah atau lebih *node* (sumber-sumber daya) yang dihubungkan dengan jalur transmisi (*link*) membentuk suatu sistem. Istilah *node* menunjukkan sumber-sumber daya seperti terminal, komputer, printer dan lain sebagainya. Sedang istilah *link* menunjukkan media penghubungnya, misalnya kabel, *microwave* atau satelit. Bentuk dari jaringan disebut dengan topologi.

2.11.1. Topologi Bintang (Star Network)

Topologi bintang menghubungkan suatu *node* pusat (*central node* atau *host node*) yang membentuk jaringan seperti bentuk bintang (*star*). Semua komunikasi ditangani dan diatur langsung oleh *central node*. *Central Node* melakukan semua tanggung jawab untuk mengatur arus informasi di antara *node* yang lainnya. Jika *node* yang satu ingin berkomunikasi dengan *node* yang lainnya, maka harus melewati *central node*. *Central node* biasanya berupa komputer besar (*large computer*) atau *mainframe computer* yang dihubungkan dengan *node* yang lainnya yang berupa beberapa terminal atau komputer mini atau komputer mikro melalui suatu link.



Gambar 2.9. Topologi Bintang

(Sumber: Sistem Teknologi Informasi, Jogiyanto HM., 2003. Hal. 197)