

BAB II

KAJIAN PUSTAKA

2.1. Tinjauan Tentang Rumah Sakit

2.1.1. Pengertian Rumah Sakit

Berdasarkan Peraturan Menteri Kesehatan Republik Indonesia No.269/MENKES/PER/XII/2008 dinyatakan bahwa sarana pelayanan kesehatan adalah tempat penyelenggaraan upaya pelayanan kesehatan yang digunakan untuk praktik kedokteran atau kedokteran gigi. (Menteri Kesehatan Republik Indonesia, 2008)

Menurut Rustyanto (2010) “Rumah sakit adalah suatu institusi pelayanan kesehatan yang kompleks, padat karya dan padat modal. Kompleks ini muncul karena pelayanan rumah sakit menyangkut berbagai fungsi pelayanan, pendidikan dan penelitian serta mencakup berbagai tingkatan maupun jenis disiplin agar rumah sakit mampu melaksanakan fungsi yang profesional baik di bidang teknis medis maupun administrasi kesehatan untuk menjaga dan meningkatkan mutu rumah sakit harus mempunyai suatu ukuran yang menjamin peningkatan mutu di semua tingkatan.

2.1.2. Tugas dan Fungsi Rumah Sakit

Tugas dan fungsi dari rumah sakit berdasarkan Undang–Undang Republik Indonesia Nomor 44 Tahun 2009 adalah (*Undang-Undang Republik Indonesia Nomor 44 Tahun 2009, 2009*)

- a. Rumah sakit mempunyai tugas memberikan pelayanan kesehatan perorangan, secara paripurna.
- b. Fungsi rumah sakit :
 1. Penyelenggaraan pelayanan pengobatan dan pemulihan kesehatan sesuai dengan standar pelayanan rumah sakit,

2. Pemeliharaan dan peningkatan kesehatan perorangan melalui pelayanan kesehatan yang paripurna tingkat kedua dan ketiga sesuai kebutuhan medis,
3. Penyelenggaraan pendidikan dan pelatihan sumber daya manusia dalam rangka peningkatan kemampuan dalam pemberian pelayanan kesehatan,
4. Penyelenggaraan penelitian dan pengembangan serta penapisan teknologi bidang kesehatan dalam rangka peningkatan pelayanan kesehatan dengan memperhatikan etika ilmu pengetahuan bidang kesehatan.

2.2.Tinjauan Tentang Rekam Medis

2.2.1. Pengertian Rekam Medis

“Rekam Medis diartikan sebagai keterangan baik yang tertulis maupun yang terekam terutama identitas, anamnese, penentuan fisik laboratorium, diagnosa segala pelayanan dan tindakan medis yang diberikan kepada pasien dan pengobatan baik yang rawat inap, rawat jalan, maupun yang mendapatkan pelayanan gawat darurat” (Rustiyanto,2010)

“Rekam Medis adalah berkas yang berisikan catatan dan dokumen tentang identitas pasien, pemeriksaan, pengobatan, tindakan, dan pelayanan lain kepada pasien pada sarana pelayanan kesehatan” (Hatta, 2008).

Menurut Permenkes No.269/MENKES/PER/III/2008 “rekam medis adalah berkas yang berisikan catatan dan dokumen tentang identitas pasien, pemeriksaan, tindakan dan pelayanan lain yang telah diberikan kepada pasien”. (Menteri Kesehatan Republik Indonesia, 2008)

2.2.2. Kegunaan Rekam Medis

Menurut Rustyanto tentang Etika Profesi Perekam Medis dan Informasi Kesehatan menyatakan bahwa :

Kegunaan rekam medis secara umum antara lain sebagai berikut :

- a. Sebagai alat komunikasi antara dokter dengan tenaga ahlinya yang ikut ambil bagian didalam memberikan pelayanan pengobatan, perawatan kepada pasien.
- b. Sebagai dasar untuk merencanakan pengobatan/perawatan yang harus diberikan kepada seorang pasien.
- c. Sebagai bukti tertulis atas segala tindakan pelayanan, perkembangan penyakit, dan pengobatan selama pasien berkunjung/dirawat di rumah sakit.
- d. Sebagai bahan yang berguna untuk analisa, penelitian dan evaluasi terhadap kualitas pelayanan yang diberikan kepada pasien.
- e. Sebagai alat komunikasi antara dokter dengan tenaga ahlinya yang ikut ambil bagian didalam memberikan pelayanan pengobatan, perawatan kepada pasien.
- f. Sebagai dasar untuk merencanakan pengobatan/perawatan yang harus diberikan kepada seorang pasien.
- g. Sebagai bukti tertulis atas segala tindakan pelayanan, perkembangan penyakit, dan pengobatan selama pasien berkunjung/dirawat di rumah sakit.
- h. Sebagai bahan yang berguna untuk analisa, penelitian dan evaluasi terhadap kualitas pelayanan yang diberikan kepada pasien. Melindungi kepentingan hukum bagi pasien, rumah sakit maupun dokter dan tenaga kesehatan lainnya.

- i. Menyediakan data khususnya yang sangat berguna untuk penelitian dan pendidikan.
- j. Sebagai dasar didalam perhitungan biaya pembayaran pelayanan medis pasien.
- k. Menjadi sumber ingatan yang harus didokumentasikan, serta sebagai bahan pertanggungjawaban dan laporan

2.2.3. Rekam Medis Elektronik

Pada era global ini, teknologi elektronik juga berimbas pada terjadinya transformasi paradigma rekam medis tradisional terdiri dari yang berbasis kertas ke arah yang berbasis informasi disebut dengan Manajemen Informasi Kesehatan (MIK). Salah satu penggunaan IT dalam dunia kesehatan yang menjadi tren dalam dunia pelayanan kesehatan secara global adalah rekam medis elektronik (RME).

Sejak dikeluarkannya Undang–Undang Informasi dan Transaksi Elektronik (UU ITE) Nomor 11 tahun 2008 telah memberikan jawaban atas keraguan yang ada. Dimana penggunaan Rekam Medis Elektronik sebagai dokumen elektronik menjadi Rekam Medis Elektronik ini memiliki dasar hukum (*Kumpulan Peraturan Perundangan Rekam Medis dan Informasi Kesehatan (Manajemen Informasi Kesehatan)*, 2014).

2.3. Tinjauan Tracer

2.3.1. Pengeluaran Rekam Medis

Menurut (Depkes RI Dirjen Pelayanan Medik, 2006) menyatakan bahwa :

Ketentuan Pokok yang harus ditaati di tempat penyimpanan adalah

- a. Tidak satu pun rekam medis boleh keluar dari ruang rekam medis, tanpa tanda keluar atau kartu permintaan. Peraturan ini tidak hanya berlaku bagi orang–orang di luar ruang rekam medis, tetapi juga bagi petugas-petugas rekam medis sendiri.

- b. Seorang yang menerima atau meminjam rekam medis, berkewajiban untuk mengembalikan dalam keadaan baik dan tepat waktunya. Harus dibuat ketentuan berapa lama jangka waktu satu rekam medis diperbolehkan tidak berada di rak penyimpanan. Seharusnya setiap rekam medis kembali lagi ke raknya setiap akhir hari kerja, sehingga dalam keadaan darurat staf di rumah sakit dapat mencari informasi yang diperlukan.
- c. Rekam medis tidak dibenarkan diambil dari rumah sakit, kecuali atas perintah pengadilan.

Dokter-dokter atau pegawai-pegawai rumah sakit yang berkepentingan dapat meminjam rekam medis, untuk dibawa ke ruangan kerjanya selama jam kerja, tetapi semua rekam medis harus dikembalikan ke ruangan rekam medis pada akhir jam kerja. Jika beberapa rekam medis akan digunakan selama beberapa hari, rekam medis tersebut disimpan di tempat sementara di ruang rekam medis.

Kemungkinan rekam medis dipergunakan oleh beberapa orang perpindahan dari satu orang ke orang lain harus dilakukan dengan mengisi “Kartu Pindah Tangan”. Karena dengan cara ini rekam medis tidak perlu bolak-balik dikirimkan ke bagian Rekam Medis, untuk diletakkan sebagai petunjuk keluarnya rekam medis, kepada siapa, untuk keperluan apa dan digunakan oleh dokter siapa.

2.3.2. Petunjuk Keluar

Menurut (Depkes RI Dirjen Pelayanan Medik, 2006) disebutkan bahwa :

Petunjuk keluar atau tracer adalah suatu alat yang penting untuk mengawasi penggunaan rekam medis. Dalam penggunaannya, petunjuk keluar

(*Tracer*) ini diletakkan sebagai pengganti pada tempat rekam medis yang diambil (keluar) dari rak penyimpanan, petunjuk keluar tetap berada di dalam rak, sampai rekam medis yang diambil (dipinjam) kembali ke rak penyimpanan (Depkes RI Dirjen Pelayanan Medik, 2006).

Petunjuk keluar yang paling umum dipakai berbentuk kartu yang dilengkapi dengan kantong tempel tempat menyimpan surat pinjam. Petunjuk dapat diberi warna, yang dimaksud untuk mempercepat petugas melihat tempat-tempat penyimpanan kembali map-map rekam medis yang bersangkutan. Petunjuk keluar ini harus dibuat dari bahan (kertas) yang keras dan kuat.

2.3.3. Standar Pelayanan

Menurut Pedoman Penyelenggaraan Rekam Medis, Departemen Kesehatan RI tentang Standar Prosedur Peminjaman Berkas Rekam Medis (2006) menyatakan bahwa :

- a. Standar pengembalian rekam medis pasien pulang rawat maksimal 2x24 jam dari pasien ditetapkan oleh dokter dapat keluar dari rumah sakit.
- b. Standar pengembalian rekam medis pasien rawat jalan maksimal 1x24 jam dari pasien selesai menerima pemeriksaan dokter atau telah menerima tindakan medis dari rumah sakit.

2.4. Tinjauan Tentang *Microsoft Visual Basic 6.0*

2.4.1. Pengertian Microsoft Visual Basic 6.0

Visual Basic 6.0 merupakan bahasa pemrograman (*Language Program*) untuk menghasilkan program – program aplikasi berbasis *Windows*. Secara umum ada

beberapa manfaat yang diperoleh dari penggunaan program Visual Basic (Subari & Yuswanto, 2008), di antaranya :

- a. Dipakai dalam membuat program aplikasi berbasis *Windows*,
- b. Dipakai dalam membuat objek – objek pembantu program, seperti fasilitas *Help*, *Control Active X*, aplikasi internet dan sebagainya,
- c. Digunakan untuk menguji program (*Debugging*), dan menghasilkan program akhir *EXE* yang bersifat *Executable*, atau dapat langsung dijalankan.

Banyak fasilitas baru yang ditawarkan oleh *Visual Basic 6.0*, diantaranya adalah penambahan koleksi fungsi, fasilitas *New Code*, penambahan *interface* baru, dan lain-lain. Selain menyediakan tipe data sendiri yang berupa argumen atau properti dan metode publik, *Visual Basic 6.0* juga bisa menghasilkan *array* dari suatu fungsi atau properti suatu prosedur.

2.4.2. Komponen Program Microsoft Visual Basic 6.0

Komponen *Microsoft Visual Basic 6.0* terdiri dari beberapa elemen, yaitu :

- a. *Title Bar* adalah judul dari *project* yang sedang dijalankan,
- b. *Menu Bar* merupakan kumpulan perintah-perintah yang dikelompokkan berdasarkan fungsinya masing-masing,
- c. *Toolbar* merupakan sekumpulan tombol yang mewakili perintah tertentu, pada umumnya tombol-tombol yang terdapat pada *toolbar* merupakan perintah-perintah yang sering digunakan,

Jendela Project Explore adalah jendela yang mengandung semua *file* di dalam aplikasi *Visual Basic*. Setiap aplikasi dalam *Visual Basic* disebut

dengan istilah project (proyek), dan setiap proyek bisa mengandung lebih dari satu file. Pada project explorer ditampilkan semua file yang terdapat pada aplikasi (proyek) misalnya *form*, *module*, *class*, dan sebagainya.

- d. *Properties Windows* adalah *windows* yang mengandung semua informasi mengenai objek yang terdapat pada aplikasi *Visual Basic*. *Properties* adalah sifat dari sebuah objek, misalnya seperti nama, ukuran, posisi, dan sebagainya.
- e. *Toolbox* adalah tempat objek-objek yang bisa langsung *didrag* lalu *didrop* ke *form windows*.
- f. *Form Windows* adalah tempat untuk mendesain tampilan *visual* dari suatu aplikasi, objek disimpan di *form windows* yang diambil dari *toolbox*. Untuk mengubah nilai-nilai objek yang ada di dalam *form windows* menggunakan *properties*. *Code Windows* merupakan tempat untuk menulis kode program, letaknya sama dengan *form windows*. Untuk mengaksesnya, hanya dengan menekan tombol *view code* yang terdapat pada jendela *solution explore* dengan *double* klik pada objek yang akan diakses.
- g. *Context Menu* berisi *sort cut* yang suatu saat bisa digunakan untuk membuka sebuah *context* menu suatu objek. Untuk membuka *context* menu ini, dengan cara mengklik kanan objek yang akan dibuka *context* menunya.

2.5. Tinjauan Tentang SQL Server 2000

Menurut (Subari & Yuswanto, 2008) dinyatakan bahwa *SQL Server 2000* merupakan bahasa pemrograman yang dirancang khusus untuk komunikasi dengan *database* relasional guna mendukung aplikasi dengan arsitektur *client-server*. Konsep penerapannya adalah *database* ditempatkan di komputer pusat yang disebut dengan *server* dan informasinya digunakan bersama-sama oleh *user* yang menjalankan aplikasi pada komputer lokal yang disebut *Client*.

Perintah-perintah *query* dalam *SQL Server* :

- a. *Create* digunakan untuk membuat tabel baru,
- b. *Select* digunakan untuk menampilkan data sesuai kriteria yang ditentukan,
- c. *Insert* digunakan untuk menyisipkan atau menambahkan baris pada tabel,
- d. *Update* digunakan untuk mengupdate atau mengubah isi data dalam tabel,
- e. *Delete* digunakan untuk menghapus baris/*record* data dalam tabel,
- f. *Drop* digunakan untuk menghapus tabel.

Menurut Dadang Sudrajat (2003), dasar-dasar SQL mempunyai tiga jenis, antara lain :

- a. *Data Definition Language* (DDL)

Merupakan bagian dari sistem manajemen *database* yang dipakai untuk mendefinisikan dan mengatur semua atribut atau properti dari sebuah *database*. Dengan DDL juga dapat didefinisikan *database*, *table* atau *view*. Bentuk umum dari pernyataan-pertanyaan DDL yang sering digunakan yaitu membuat (*create*), mengganti (*alter*), membuang (*drop*) objek dan menentukan integritas *database*.

- b. *Data Manipulation Language* (DML)

Merupakan perintah-perintah yang digunakan untuk menampilkan data (*select*), menambah (*insert*), mengubah (*update*), menghapus (*delete*), baris data di dalam tabel, menuliskan perubahan ke dalam *disk* secara permanen (*commit*) dan membatalkan perubahan setelah perintah *commit* yang terakhir (*rollback*).

c. *Data Control Language (DCL)*

Digunakan untuk mengontrol hak-hak pada objek-objek *database*. Bentuk umum dari pernyataan-pernyataan DCL yang sering digunakan yaitu memberikan hak kepada *user* untuk mengakses *database* (*grant*), membuang hak yang telah diberikan (*revoke*) dan membuat sebuah *entry* dalam sistem keamanan yang melarang sebuah izin pada sebuah *account* melalui group atau keanggotaan *role* (*deny*).

2.6. Tinjauan Tentang *Crystal Report*

“*Crystal report* merupakan program yang dapat digunakan untuk membuat, menganalisa dan menerjemahkan informasi yang terkandung dalam *database* atau program ke dalam berbagai jenis laporan yang sangat *flexibel*” (Andri Kuniyo dan Kusri, 2007).

Beberapa kelebihan dari *crystal report* adalah


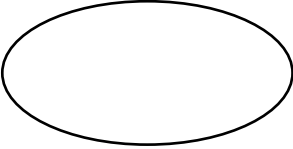
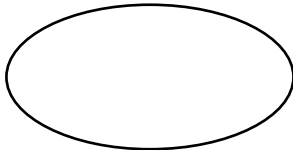
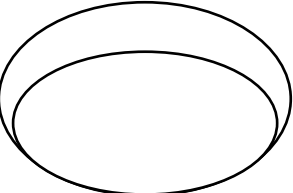
- a. Dari segi pembuatan laporan, tidak terlalu rumit yang memungkinkan para programmer pemula sekalipun dapat membuat laporan yang sederhana tanpa melibatkan banyak kode pemrograman.
- b. Integritas dengan bahasa-bahasa pemrograman lain yang memungkinkan dapat digunakan oleh banyak programmer dengan masing-masing keahlian.
- c. Fasilitas impor hasil laporan yang mendukung format-format populer seperti *Microsoft Word, Excel, Access, Adobe Acrobat Reader, HTML* dan sebagainya.

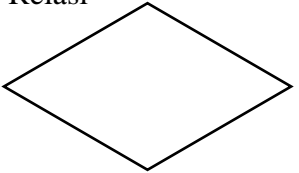
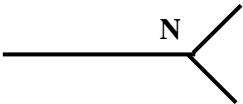
2.7. Tinjauan Tentang *Entity Relationship Diagram (ERD)*

Entity Relationship Diagram (ERD) adalah pemodelan awal basis data yang akan dikembangkan berdasarkan teori himpunan dalam bidang matematika untuk pemodelan basis data relasional (Sukanto dan Shlahudin M., 2014).

ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow's Foot, dan beberapa notasi lain. Namun banyak yang digunakan adalah notasi dari Chen (Sukamto dan Shlahudin M., 2014). Berikut adalah simbol-simbol yang digunakan pada ERD dengan notasi Chen :

Tabel 2. 1 Simbol - Simbol Entity Relationship Diagram (ERD)

SIMBOL	KETERANGAN
Entitas/ <i>Entity</i> 	Entitas merupakan data inti yang akan disimpan, bakal tabel pada basis data, benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer, penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.
Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
Atribut Kunci Primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan, biasanya berupa ide, kunci primer dapat lebih dari satu kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).
Atribut Multinilai/ Multivalue 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.

SIMBOL	KETERANGAN
Relasi 	Relasi yang menghubungkan antar entitas, biasanya diawali dengan kata kerja.
Asosiasi/Assosiation 	Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas I ke N sering disebut dengan <i>anotomy</i> menghubungkan entitas A dan entitas B.

2.8.Database

2.8.1. Pengertian Database

Database merupakan sekumpulan data atau informasi yang teratur berdasarkan kriteria tertentu yang saling berhubungan. Dalam dunia komputer, *database* bisa dikategorikan sangat spesial karena selalu menjadi hal utama dalam perancangan dalam perancangan sistem komputer suatu perusahaan (Subari & Yuswanto, 2008). Tentunya ada alasan tersendiri mengapa *database* menjadi prioritas utama dalam kinerja manajemen perusahaan, di antaranya :

1. *Database* tidak hanya berisi data tetapi juga berisi rencana atau model data.
2. *Database* dapat menjadi sumber utama yang digunakan secara bersama-sama oleh berbagai pemakai dalam perusahaan sesuai dengan kebutuhan.

2.8.2. Komponen Database

Menurut (Winarko, 2006) komponen utama dalam penerapan *database* adalah

1. Hardware

Perangkat keras yang biasanya terdapat dalam sistem *database* adalah komputer, *hardisk*, *disk*, media komunikasi (jaringan) dan lain-lain.

2. *Operating System*

Sistem operating adalah program yang mengaktifkan suatu komputer, mengendalikan seluruh *resource* yang ada serta melakukan operasi-operasi dasar I/O, pengolahan file dan lain-lain. Beberapa sistem operasi yang ada, yaitu MS-Windows, MS-Windows NT, Unix, Linux, Sun-Solaris.

3. Program Aplikasi

Perangkat lunak yang digunakan untuk membuat *user interface* yang berfungsi sebagai perantara untuk berkomunikasi dengan *database*. Beberapa program yang biasa digunakan adalah *visual basic*, *C-Builder++*, *visual foxpro*, *visual dbase*, PHP, dan lain-lain.

4. *Database*

Dalam sebuah *disk* penyimpanan bisa terdapat beberapa *database*, dimana dalam sebuah *database* bisa terdapat banyak tabel yang menunjukkan kumpulan data-data yang memiliki keamanan.

5. *User*

Pada sistem komputer ada beberapa tipe *user* dalam penanganan *database*, yaitu :

a. Programmer aplikasi (*application programmer*)

User ini berinteraksi dengan *database* dengan menggunakan data *manipulation language* (DML).

b. *User* mahir (*casualuser*)

User yang berinteraksi dengan *database* tanpa menulis modul program, tetapi menggunakan query yang telah disediakan oleh suatu DBMS.

c. *User* umum (*enduser* atau *native user*)

User yang berinteraksi dengan *database* dengan melalui pemanggilan sebuah program aplikasi yang permanen (*executable program*).

d. *User* khusus (*specialized user*)

User yang menulis aplikasi *database* untuk keperluan-keperluan khusus, seperti aplikasi AI, sistem pakar, pengolahan citra dan lain-lain.

2.8.3. Elemen - Elemen *Database*

Menurut (Kadir, 2014), elemen-elemen *database* relasional terdiri dari :

1. Tabel

Tabel merupakan kumpulan informasi secara logis yang terkait dan diperlakukan sebagai unit. Setiap tabel terdiri atas lajur vertikal yang biasa disebut dengan kolom atribut (*coloumn/field*) dan lajur *horizontal* yang biasa disebut dengan baris data (*row/record*).

2. Kolom (*field*)

Kolom merupakan atribut data yang ada pada suatu tabel pada *database*.

3. Baris (*record*)

Baris adalah kejadian tunggal yang berisi data di dalam tabel. Suatu tabel terdiri dari baris (*record*) yang mengisi setiap kolom (*field*).

4. Kunci (*key*)

a. *Primary key*

Primary key adalah suatu kolom yang memiliki nilai unik dan digunakan untuk mengidentifikasi setiap baris dalam tabel. Karena sifatnya yang unik maka dapat digunakan sebagai acuan dalam

menampilkan, mengubah dan menghapus baris data. Dengan adanya *primary key* juga dapat mencegah terjadinya duplikasi data karena sifatnya yang unik.

b. *Foreign key*

Foreign key merupakan nilai kolom pada suatu tabel yang berfungsi sebagai penghubung antara tabel dan dapat memberikan keterkaitan antara baris data pada satu tabel dengan baris data pada tabel lainnya.

2.8.4. Tabel *Database*

Menurut (Winarko, 2006), tabel menyimpan data yang dikelompokkan di dalam bentuk baris dan kolom seperti layaknya lembar kerja. Setiap baris mewakili *record*, dan setiap kolom adalah atribut atau *field*. Setiap *field* mengandung satu jenis informasi. Tabel dalam *database* dibedakan menjadi 2, yaitu :

1. Tabel Master

Tabel master merupakan tabel yang mewakili entitas tertentu, ia berdiri independen. Tabel master yang nantinya akan dilakukan proses manipulasi data, seperti : *update, insert, delete*. Contohnya : tabel barang, mahasiswa, dosen, mata kuliah, *customer, supplier*, dan sebagainya.

2. Tabel Transaksi

Tabel transaksi merupakan tabel yang terbentuk dari hasil transaksi pada suatu *form* transaksi. Ia tidak bersifat independen atau bergantung dengan tabel lain. Contohnya : tabel transaksi pemesanan barang, transaksi penjualan, dsb. Tabel transaksi ini dibagi menjadi 2, yaitu :

a. Transaksi *Header*

Tabel dimana setiap *field* hanya ditulis/diinput/disimpan sekali. Artinya dalam sebuah faktur contohnya tanggal faktur, nomor faktur, ide pelanggan akan ditulis atau diinput satu kali saja atau tidak berulang.

b. Transaksi Detail

Tabel dimana setiap *field* yang ditulis/diinput/disimpan boleh lebih dari satu kali. Artinya dalam satu nomor faktur, kita dapat melakukan transaksi lebih dari satu kali. Misalnya kita dapat memesan dua monitor, satu keyboard, satu “tikus”. Dimana data tersebut boleh berulang dalam satu nomor faktur.

2.9. Metodologi Pengembangan Sistem

Penelitian ini menggunakan metodologi pembangunan sistem “*System Development Life Cycle (SDLC)*” dengan metode *waterfall*.

Menurut (Simarmata, 2010), *waterfall* model adalah model yang memacu tim pengembang untuk mengumpulkan dan menentukan apa yang seharusnya dilakukan sebelum sistem dikembangkan. Model ini cocok untuk sistem yang mengedepankan kualitas dibandingkan biaya pengembangan atau waktu pengembangan. Kelebihan *waterfall* model adalah memberikan kemudahan serta kejelasan interpretasinya. Model ini terstruktur serta cocok diadaptasi untuk *management control*.

Terdapat enam tahapan dalam metode SDLC sebagai berikut:

1. Perencanaan

Tahapan ini akan menghasilkan dokumen *user requirement* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan *software*, termasuk rencana yang akan dilakukan.

2. Analisis

Analisis sistem dilakukan untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikan. Pada tahap ini dilakukan dalam analisis sistem, antara lain :

a. Analisa Identifikasi Masalah

Identifikasi masalah merupakan langkah awal dari satu analisis sistem.

Dalam tahap ini diidentifikasi masalah yang harus dipecahkan.

b. Analisa Kebutuhan

Menganalisis kebutuhan pemakai sistem perangkat lunak (*user*) dan mengembangkan kebutuhan *user* (Sukamto dan Shlahudin M., 2014).

c. Analisa Kelayakan Sistem

Studi kelayakan digunakan untuk menentukan kemungkinan keberhasilan solusi yang diusulkan. Tahapan ini berguna untuk memastikan bahwa solusi yang diusulkan tersebut benar-benar dapat dicapai (Kadir, 2014).

3. Perancangan

Dari perangkat lunak adalah proses multi langkah yang focus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antar muka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya (Sukamto dan Shlahudin M., 2014).

4. Implementasi

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain (Sukamto dan Shlahudin M., 2014).

5. Pengujian

Pengujian *blackbox* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian metode ini memungkinkan perakayasa

perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. (Pressman, 2012)

6. Pemeliharaan

Pada dasarnya tahapan ini merupakan tahap yang membutuhkan waktu paling lama diantara semua tahapan. Tahapan ini merupakan tahapan penggunaan sistem oleh pengguna. Pengguna akan mengetahui hasil dari sistem yang telah diinginkan. Setelah itu, dilakukan tahap perawatan atau *maintenance*. Pemeliharaan suatu *software* diperlukan, termasuk didalamnya adalah pengembangan, karena *software* yang dibuat tidak selamanya seperti ini. Ketika dijalankan mungkin saja masih ada permasalahan yang tidak ditemukan sebelumnya, atau ada penambahan fitur - fitur yang belum ada pada *software* tersebut.

2.10. Pengujian System

2.10.1. Black Box Testing

Menurut (Pressman, 2012) *Black Box Testing* berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan *engineers* untuk memperoleh set kondisi *input* yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program.

Menurut (Pressman, 2012), *Black Box-Testing* berusaha untuk menentukan kesalahan dalam kategori berikut

1. Fungsi yang tidak benar atau fungsi yang hilang,
2. Kesalahan antarmuka,
3. Kesalahan dalam struktur data atau akses *database* eksternal,
4. Kesalahan perilaku (*behavior*) atau kesalahan kinerja,
5. Inisialisasi dan pemutusan kesalahan.

Menurut (Pressman, 2012) tes ini dirancang untuk menjawab beberapa pertanyaan-pertanyaan sebagai berikut

1. Bagaimana validitas fungsional diuji?
2. Bagaimana perilaku dan kinerja sistem diuji?
3. Apa kelas *input* akan membuat kasus uji yang baik?
4. Apakah sistem *sensitive* terhadap nilai *input* tertentu?
5. Bagaimana batas-batas kelas data yang terisolasi?
6. Kecepatan dan volume data seperti apa yang dapat ditolerir sistem?
7. Efek apakah yang akan menspesifikasikan kombinasi data dalam sistem operasi?