

BAB 2

KAJIAN PUSTAKA

2.1 Rekam Medis

2.1.1 Pengertian Rekam Medis

Menurut Peraturan Menteri Kesehatan Republik Indonesia Nomor 269/MENKES/PER/III/2008 tentang rekam medis. Rekam medis adalah dokumen yang berisikan catatan dan dokumen tentang identitas pasien, pemeriksaan, pengobatan, tindakan dan pelayanan lain yang telah diberikan kepada pasien (Permenkes RI, 2008)

Dalam Peraturan Menteri kesehatan Nomor 749A/MENKES/PER/XII/1989 tentang rekam medis. Dijelaskan bahwa rekam medis adalah dokumen yang berisikan catatan dan dokumen tentang identitas pasien, pemeriksaan, pengobatan, tindakan dan pelayanan lain kepada pasien pada sarana pelayanan kesehatan (Depkes RI, 1989)

2.1.2 Kegunaan Rekam Medis

Kegunaan rekam medis menurut Direktorat Jendral Bina Pelayanan Medik dalam bukunya pedoman Penyelenggaraan dan Prosedur Rekam Medis di Rumah Sakit di Indonesia revisi II 2006:13-15.

Terdiri dari beberapa aspek diantaranya aspek administrasi, legal, financial, riset, edukasi dan dokumentasi, yang dijelaskan sebagai berikut:

1. Aspek Administrasi

Suatu dokumen rekam medis mempunyai nilai administrasi, karena isinya menyangkut tindakan berdasarkan wewenang dan tanggung jawab sebagai tenaga medis dan para medis dalam mencapai tujuan pelayanan kesehatan.

2. Aspek Medis

Suatu dokumen rekam medis mempunyai nilai medis, karena catatan tersebut dipergunakan sebagai dasar untuk merencanakan pengobatan/perawatan yang diberikan kepada pasien dan dalam rangka mempertahankan serta meningkatkan mutu pelayanan melalui kegiatan audit medis, manajemen risiko klinis serta keamanan/keselamatan pasien dan kendali biaya.

3. Aspek Hukum

Suatu dokumen rekam medis mempunyai nilai hukum, karena isinya menyangkut masalah adanya jaminan kepastian hukum atas dasar keadilan dalam rangka usaha menegakkan hukum dan menyediakan bahan tanda bukti untuk menegakkan keadilan.

4. Aspek Keuangan

Suatu dokumen rekam medis mempunyai nilai uang, karena isinya mengandung data/informasi yang dapat dipergunakan sebagai aspek keuangan. Rekam medis adalah milik dokter dan rumah sakit sedangkan isinya yang terdiri dari identitas pasien, pemeriksaan, pengobatan, tindakan dan pelayanan lain yang telah diberikan kepada pasien adalah sebagai informasi yang dapat dimiliki oleh pasien sesuai dengan peraturan dan perundang-undangan yang berlaku (Undang-undang Praktik Kedokteran RI No.29 Tahun 2004 pasal 47 ayat (1)).

5. Aspek Penelitian

Suatu dokumen rekam medis mempunyai nilai penelitian, karena isinya menyangkut data dan informasi yang dapat dipergunakan sebagai aspek penelitian dan pengembangan ilmu pengetahuan di bidang kesehatan.

6. Aspek Pendidikan

Suatu dokumen rekam medis mempunyai nilai pendidikan, karena isinya menyangkut data/informasi tentang perkembangan kronologis dan kegiatan pelayanan medis yang diberikan kepada pasien. Informasi tersebut dapat dipergunakan sebagai bahan/referensi pengajaran di bidang profesi kesehatan.

7. Aspek Dokumentasi

Suatu dokumen rekam medis mempunyai nilai dokumentasi, karena isinya menyangkut sumber ingatan yang harus didokumentasikan yang dipakai sebagai bahan pertanggungjawaban laporan di rumah sakit. Perkembangan ilmu pengetahuan dan teknologi informasi dapat diaplikasikan penerapannya didalam penyelenggaraan dan pengelolaan rekam medis yang cukup efektif dan efisien. Pendokumentasian data medis seorang pasien dapat dilaksanakan dengan mudah dan efektif sesuai aturan serta prosedur yang telah ditetapkan.

2.1.3 Rekam Medis Elektronik

Salah satu penggunaan teknologi informasi (IT) di bidang kesehatan yang menjadi trend dalam pelayanan kesehatan secara global adalah rekam medic elektronik. Di Indoneisa, dikenal dengan Rekam Medik Elektronik (RME). RME sudah banyak digunakan di berbagai Rumah sakit di dunia sebagai pengganti atau pelengkap rekam medic kesehatan berbentuk kertas.

Selama ini rekam medik mengacu pada pasal 46 dan 47 UU No. 29 Tahun 2004 tentang praktik kedokteran dan permenkes No.269/Menkes/PER/III/2008 tentang rekam medik (Depkes RI, 2008), sebagai pengganti dari Peraturan Menteri Kesehatan No.749a /Menkes/PER/XII/1989. Tetapi sejak dikeluarkannya Undang-undang Informasi dan Transaksi Elektronik (UU ITE) Nomor 11 Tahun 2008 telah memberikan jawaban atas keraguan yang ada. UU ITE telah memberikan peluang untuk implementasi RME.

2.2 *Filling dan Retrieval*

Menurut Sugiarto (2005) “Sistem penyimpanan adalah sistem yang digunakan pada penyimpanan dokumen agar kemudahan kerja penyimpanan dapat diciptakan dan penemuan dokumen yang sudah disimpan dapat dilakukan dengan cepat bilamana dokumen tersebut sewaktu waktu dibutuhkan”.

Ada dua cara pengurusan penyimpanan dalam penyelenggaraan Rekam Medis yaitu:

1. Sentralisasi adalah penyimpanan rekam medis seorang pasien dalam satu kesatuan baik catatan-catatan kunjungan poliklinik (rawat jalan) maupun catatan-catatan selama seorang pasien dirawat (rawat inap) yang disimpan dalam satu tempat.
2. Desentralisasi adalah penyimpanan antara dokumen rekam medis rawat jalan dan rawat inap terpisah.

Sedangkan *retrieval* adalah pengambilan kembali dokumen rekam medis di tempat penyimpanan. Pengeluaran dokumen rekam medis harus ada

ketentuan/peraturan tentang pengeluaran DRM yang ditentukan oleh Rumah Sakit diantaranya :

1. DRM tidak boleh dibawa keluar ruang penyimpanan tanpa surat permintaan yang sah.
2. Identitas peminjam dan tujuan peminjaman harus jelas.
3. Jangka waktu peminjaman.
4. Nama petugas yang mengeluarkan.
5. Penggunaan DRM untuk *research* harus dengan persetujuan pimpinan.
6. DRM tidak dibenarkan berada diluar rumah sakit kecuali atas perintah pengadilan.

2.3 Pengeluaran Rekam Medis

Ketentuan pokok yang harus ditaati ditempat penyimpanan adalah:

- a. Tidak satu pun rekam medis boleh keluar dari ruang rekam medis, tanpa tanda keluar/kartu peminjaman. Peraturan ini tidak hanya berlaku bagi orang-orang diluar ruang rekam medis, tetapi juga bagi petugas-petugas rekam medis sendiri.
- b. Seseorang yang menerima/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 dirak penyimpanan. Seharunya setiap rekam medis kembali lagi ke raknya pada setiap akhir hari kerja, sehingga dalam keadaan darurat staf rumah sakit dapat mencari informasi yang diperlukan.

- c. Rekam medis tidak dibenarkan diambil dari rumah sakit, kecuali atas perintah pengadilan.
- d. Dokter-dokter atau pegawai rumah sakit yang berkepentingan dapat meminjam rekam medis, untuk dibawa ke ruang kerjanya selama jam kerja, tetapi semua rekam medis harus dikembalikan ke ruang rekam medis pada akhir jam kerja
- e. Jika beberapa rekam medis akan digunakan selama beberapa hari, rekam medis tersebut disimpan dalam tempat sementara diruang rekam medis.

Kemungkinan rekam medis dipergunakan oleh beberapa orang perpindahan dari orang satu ke lain orang lain, harus dilakukan dengan mengisi “Kartu Pindah Tangan”, karena dengan cara ini rekam medis tidak perlu bolak balik dikirim ke bagian rekam medis. Kartu pindah tangan ini dikirimkan ke bagian rekam medis, untuk diletakkan sebagai petunjuk keluarnya rekam medis. Kartu pindah tangan tersebut berisi : tanggal, pindah tangan dari siapa, kepada siapa, untuk keperluan apa dan digunakan oleh dokter siapa.

2.4 Peminjaman Rekam Medis

Ketentuan peminjaman dokumen rekam medis merujuk pada Permenkes nomor 269 tahun 2008 tentang Rekam Medis, bahwa yang berhak meminjam dokumen rekam medis hanya dokter yang merawat. Secara umum peminjaman rekam medis dibagi menjadi 2 (dua) yaitu pinjaman rutin dan tidak rutin. Peminjaman rutin adalah peminjaman dokumen rekam medis oleh dokter dikarenakan pasien yang memiliki dokumen tersebut memerlukan atau sedang mendapatkan perawatan di unit pelayanan. Peminjaman tidak rutin adalah

peminjaman dokumen rekam medis oleh tenaga kesehatan atau dokter untuk keperluan penelitian, makalah atau sejenisnya. Ketentuan mengenai peminjaman ini dengan ketentuan sebagai berikut :

- a. Semua dokumen rekam medis yang keluar dari ruangan rekam medis wajib dicatat pada *tracer*) Semua dokumen rekam medis rawat jalan harus kembali dalam waktu 1x24 jam, diluar ketentuan tersebut perawat/dokter yang masih membutuhkan rekam medis, wajib memberitahu kepada petugas rekam medis pada rawat jalan)
- b. Dokumen rekam medis rawat inap harus dikembalikan ke unit rekam medis 2x 24 jam setelah pasien pulang) Peminjaman rekam medis untuk keperluan riset, penelitian dilakukan di unit rekam medik.

Pihak yang berhak meminjam DRM pasien:

- a. Pihak yang bertanggung jawab langsung terhadap pasien-para tenaga kesehatan (dokter, paramedis, fisioterapis dll)
- b. Pihak yang tidak bertanggung jawab langsung terhadap pasien yang diberi wewenang untuk menggunakan RM (petugas RM, staf medis)
- c. Pihak ketiga diluar RS yang tidak langsung bertanggung jawab terhadap pasien (asuransi, peneliti, polisi dll)

2.5 Pengembalian Rekam Medis

Dokumen rekam medis bersifat rahasia, (PP No. Tahun 1966) seorang yang menerima/meminjam rekam medis berkewajiban untuk menjaganya dalam

keadaan baik dan menyerahkan kepada petugas RM. Untuk peminjaman dari UGD dan Rawat Inap berlaku jam pengembalian

2.6 *Microsoft Visual Basic 6.0*

Menurut Stefano (2014) “*Visual Basic* merupakan sebuah bahasa pemrograman yang menawarkan *Intergrated Development Environment (IDE)* visual untuk membuat program perangkat lunak berbasis operasi *Microsoft Windows* menggunakan model pemrograman (COM)”.

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.
- d. Jendela Project Explorer adalah jendela yang mengandung semua file didalam 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.
- e. Properties Windows adalah sebuah windows yang mengandung semua informasi mengenai objek yang terdapat pada aplikasi Visual Basic.

Properties adalah sifat dari sebuah objek, misalnya seperti nama, warna, ukuran, posisi dan sebagainya.

- f. Toolbox adalah tempat objek-objek yang bisa langsung di drag lalu di drop ke *form* windows.
- g. *Form* Windows adalah tempat untuk mendesain tampilan visual dari suatu aplikasi, objek disimpan di *form* windows yang diambil dari toolbox. Untuk merubah nilai-nilai objek yang ada di dalam *form* windows bisa 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 explorer dengan double klik pada objek yang akan diakses.
- h. Context Menu berisi shortcut 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.
- i. Windows *Form* Layout dapat digunakan untuk mengontrol posisi *form* pada aplikasi dengan menggunakan sistem grafik dalam sebuah layar. Dengan fasilitas ini, anda dapat melihat dan mengetahui posisi *form* yang baru anda desain.

2.7 Entity Relationship Diagram (ERD)

Menurut Brady dan Loonam 2010 “*Entity Relationship Diagram (ERD)* merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu

organisasi, biasanya oleh System Analyst dalam tahap analisis persyaratan proyek pengembangan sistem". Brady (2010)

ERD merupakan suatu model jaringan yang menggunakan susunan data yang disimpan pada sistem secara abstrak. Biasanya ERD digunakan oleh perancang sistem untuk memodelkan data yang nantinya akan dikembangkan menjadi *Database*.

Secara garis besar ERD terdiri dari atas tiga komponen, yaitu entitas (*entity*), atribut (*attribute*), dan relasi atau hubungan (*relation*):

1. Entitas (*Entity*)

Entitas menunjukkan objek-objek dasar yang terkait di dalam sistem. Objek dasar dapat berupa orang, benda, atau hal lain yang keterangannya perlu disimpan dalam *Database*.

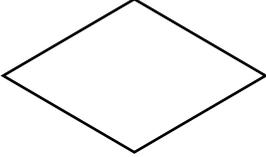
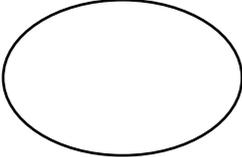
2. Atribut (*Attribute*)

Atribut sering juga disebut sebagai properti, merupakan keterangan-keterangan yang terkait pada sebuah entitas yang perlu disimpan sebagai *Database*.

3. Relasi (*Relation*)

Relasi atau hubungan adalah kejadian atau transaksi yang terjadi di antara dua entitas yang keterangannya perlu disimpan dalam *Database*.

Tabel 2. 1 Simbol ERD (Entity Relationship Diagram)

Simbol	Keterangan
 <p data-bbox="512 629 616 658">Entitas</p>	Entitas yaitu kumpulan dari objek yang dapat diidentifikasi secara unik
 <p data-bbox="520 920 608 949">Relasi</p>	Relasi yaitu hubungan yang terjadi antara salah satu lebih entitas. Jenis hubungan anantara lain: one to one, one to many, dan many to many.
 <p data-bbox="512 1216 616 1245">Atribut</p>	Atribut yaitu karakteristik dari entitas atau relasi yang merupakan penjelasan detail tentang entitas.
 <p data-bbox="491 1440 652 1469">Hubungan</p>	Hubungan antara entitas dengan atributnya dan himpunan entitas dengan himpunan relasinya.

2.8 Database

Menurut Edhy Sutanta (2004:18) “Kumpulan data terhubung (*interrelated data*) yang disimpan secara bersama-sama pada suatu media”(Sembiring, 2012)

Beberapa element dari Sistem Manajemen *Database* Relasional seperti dijelaskan berikut ini :

1. *Database*

Database merupakan sekumpulan data yang berisi informasi dan saling berhubungan. Pengertian ini sangat berbeda antara *Database* Relasional dan Non Relasional. Pada *Database* Non Relasional, sebuah *Database* yang berupa sebuah file.

2. Tabel

Sebuah tabel atau entitas dalam model relasional digunakan untuk mendukung antar muka komunikasi antara pemakai dengan profesional komputer. Dalam tabel itu sebenarnya merupakan matriks dari item-item data yang di organisir menjadi baris dan kolom. Kemudian terdapat istilah pada tabel, seperti:

a. Tabel Master

Tabel master merupakan tabel yang mewakili entitas tertentu, tabel ini berdiri independen. Tabel master yang nantinya akan dilakukan proses manipulasi data, seperti : update, insert, dan delete.

b. Tabel Transaksi

Tabel transaksi merupakan tabel yang terbentuk dari hasil transaksi pada suatu *form* transaksi, tabel ini bersifat independen atau bergantung dengan tabel lain.

3. *Record*

Record atau baris atau dalam istilah model rasional yang *formal* disebut dengan tuple adalah kumpulan data yang terdiri dari satu atau lebih suatu *field*. Pada setiap baris-baris ini tersimpan data-data dari subyek tabel yang

bersangkutan. Disamping itu data-data yang ada dalam satu *record* bisa terdiri dari bermacam-macam tipe data.

4. *Field*

Field atau kolom atau dalam istilah model relasional yang *formal* disebut dengan attribute adalah kumpulan data yang mempunyai/menyimpan fakta yang sama/sejenis untuk setiap baris pada tabel. Yang perlu diperhatikan bahwa urutan data dalam suatu kolom untuk tiap-tiap baris tidak memiliki arti sehingga data-data tersebut tidak berpengaruh walaupun diubah.

5. Indeks

Indeks merupakan tipe dari suatu tabel tertentu yang berisi nilai-nilai *field* kunci atau *field* (yang ditetapkan oleh pemakai) dan pointer kelokasi *record* yang sebenarnya. Nilai-nilai dari pointer ini disimpan dalam urutan tertentu dan dapat digunakan untuk meyajikan data dalam urutan *Database*.

6. Query

Query merupakan sekumpulan perintah SQL yang dirancang untuk memanggil kelompok *record* tertentu dari satu tabel atau lebih untuk melakukan operasi pada tabel. Meskipun SQL dapat dijalankan langsung dari program, query memungkinkan Anda menamai perintah dan menyimpannya dalam *Database* itu sendiri.

2.9 Microsoft Structured Query Language (SQL) Server 2005

SQL Server adalah *Relational Database Management System (RDBMS)* yang dirancang untuk aplikasi dengan struktur *client/server*. *SQL Server 2005*

merupakan salah satu produk dari Relational *Database Management System* (RDBMS) (Lancer, 2011)

Secara umum SQL Server adalah sebuah software yang dibuat oleh perusahaan *Microsoft* yang digunakan untuk membuat *Database* yang dapat diimplementasikan untuk *Client Server*. Client adalah setiap komponen dari sebuah sistem yang meminta layanan atau sumber daya (*resource*) dari komponen sistem lainnya. Sedangkan *server* adalah setiap komponen sistem yang menyediakan layanan atau sumber daya ke komponen sistem lainnya.

Komponen yang digunakan pada *Microsoft Structured Query Language* (SQL) server 2005 :

- a. Relational *Database Engine* : komponen utama atau jantung *SQL Server 2005*.
- b. Analysis Services : Basis dari solusi intelijen bisnis yang ampuh (powerful), dan mendukung aplikasi-aplikasi OLAP (online analytical processing), serta data minning.
- c. Data Transformation Service (DTS): sebuah mesin untuk membuat solusi ekspor dan impor data, serta untuk mentransformasi data ketika data tersebut ditransfer.
- d. Notification Services: sebuah framework untuk solusi dimana pelanggan akan dikirim notifikasi ketika sebuah event muncul.
- e. Reporting Services: service yang akan mengambil data dari SQL Server, dan menghasilkan laporan-laporan.
- f. Service broker: sebuah mekanisme antrian yang akan menangani komunikasi berbasis pesan diantara service.

- g. Native HTTP Support: dukungan yang memungkinkan *SQL Server 2005* yang (jika diinstall pada Windows Server 2003) akan merespon request terhadap HTTP endpoint, sehingga memungkinkan pembangunan sebuah web service untuk SQL Server tanpa menggunakan IIS.
- h. SQL server Agent : akan mengotomatiskan perawatan *Database* dan mengatur task, event dan alert.
- i. NET CLR (Common Language Runtime): akan memungkinkan pembuatan solusi menggunakan managed code yang ditulis dalam salah satu bahasa .NET.
- j. Replication: serangkaian teknologi untuk menjalin dan mendistribusikan data dan obyek *Database* dari sebuah *Database* ke *Database* lain, dan melakukan sinkronisasi untuk menjaga konsistensinya.
- k. Full-Text Search: memungkinkan pengindeksan yang cepat dan flexibel untuk query berbasis kata kunci (terhadap data teks yang disimpan dalam *Database*).

2.10 Structured Query Language (SQL)

Menurut (Raharjo, 2011) “SQL yaitu kependekan dari *Structured Query Language*, yang merupakan bahasa atau kumpulan perintah standar yang digunakan untuk berkomunikasi dengan *Database*”. Perintah SQL ini sering juga disingkat dengan sebutan “query”.

Perintah atau instruksi SQL dapat dikelompokkan berdasarkan jenis dan fungsinya. Terdapat tiga jenis perintah dasar SQL:

1. *Data Definition Language* (DDL)

Data Definition Language (DDL) adalah jenis instruksi SQL yang berkaitan dengan pembuatan struktur tabel maupun *Database*. Termasuk diantaranya: CREATE, DROP, ALTER, dan RENAME.

2. *Data Manipulation Language* (DML)

Data Manipulation Language (DML) adalah jenis instruksi SQL yang berkaitan dengan data yang ada dalam tabel, tentang bagaimana menginput, menghapus, memperbaharui serta membaca data yang tersimpan di dalam *Database*. Contoh perintah SQL untuk DML: SELECT, INSERT, DELETE, dan UPDATE.

3. *Data Control Language* (DCL)

Data Control Language (DCL) adalah jenis instruksi SQL yang berkaitan dengan manajemen hak akses dan pengguna (user) yang dapat mengakses *Database* maupun tabel. Termasuk diantaranya: GRANT dan REVOKE.

2.11 *Crystal report 8.5*

Menurut Andri Kuniyo dan Kusriani “*Crystal report* merupakan program yang dapat digunakan untuk membuat, menganalisis dan menterjemahkan informasi yang terkandung dalam *Database* atau program ke dalam berbagai jenis laporan yang sangat fleksibel” (Fauzi, 2013)

Crystal reports dirancang untuk membuat laporan yang dapat digunakan dengan berbagai bahasa pemrograman berbasis Windows, seperti Visual Basic, Visual C/C++, Visual Interdev, dan Borland Delphi.

Beberapa Kelebihan dari *Crystal report* adalah sebagai berikut :

1. Pembuatan laporan dengan *Crystal report* tidak terlalu rumit dan banyak melibat-kan kode program.
2. Program *Crystal reports* banyak digunakan karena mudah terintegrasi dengan bahasa lain.
3. Fasilitas impor hasil laporan yang mendukung *format-format* paket program lain, se-perti *Microsoft Office*, *Adobe Acrobat Reader*, *HTML*, dan sebagainya.
4. Koneksi yang mudah karena disertai beberapa *form* yang memudahkan koneksi.

2.12 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. *Black Box* testing berusaha untuk menemukan kesalahan dalam kategori berikut:

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

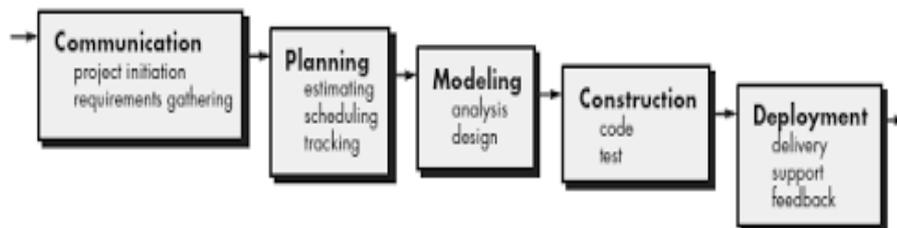
Tes ini dirancang untuk menjawab beberapa pertanyaan-pertanyaan berikut ini:

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?

2.13 System Development Life Cycle (SDLC)

“System Devopment Life Cycle (SDLC) adalah suatu kerangka yang menggambarkan kegiatan yang dilakukan pada setiap tahap pembuatan sebuah *software*” (Al Fatta, 2007:24). Model SDLC yang dipakai dalam penelitian ini adalah model *waterfall*. *Waterfall Model* atau *Classic Life Cycle* merupakan model yang paling banyak dipakai dalam *software enginnering* (SE).



. 1 Tahapan Metode *Waterfall*

Menurut Pressman Roger. S (2010) menjelaskan bahwa :

Waterfall adalah metode air terjun atau yang sering disebut metode *waterfall* sering dinamakan siklus hidup klasik (*classic life cycle*), dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan – tahapan perencanaan (*planning*), permodelan (*modeling*), konstruksi (*construction*), serta penyerahan sistem ke para pelanggan/pengguna (*deployment*), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan. Tahapan metode *waterfall* dapat dilihat pada gambar di bawah ini.

Dalam pengembangannya metode *waterfall* memiliki beberapa tahapan yang berurut yaitu *communication, planning, modelling, construction, and deployment*.

Tahapan –tahapan dari metode *waterfall* adalah sebagai berikut :

1. *Communication*

Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi ini biasanya dapat diperoleh melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

2. *Planning*

Proses *planning* merupakan lanjutan dari proses *communication (analysis requirement)*. 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.

3. *Modeling*

Proses *modelling* ini akan menerjemahkan syarat kebutuhan ke sebuah perancangan *software* yang dapat diperkirakan sebelum dibuat *coding*. Proses ini berfokus pada rancangan struktur data, arsitektur *software*, representasi *interface*, dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut *software requirement*.

4. *Construction*

Construction merupakan proses membuat kode. *Coding* atau pengkodean merupakan penerjemahan desain dalam bahasa yang bisa kita kenali oleh komputer. *Programmer* akan menerjemahkan transaksi yang diminta oleh *user*. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu *software*, artinya penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah dibuat tadi. Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap sistem tersebut untuk kemudian bisa diperbaiki.

5. *Deployment*

Tahapan ini bisa dikatakan final dalam pembuatan sebuah *software* atau sistem. Setelah melakukan analisis, desain dan pengkodean maka aplikasi yang sudah jadi akan digunakan oleh *user*. Kemudian *software* yang telah dibuat harus dilakukan pemeliharaan secara berkala.