



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

TELAAH LITERATUR

2.1 Rumah Sakit

Rumah Sakit adalah salah satu sarana kesehatan tempat menyelenggarakan kesehatan dengan memberdayakan berbagai kesatuan personel terlatih dan terdidik dalam menghadapi dan menangani masalah medis untuk pemulihan dan pemeliharaan kesehatan yang baik.

Upaya kesehatan adalah setiap kegiatan untuk memelihara dan meningkatkan kesehatan yang bertujuan untuk mewujudkan derajat kesehatan yang optimal bagi masyarakat dan tempat yang digunakan untuk menyelenggarakannya disebut sarana kesehatan. Sarana kesehatan berfungsi melakukan upaya kesehatan dasar, kesehatan rujukan dan atau upaya kesehatan penunjang. Upaya kesehatan diselenggarakan dengan pendekatan pemeliharaan, peningkatan kesehatan (promotif), pencegahan penyakit (preventif), penyembuhan penyakit (kuratif) dan pemulihan kesehatan (rehabilitatif) yang diselenggarakan secara menyeluruh, terpadu dan berkesinambungan (Siregar, 2004).

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

2.2 Sistem Informasi

Suatu instansi baik perusahaan kecil maupun instansi besar biasanya memiliki sebuah sistem operasional dalam kegiatan organisasinya. Sistem yang dimiliki antara satu instansi dengan instansi lainnya biasanya memiliki perbedaan, hal tersebut disebabkan oleh berbagai faktor seperti perbedaan bidang bisnis, kebijakan instansi dan banyak hal lainnya. Dalam bukunya O'Brien dan Marakas (2010), mendefinisikan sistem sebagai sekumpulan komponen dengan batasan-batasan yang telah ditentukan, dan komponen-komponen tersebut bekerja sama untuk mencapai suatu tujuan. Kemudian Kendall dan Kendall (2011), menambahkan bahwa komponen-komponen tersebut saling tergantung antara satu dengan yang lainnya, dan menyebutkan bahwa sebuah sistem terdiri dari masukan, proses dan keluaran. Sedangkan Shelly dan Rosenblatt (2012), menambahkan bahwa sebuah sistem membutuhkan masukan data.

Informasi merupakan suatu hal yang sangat penting bagi perusahaan, karena setiap keputusan yang dibuat untuk keperluan strategis perusahaan didapat dari informasi-informasi yang ada. Menurut O'Brien dan Marakas (2010), informasi merupakan data yang telah diubah menjadi sesuatu yang memiliki arti dan berguna bagi pengguna. Definisi tersebut diperkuat oleh Shelly dan Rosenblatt (2012). Sedangkan Rainer, Turban dan Potter (2006) menyebutnya sebagai data yang telah diorganisir.

Sistem Informasi merupakan kombinasi perangkat keras (*Hardware*), piranti lunak (*Software*) dan Jasa (*Service*) yang digunakan oleh orang (*People*) untuk mengatur, mengkomunikasikan dan membagi informasi (Shelly & Rosenblatt, 2012), dan ditambahkan bahwa sistem informasi juga terdiri dari jaringan komunikasi (*Computer Network and Data Communication*) dan basis data (*database*) yang dipergunakan untuk mengumpulkan, mengubah dan menyebarkan informasi dalam sebuah organisasi (O'Brien & Marakas, 2008).

Sesuai dengan definisi sistem informasi di atas, komponen-komponen sistem informasi adalah sebagai berikut:

- Perangkat Keras

Perangkat keras terdiri dari lapisan fisik sistem informasi. Sebagai contoh, perangkat keras dapat meliputi *server*, *workstation*, jaringan, peralatan telekomunikasi, kabel fiber optik, perangkat *mobile*.

- Perangkat Lunak

Perangkat lunak merupakan program yang mengontrol perangkat keras dan menghasilkan informasi dari hasil yang diharapkan.

- Data

Data merupakan material mentah yang diubah sistem informasi menjadi suatu informasi yang berguna. Data tersebut disimpan di sebuah lokasi yang disebut dengan tabel dan dari mempergunakan tabel tersebut, informasi dapat dihasilkan.

- Proses

Untuk dapat membangun sistem informasi yang baik, seorang analis harus memahami proses bisnis dari dokumen-dokumen yang ada. Sebuah proses merupakan tugas dan fungsi-fungsi bisnis yang dilakukan oleh orang.

- Orang

Orang merupakan entitas yang berinteraksi dengan sistem. Contohnya adalah *staff* IT, penganalisa sistem, pembuat program, dan admin jaringan merupakan entitas yang membangun dan mendukung sistem, sedangkan pengguna (*user*) merupakan entitas yang nantinya akan menggunakan sistem tersebut.

2.3 Sistem Informasi Rumah Sakit

Sistem Informasi Rumah Sakit adalah suatu tatanan yang berurusan dengan pengumpulan data, pengelolaan data, penyajian informasi, analisa dan penyimpulan informasi serta penyampaian informasi yang dibutuhkan untuk kegiatan rumah sakit.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

2.3.1 Sistem Informasi Rumah Sakit Subsistem Laboratorium

Pelayanan laboratorium merupakan salah satu pelayanan penunjang di lingkungan rumah sakit dalam menjalankan fungsinya untuk memberikan pelayanan kepada masyarakat.

Divisi laboratorium bertanggung jawab terhadap pemberian jasa pelayanan yang berhubungan dengan pemeriksaan penunjang medis laboratorium baik pemeriksaan medis, patologi, dan mikrobiologi bagi seorang pasien. Selanjutnya saat ini divisi laboratorium juga diberikan tanggung jawab sebagai pengelola Bank Darah.

Kegiatan pemeriksaan patologi klinik terdiri dari:

- a. Pemeriksaan Hematologi (antara lain: Sitologi sel darah, sitokimia darah, analisa Hemoglobin (Hb) , Perbankan darah, Hemostasis dan Kelompok pemeriksaan lain).
- b. Pemeriksaan Kimia klinik (antara lain: protein, karbohidrat, lipid, lipoprotein, apoprotein, enzim, mikronutrient dan monitoring kadar terapi obat, elektrolit, fungsi organ, hormon dan fungsi endokrin serta kelompok pemeriksaan lain) 2 sampai dengan 2.9.14.

Untuk pemeriksaan laboratorium patologi klinik meliputi semua kegiatan pemeriksaan laboratorium patologi klinik di rumah sakit yang total kegiatannya akan terlihat pada lajur 99. Pengisian formulir pemeriksaan laboratorium harus sesuai antara jumlah di setiap poin dengan jumlah sub poin dan jumlah sub sub poin.

2.4 Analisis dan Perancangan Sistem

Untuk mengetahui dan menilai apakah sistem informasi di dalam suatu instansi mengalami masalah atau tidak, perlu dilakukan analisis terhadap sistem informasi tersebut. Jika ditemukan masalah maka harus dicari solusi yang terbaik untuk mengatasi masalah tersebut dimana diperlukan perancangan sistem informasi yang baru. Oleh karena itu, sebelum melakukan analisis dan perancangan sistem informasi terlebih dahulu harus dipahami konsep dasar dari analisis dan perancangan sistem informasi tersebut.

Analisis sistem sebagai proses penyelidikan terhadap kebutuhan informasi pemakai di dalam suatu organisasi agar dapat menetapkan tujuan dan spesifikasi desain suatu sistem dan ditambahkan bahwa analisis sistem dilakukan kepada sistem yang telah ada dengan tujuan merancang dan memperbaharui sistem tersebut (McLeod & Schell, 2006). Whitten & Bentley (2007) juga menyebutkan bahwa analisis sistem dilakukan dengan mempelajari area masalah yang ada pada proses bisnis.

Sedangkan perancangan sistem didefinisikan oleh McLeod (2001) sebagai penentuan proses dan data yang nantinya akan diperlukan oleh sistem yang baru dan pendapat ini ditambahkan bahwa perancangan sistem merupakan spesifikasi solusi berbasis komputer yang sesuai dengan kebutuhan bisnis yang dilakukan setelah tahap analisis sistem (Whitten & Bentley, 2007).

2.4.1 *Rich Picture*

Rich Picture adalah penggambaran sistem atau situasi dengan menggunakan gambar-gambar. Gambaran keseluruhan dari orang, objek, proses, struktur, dan masalah pada keseluruhan proses bisnis yang ada di perusahaan.

Rich Picture digunakan untuk menggambarkan keseluruhan proses bisnis secara jelas dengan gambar dan hubungan antar gambar tersebut dengan penjelasan singkat agar orang yang melihat dapat dengan mudah untuk mengerti dan memahami maksud dari gambar tersebut.

2.4.2 *Data Flow Diagram (DFD)*

Data Flow Diagram (DFD) adalah suatu alat yang menggambarkan aliran data yang ada dalam sistem dan suatu proses yang dilakukan oleh suatu sistem. Sering juga disebut sebagai *buble chart*, *transformation graph*, dan model proses. DFD dapat digunakan untuk merepresentasikan suatu sistem yang otomatis maupun manual dengan melalui gambar yang berbentuk jaringan grafik.

DFD ini adalah salah satu alat pembuatan model yang sering digunakan khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem. DFD ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem

kepada pemakai maupun pembuat program (Burch, 1992). Berikut adalah istilah-istilah yang ada dalam DFD.

Entitas Eksternal

- Entitas yang berada **di luar sistem**, yang **memberikan** data kepada sistem disebut source atau yang **menerima** informasi dari sistem disebut sink.
- **Tidak termasuk** bagian dari sistem.
- Bila sistem informasi dirancang untuk satu bagian (departemen), maka bagian lain yang masih terkait menjadi entitas eksternal.

Proses

- Menggambarkan apa yang dilakukan oleh sistem.
- Berfungsi mentransformasikan satu atau beberapa data masukan menjadi satu atau beberapa data keluaran sesuai dengan spesifikasi yang diinginkan.
- Setiap proses memiliki satu atau beberapa data masukan serta menghasilkan satu atau beberapa data keluaran.

Data Store dan Data Flow

Data Store:

- Tempat menyimpan data.
- Proses dapat mengambil data dari atau memberikan data ke *data store*.

Data Flow

- Menggambarkan aliran data dari suatu *entity* ke *entity* lainnya.
- Aliran data antara dua proses yang berurutan.
- Aliran data dari *data store* ke process dan sebaliknya
- Aliran data dari entitas ke proses atau sebaliknya.

Gane/Sarson	Yourdon/De Marco	Keterangan
		Entitas eksternal, dapat berupa orang/unit terkait yang berinteraksi dengan sistem tetapi diluar sistem
		Orang, unit yang mempergunakan atau melakukan transformasi data. Komponen fisik tidak diidentifikasi.
		Aliran data dengan arah khusus dari sumber ke tujuan
		Penyimpanan data atau tempat data direfer oleh proses.

Gambar 2.1 Notasi Data Flow Diagram

2.4.3 Entity Relationship Diagram (ERD)

Entity Relationship Diagram adalah suatu teknik untuk mengorganisasikan dan mendokumentasikan data dari sistem. ERD sering juga disebut *database modeling*, karena data model biasanya diimplementasikan dalam sebuah database. Sering juga disebut *Information modeling*. Konsep mengenai data yang akan ditangkap dan disimpan. Setiap entitas dapat dibedakan dengan entitas lainnya.

Sebuah *entity instance* adalah sebuah *occurrence* tunggal dari sebuah entitas. Sebagai contoh entitas STUDENT memiliki banyak *instance*. Mary, Joe, Mark, Susan, Cheryll dan sebagainya.

Atribut adalah keterangan properti atau karakteristik dari suatu entitas. Sering juga disebut *element, property dan field*.

Relationship adalah asosiasi bisnis secara umum yang terdapat antara satu atau lebih entitas. *Relationship* dapat menggambarkan suatu kejadian yang menghubungkan entitas-entitas atau hubungan secara logis yang ada antar entitas.

Cardinality menunjukkan jumlah minimum dan maksimum dari kejadian satu entitas yang berhubungan dengan sebuah kejadian dari entitas lain. Karena semua *relationship bidirectional*, *cardinality* harus didefinisikan dalam dua arah untuk setiap hubungan.

2.4.4 Kamus Data

Menurut Kendall (2011) kamus data adalah suatu aplikasi khusus dari jenis kamus-kamus yang digunakan sebagai referensi. Kamus data merupakan hasil referensi dan mengenai data (metadata), suatu data yang disusun oleh penganalisis sistem untuk membimbing mereka selama melakukan analisis desain. Sebagai suatu dokumen, kamus data mengumpulkan istilah-istilah data tertentu dan menjelaskan apa arti setiap istilah yang ada.

Kamus data ikut berperan dalam perancangan dan pembangunan sistem informasi karena peralatan ini berfungsi untuk (Oetomo, 2002):

1. Menjelaskan arti aliran data dan penyimpanan dalam penggambaran dalam diagram aliran data.
2. Mendeskripsikan komposisi paket data yang bergerak melalui aliran misalnya data alamat diuraikan menjadi nama jalan, nomor, kota, negara dan kode pos.
3. Menjelaskan spesifikasi nilai dan satuan yang relevan terhadap data yang mengalir dalam sistem tersebut.

Kamus data otomatis sangat berguna karena memiliki kapasitas dalam hal referensi silang item-item data, dengan demikian memungkinkan dilakukannya perubahan-perubahan program terhadap semua program yang berbagi suatu elemen biasa. Fitur ini menggantikan pengubahan program serampangan, atau mencegah penundaan sampai program tidak bisa berjalan karena perubahan tersebut tidak diimplementasikan pada semua program yang berbagi item-item yang telah diperbaharui. Jelasnya kamus data otomatis menjadi sangat penting untuk sistem-sistem besar karena mampu menghasilkan ribuan elemen data yang dikatalogkan dan dibuat referensi silang. Kamus data dibuat dengan memperhatikan dan menggambarkan muatan aliran data, simpanan data dan proses-proses. Setiap simpanan data bisa diterapkan dan kemudian diperluas sampai mencakup detail-detail elemen yang dibuatnya yang mengalir menuju dan keluar dari proses tersebut. Ketidak hati-hatian dan kesalahan-kesalahan

perancangan lainnya bisa ditegaskan dan dicari penyelesaiannya. Untuk dokumentasi serta mengurangi redudansi, kamus data bisa digunakan untuk:

1. Menvalidasi Diagram aliran data dalam hal kerangkapan dan keakuratan.
2. Menyediakan suatu titik awal untuk mengembangkan layar dan laporan.
3. Menentukan muatan data yang disimpan di file-file.
4. Mengembangkan logika untuk proses-proses diagram aliran data.

Berikut ini sejumlah simbol yang digunakan dalam penggambaran kamus data (Oetomo,2002) :

Tabel 2.1 Simbol-simbol kamus data

Simbol	Uraian
=	Diuraikan menjadi, mendefinisikan atau artinya
+	Dan
()	Opsional (Pilihan boleh atau boleh tidak)
{ }	Pengulangan
[]	Memilih salah satu dari sejumlah alternatif atau seleksi
**	Komentar
	Pemisah sejumlah alternatif pilihan antara simbol []

Berikut ini adalah contoh penggunaan kamus data untuk perancangan database pembicaraan seminar (Oetomo, 2002):

1. Data_pembicara = kode_pembicara + nama + alamat + telepon + honor + spesialisai
2. Kode_pembicara = level_pembicara + nomor_urut
3. Nama = sebutan + nama1 + nama2 + gelar1 + gelar2
4. Alamat = jalan + nomor + kota + negara + kodepos
5. Telepon = kodearea + nomor
6. Honor = *999.999.999*
7. Spesialisasi = () {karakter} 15

2.5 Basis Data

Oetomo (2002) menjelaskan bahwa basis data merupakan himpunan kelompok data yang saling berkaitan. Data tersebut diorganisasikan sedemikian rupa agar tidak terjadi duplikasi data, sehingga dapat diolah atau dieksplorasi secara tepat dan mudah untuk menghasilkan informasi. Kendall & Kendall (2011) juga menambahkan bahwa basis data digunakan bersamaan oleh banyak pengguna untuk beragam aplikasi.

Dalam pembangunan basis data, analisis sistem harus dapat menentukan dalam model arsitektur mana basis data itu akan diletakkan. Dinilai dari penempatannya, arsitektur basis data dapat dikategorikan dalam tiga bagian, yaitu:

1. Sistem basis data tunggal

Pada arsitektur ini basis data dan aplikasinya diletakkan pada komputer yang sama dan tidak berada dalam lingkungan jaringan komputer, sehingga basis data itu hanya dapat diakses oleh aplikasi tunggal. Sistem ini biasanya digunakan oleh perusahaan kecil

2. Sistem basis data terpusat

Pada arsitektur ini, lokasi basis data secara fisik berada dalam komputer pusat dalam suatu lingkungan jaringan. Meskipun pemasukan dan akses data dapat dilakukan dari berbagai terminal yang terhubung dengan komputer tersebut, namun proses pengolahan data hanya berlangsung di komputer pusat. Dengan sistem ini komputer pusat. Dengan sistem ini komputer pusat menjadi titik kritis dari proses pengolahan basis data. Bila komputer pusat terganggu maka secara keseluruhan sistem informasi akan terganggu.

3. Sistem basis data terdistribusi

Pada arsitektur ini salinan basis data, baik sebagian maupun secara keseluruhan terdistribusi di beberapa lokasi. Pada model ini, titik kritis pada sistem terpusat dapat dihindari. Namun pada sistem ini, tantangan terbesar yang dihadapi adalah proses pengintegrasian untuk menjaga konsistensi data yang tersebar di beberapa lokasi. (Oetomo, 2002).

Dalam basis data dikenal suatu istilah konsep basis data. Dalam bidang realitas adalah entitas dan atribut-atribut, sedangkan dalam bidang sebenarnya adalah munculnya *record* dan *item* data kejadian, dan dalam bidang *record* dan definisi data.

- Entitas

Objek atau kejadian apapun mengenai seseorang yang memilih untuk mengumpulkan data adalah sebuah entitas. Entitas dapat berupa orang, tempat, atau sesuatu. Entitas apapun juga dapat merupakan satu kejadian atau unit waktu seperti mesin rusak, penjualan, bulan, atau tahun.

- Hubungan

Hubungan diasosiasikan antara entitas terdiri dari jenis:

1. Hubungan satu-ke-satu (ditandakan 1:1)
2. Hubungan satu-ke-banyak (1:B)
3. Hubungan banyak-ke-banyak (B:B)

- Atribut

Atribut merupakan beberapa karakteristik dari satu entitas. Terdapat beberapa atribut untuk masing-masing entitas.

- *Record*

Sebuah *record* adalah kumpulan item data yang memiliki sesuatu secara umum dengan entitas yang dideskripsikan.

- Metadata

Metadata adalah data mengenai data dalam *file* atau basis data. Metadata mendeskripsikan nama yang diberikan dan panjang yang ditentukan dari setiap item data. Metadata juga mendeskripsikan panjang dan komposisi setiap *record*.

2.6 Normalisasi Data

Pembangunan sistem informasi bertumpu pada kualitas basis data yang disusun dan dibentuk. Basis data yang dibentuk diharapkan memiliki sifat antara lain:

1. Efisien dan Efektif dalam pengorganisasiannya, artinya untuk menambah, menyisipkan atau menghapus data dapat dilakukan dengan mudah dan sederhana.
 2. Bebas Redudansi, meskipun pada batas-batas tertentu yang dapat ditolerir, redudansi juga diperbolehkan misalnya untuk mengurangi kompleksitas dalam penulisan program.
 3. Fleksibel, artinya basis data dapat diakses dengan mudah, dinamis dan tidak tergantung sepenuhnya pada aplikasi-aplikasi tertentu.
 4. Sistem Basis data yang dapat diakses secara bersama dalam lingkungan jaringan sehingga mendukung penggunaan bersama dan distribusi data.
- (Budi Sutejo Dharma Oetomo, 2002, 115).

Normalisasi merupakan peralatan yang digunakan untuk melakukan proses pengelompokan data menjadi tabel-tabel yang menunjukkan entitas dan relasinya. Normalisasi adalah transformasi tinjauan pemakai yang kompleks dan data tersimpan ke sekumpulan bagian-bagian struktur data yang kecil dan stabil. Di samping menjadi lebih sederhana dan lebih stabil, struktur data yang dinormalisasikan lebih mudah diatur daripada struktur data lainnya. Tahapan normalisasi yakni:

1. Tahap Pertama

Tahap pertama dari proses meliputi menghilangkan semua kelompok terulang dan mengidentifikasi kunci utama. Untuk mengerjakannya, hubungan perlu dipecah ke dalam dua atau lebih hubungan, pada titik ini, hubungan mungkin sudah menjadi bentuk normalisasi ketiga, bahkan lebih banyak tahap akan diperlukan untuk mengtransformasi hubungan ke bentuk normalisasi ketiga.

2. Tahap kedua

Tahap kedua menjamin bahwa semua atribut bukan kunci sepenuhnya tergantung pada kunci utama. Semua ketergantungan parsial diubah dan diletakkan dalam hubungan lain.

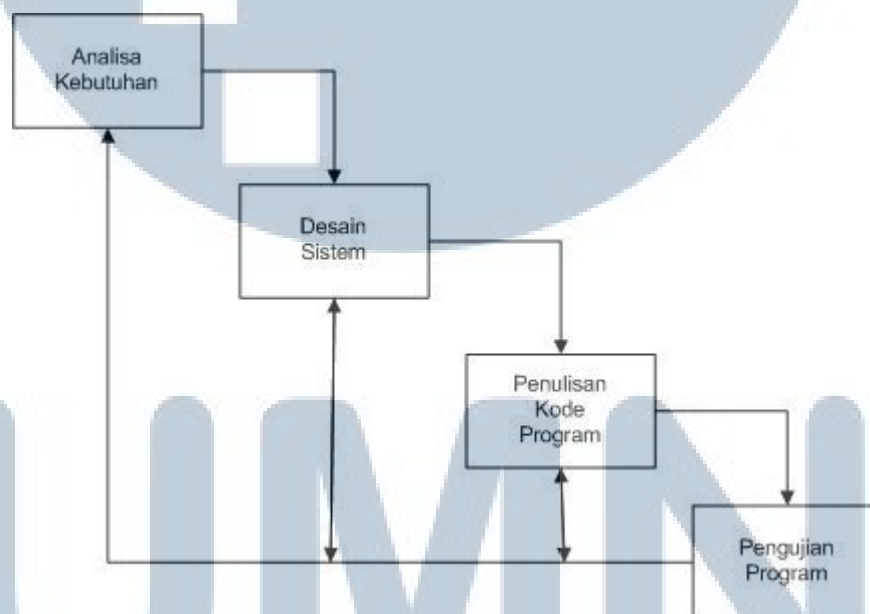
3. Tahap Ketiga

Tahap ketiga mengubah ketergantungan transitif manapun. Suatu ketergantungan transitif adalah sesuatu di mana atribut bukan kunci tergantung pada atribut bukan kunci lainnya.

Tujuan utama dari proses normalisasi adalah menyederhanakan semua kekompleksan item data yang sering ditemukan dalam tinjauan pemakai.

2.7 *Adapted Waterfall Model*

Menurut Budi Hermawan (2009), *Waterfall Model* adalah sebuah metode pengembangan software yang bersifat sekuensial dan terdiri dari 5 tahap yang saling terkait dan mempengaruhi seperti terlihat pada gambar berikut



Gambar 2.2 *Adapted Waterfall Model*

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Keterkaitan dan pengaruh antar tahap ini ada karena output sebuah tahap dalam *Waterfall Model* merupakan input bagi tahap berikutnya, dengan demikian ketidaksempurnaan hasil pelaksanaan tahap sebelumnya adalah awal ketidaksempurnaan tahap berikutnya. Memperhatikan karakteristik ini, sangat penting bagi tim pengembang dan perusahaan untuk secara bersama-sama melakukan analisa kebutuhan dan desain sistem sesempurna mungkin sebelum masuk ke dalam tahap penulisan kode program. Berikut adalah penjelasan detail dari masing-masing tahap dalam *Waterfall Model*.

A. Analisa Kebutuhan

Kelancaran proses pembuatan *software* secara keseluruhan dan kelengkapan fitur *software* yang dihasilkan sangat tergantung pada hasil analisa kebutuhan ini. Untuk memperoleh informasi tentang proses bisnis dan kebutuhan perusahaan, umumnya tim pengembang melakukan wawancara, diskusi dan survey. Dalam proses wawancara, diskusi dan survey diperlukan komunikasi yang intensif dan terbuka antara tim pengembang dan tim *counter-partner* dari perusahaan. Selain itu, peran aktif dari manajemen sebagai pihak yang memahami seluk beluk perusahaan secara komprehensif juga diperlukan agar diperoleh informasi proses bisnis dan daftar kebutuhan perusahaan yang lengkap. Beberapa perusahaan membantu memperlancar penyelesaian tahap ini dengan terlebih dahulu menyusun *scope of work software* yang akan dibuat sebagai acuan kerja tim pengembang. Hasil analisa kebutuhan yang tidak lengkap berpotensi menyebabkan beberapa permasalahan yang tidak

diharapkan, antara lain waktu pembuatan *software* menjadi lebih lama, proses dalam *software* tidak sesuai dengan proses bisnis dan *software* tidak dapat memenuhi semua kebutuhan perusahaan. Untuk meminimalkan risiko ini, disarankan perusahaan melakukan konfirmasi pemahaman tim pengembang tentang proses bisnis dan kebutuhan perusahaan dengan cara meminta *resume* hasil analisa kebutuhan dan menyempurnakannya bersama tim pengembang jika diperlukan.

B. Desain Sistem

Desain sistem merupakan tahap penyusunan proses, data, aliran proses dan hubungan antar data yang paling optimal untuk menjalankan proses bisnis dan memenuhi kebutuhan perusahaan sesuai dengan hasil analisa kebutuhan. Dokumentasi yang dihasilkan dari tahap desain sistem ini antara lain: *System Flow*, *Data Flow Diagram* (DFD) dan *Entity Relationship Diagram* (ERD) berikut dengan normalisasinya. *System Flow* merupakan bagan aliran dokumen dari satu bagian perusahaan ke bagian lain baik secara manual maupun melalui Sistem Informasi. *Data Flow Diagram* adalah diagram yang menunjukkan aliran data di antara pengguna, proses dan database yang terkait dengan *software*. *Entity Relationship Diagram* merupakan diagram yang menunjukkan bagaimana data dan informasi *software* akan di simpan di dalam database beserta dengan hubungan antar data. Seperti halnya tahap analisa kebutuhan, pada saat desain sistem selesai dibuat sebaiknya desain yang sudah dibuat

pengembang didiskusikan dan disempurnakan oleh tim pengembang bersama dengan tim *counter-partner* dan manajemen perusahaan.

C. Penulisan Kode Program

Penulisan kode program merupakan tahap penerjemahan desain sistem yang telah dibuat ke dalam bentuk perintah-perintah yang dimengerti komputer dengan mempergunakan bahasa pemrograman, *middleware* dan database tertentu di atas platform yang menjadi standar perusahaan. Bahasa pemrograman yang saat ini dipergunakan di perusahaan adalah: Microsoft Visual Basic .NET untuk pembuatan software dalam kategori *Desktop Application* dengan ASP.NET sebagai kontrol servernya. Database yang saat ini dipergunakan di perusahaan adalah : Microsoft SQL Server 2008 R2 *Enterprise Edition* untuk kebutuhan penyimpanan data dalam jumlah besar, fitur keamanan data yang handal dan penggunaan oleh banyak pengguna.

D. Pengujian Program

Pengujian *software* dilakukan untuk memastikan bahwa *software* yang dibuat telah sesuai dengan desainnya dan semua fungsi dapat dipergunakan dengan baik tanpa ada kesalahan. Pengujian *software* biasanya dilakukan dalam 2 atau 3 tahap yang saling independen, yaitu : pengujian oleh internal tim pengembang, pengujian oleh divisi *Quality Assurance* dan pengujian oleh pengguna di perusahaan. Dalam tahap ini,

perusahaan harus memastikan bahwa kerangka / skenario pengujian software dibuat dengan lengkap meliputi semua proses, kebutuhan dan pengendalian yang ada di dalam dokumen analisa kebutuhan dan desain sistem.

2.8 Intranet

Pembangunan server intranet sebagai salah satu bagian pembangunan infrastruktur teknologi informasi sangat dibutuhkan masyarakat untuk memenuhi segala kebutuhan masyarakat menjelang era informasi. Intranet (*Internal Network*) mulai didengung-dengungkan pada pertengahan tahun 1995 oleh beberapa penjual produk jaringan yang mengacu pada kebutuhan informasi dalam bentuk web dalam perusahaan.

Intranet adalah jaringan komputer dalam perusahaan atau organisasi yang menggunakan komunikasi data standar seperti dalam internet. Artinya, dapat menggunakan semua fasilitas internet untuk kebutuhan dalam perusahaan. Dengan kata lain intranet dapat dikatakan berinternet dalam lingkungan perusahaan.

Intranet banyak mendapat keuntungan karena suksesnya dukungan *World Wide Web* yang memungkinkan penggunaan yang luas karena digunakan oleh masyarakat yang menggunakan internet. Karena teknologinya sudah terbentuk dari internet, maka kemudahan, rendahnya biaya operasi, *fleksibilitas* serta informasi aksesnya dapat lebih cepat.

2.9 .NET Framework

.NET Framework merupakan sebuah platform komputasi baru yang mempermudah pengembangan aplikasi dalam lingkungan yang benar-benar terdistribusi di internet. .NET Framework memiliki dua komponen utama yaitu *Common Language Runtime* (CLR) dan *Librari Kelas .NET Framework*. CLR merupakan dasar dari .NET Framework.

Komponen *librari kelas* merupakan sekumpulan tipe yang dapat digunakan kembali dan berorientasikan objek, yang dapat digunakan untuk mengembangkan bermacam-macam aplikasi mulai dari aplikasi GUI (*Graphical User Interface*) sampai pada aplikasi berbasis inovasi terakhir yang disediakan oleh ASP.NET, seperti web form.

Dengan .NET Framework kita dapat membuat aplikasi windows, aplikasi *web services*, dll di berbagai sistem platform, bahkan di alat-alat *mobile* seperti PDA dengan cepat dan mudah tanpa harus memikirkan bagaimana harus berinteraksi dengan *hardware*, sistemnya, dan tanpa harus mengubah code untuk disesuaikan dengan *system platform* tertentu. Selama terdapat .NET Framework pada sistem tersebut.

.NET Framework telah didesain sehingga dapat digunakan dengan berbagai macam bahasa yaitu C#, VB .NET, C++, J#. Ini tentunya sangat memudahkan bagi para developer, jadi mereka tidak perlu harus mendalami semua bahasa, cukup satu saja dan hasil aplikasinya akan sama persis. Karena seluruh bahasa yang digunakan tadi akan *dcompile* dengan .NET Framework dan akan menghasilkan keluaran yang sama.

Tabel di bawah ini adalah tabel perbandingan .NET Framework versi pertama dan versi yang digunakan dalam penelitian ini.

Tabel 2.2 Perbandingan Versi .NET Framework

	.NET Framework 1.0	.NET Framework 2.0
Tanggal Rilis	13 Februari 2002	7 November 2005
Versi Visual Studio	Visual Studio .NET	Visual Studio 2005
Default pada Windows	N/A	Windows Server 2003 R2
Versi CLR	1.0	2.0
Fitur	Versi Pertama dari CLR dan dasar Librari Kelas	• Generik • Dukungan bahasa untuk generik yang dirancang langsung kedalam .NET CLR • Dukungan penuh 64 bit untuk platform x64 ataupun IA-64 • Integrasi SQL Server • Hosting baru API untuk aplikasi <i>native</i> yang ingin menjadi host dari sebuah

		<i>instance</i> dalam <i>.NET runtime</i> • Banyak tambahan dan peningkatan kontrol web ASP .NET • Kontrol data baru dengan <i>data binding</i> deklaratif • Fitur personalisasi yang baru untuk ASP .NET
--	--	--

2.9.1 ASP.NET

ASP.NET adalah pengembangan terbaru dari Microsoft Active Server Pages (ASP) dan merupakan suatu teknologi *server side* yang baru dan handal untuk membuat halaman web dinamis. ASP.NET merupakan platform pembuatan aplikasi web yang menyatu dengan .NET Framework serta menyediakan fasilitas-fasilitas bagi developer untuk membangun aplikasi web untuk level enterprise. ASP.NET dikompilasi dan berbasis .NET Framework sehingga dapat membuat aplikasi dalam berbagai macam bahasa yang mendukung .NET (kompatibel dengan .NET), termasuk Visual Basic .NET, Jscript .NET.

ASP.NET didesain supaya dapat bekerja dengan WYSIWYG HTML editor dan tool pemrograman lain termasuk Microsoft Visual Studio .NET. Hal ini dapat membuat pembuatan aplikasi web lebih mudah, tetapi juga membuat berbagai fasilitas yang ada pada tool tersebut dapat digunakan, termasuk GUI (*Graphical User Interface*) dimana developer dapat men-*drag and drop* server kontrol ke halaman web dan menggunakan fasilitas debugging yang ada.

Keunggulan penggunaan ASP.NET sebagai sebuah aplikasi berbasis web adalah sebagai berikut:

1. Penyederhanaan. ASP.NET membuat mudah tugas umum seperti pembuatan form, otentikasi *client*, validasi data, konfigurasi situs, dan *development*.
2. Perbaikan performa. ASP.NET dikompilasi ke CLR sehingga performanya lebih baik dari ASP yang *interpreter*.
3. *Web Form*. Merupakan model pemrograman baru yang menggabungkan aplikasi ASP dengan kemudahan pengembangan dan produktifitas *Visual Basic*
4. Kode non spaghetti. Model pemrograman ASP.NET memisahkan kode dari presentasi sehingga mempermudah membuat konstruksi dan mengelola kode
5. Perbaikan Manajemen Status. ASP.NET menyediakan status aplikasi dan sesi yang mudah digunakan. ASP.NET mengatasi keterbatasan tersebut dengan menyediakan dukungan pendistribusian status sesi dalam server

web, menaruh informasi status dalam SQL Server, serta menyediakan pengelolaan status tanpa *cookies*.

6. Keamanan. ASP.NET menyediakan layanan otorisasi (menentukan apakah pengguna memiliki izin untuk melakukan tindakan yang diminta) dan otentikasi (menentukan identitas pengguna yang melakukan permintaan) yang telah diperbaiki menggunakan “CookieAuthenticationModule” dan “URLAuthorizationModule”.
7. Konfigurasi. ASP.NET menggunakan file XML untuk menyimpan pengaturan konfigurasi. Hal ini membuat pengembangan situs menjadi lebih mudah
8. Layanan Web. ASP.NET dapat digunakan untuk mengekspos fungsi bisnis ke partner melalui protokol Web standar.
9. *Caching*. Disediakan mesin *caching* untuk meningkatkan kinerja aplikasi dan mengurangi beban pemroses server web dan server *database*
10. *Debugging*. ASP.NET memiliki utilitas *tracking* yang *build-in*.
11. *Deployment*. *Deployment* dapat dilakukan dengan cara menyalin file karena semua pengaturan konfigurasi situs terdapat dalam file XML.

2.9.2 Visual Basic .NET

Visual Basic .NET adalah sebuah perangkat lunak untuk mengembangkan dan membangun aplikasi yang bergerak di atas sistem .NET Framework, dengan menggunakan bahasa BASIC. Dengan menggunakan perangkat lunak ini, para programmer dapat membangun aplikasi Windows Forms, Aplikasi web berbasis

ASP .NET, dan juga aplikasi command-line. Perangkat ini dapat diperoleh secara terpisah dari beberapa produk lainnya (seperti Microsoft Visual C++, Visual C#, atau Visual J#) atau juga dapat diperoleh secara terpadu dalam Microsoft Visual Studio .NET. Bahasa Visual Basic .NET sendiri menganut paradigma bahasa pemrograman berorientasi objek yang dapat dilihat sebagai evolusi dari Microsoft Visual Basic versi sebelumnya yang diimplementasikan di atas .NET Framework. Peluncurannya mengundang kontroversi, mengingat banyak sekali perubahan yang dilakukan oleh Microsoft, dan versi baru ini tidak kompatibel dengan versi terdahulu.

Keunggulan dari penggunaan Visual Basic .NET sebagai *code-behind programming language* dalam ASP.NET adalah sebagai berikut (Microsoft, 2010):

1. Kurva pembelajaran dan pengembangan yang lebih singkat dibandingkan bahasa pemrograman yang lain seperti C/C++, Delphi atau bahkan *PowerBuilder* sekalipun.
2. Menghilangkan kompleksitas pemanggilan fungsi Windows API, karena banyak fungsi-fungsi tersebut sudah ditanamkan ke dalam syntax Visual Basic.
3. Cocok digunakan untuk mengembangkan aplikasi/program yang bersifat "*Rapid Application Development*".
4. Sangat cocok digunakan untuk membuat program/aplikasi bisnis.
5. Kemudahan sinkronisasi antar produk microsoft lainnya.

6. Dapat membuat ActiveX Control sehingga terintegrasi dengan Crystal Report.
7. Dapat menggunakan OCX/Komponen yang disediakan oleh pihak ketiga (“*Third Party*”) sebagai *tools* pengembangan.
8. Menyediakan *wizard* yang sangat berguna untuk mempersingkat/mempermudah pengembangan aplikasi.
9. Mendekati *Object Oriented Programming* karena memiliki *Class*.
10. Dapat di-integrasikan dengan internet, baik itu pada sisi *Client* maupun pada sisi server.
11. Dapat membuat ActiveX *Automation Server*.
12. Integrasi dengan *Microsoft Transaction Server*.
13. Dapat menjalankan server dari mesin yang sama atau bahkan dari mesin/komputer yang lain.

