

BAB II

LANDASAN TEORI

2.1 Rumah Sakit

Menurut Peraturan Menteri Kesehatan (PERMENKES) Republik Indonesia Nomor 340/Menkes/Per/III/2010-Pasal 1 (1), Rumah sakit merupakan instansi pelayanan kesehatan yang menyelenggarakan pelayanan kesehatan perorangan secara paripurna, diantaranya menyediakan pelayanan tentang rawat inap, rawat jalan, dan gawat darurat.

2.2 Rawat Jalan

Menurut Keputusan Menteri Kesehatan Republik Indonesia Nomor 560/MENKES/SK/IV/2003 tentang pola tarif perjan rumah sakit, rawat jalan merupakan pelayanan pasien untuk observasi, diagnosis, pengobatan, rehabilitasi medis dan pelayanan kesehatan lainnya tanpa diharuskan untuk menginap di rumah sakit.

Menurut Rachman dan Nastiti (2010) tujuan kunjungan rawat jalan banyak dan fokusnya akan bervariasi antara satu kunjungan dengan kunjungan lain:

- a. **Diagnosis dan pengobatan penyakit** merupakan unsur penting pada setiap kunjungan. Dokter harus mengevaluasi status penyakit kronik dan mencari adanya masalah baru. Karena terbatasnya waktu tidak semua masalah dapat dianalisis secara mendalam. Sering kali diperlukan jadwal kunjungan tambahan untuk menelusuri semua masalah dan keluhan secara lebih mendalam.

- b. **Penjelasan dan penghilangan gejala** merupakan kunci kepuasan pasien. Keluhan utama pasien seharusnya jangan pernah dilupakan. Penghilangan gejala tidak selalu memerlukan terapi obat, tetapi dapat dengan terapi panas, es, fisik, dan perubahan pola makan atau aktivitas. Apa yang dicari pasien adalah keyakinan bahwa gejala yang dimiliki tidak menggambarkan penyakit yang serius (misal, bahwa *irritable bowel syndrome* dapat menyebabkan rasa sakit, tetapi tidak akan menjadi kanker).
- c. **Penapisan untuk penyakit asimtomatik** dilakukan pada stadium awal yang masih dapat diobati. Dokter seharusnya menyimpan kartu berobat untuk menjaga catatan jadwal uji penapisan yang meliputi penapisan untuk kanker yang sering terjadi, hipertensi, hiperkolesterolemia, osteoporosis, dan lain-lain.
- d. **Pencegahan penyakit** dapat dilakukan melalui imunisasi, modifikasi faktor risiko (misal, pengobatan hipertensi) dan pendidikan pasien. Pasien mencari saran yang dapat dipercaya mengenai gaya hidup sehat, termasuk diet, olahraga, kebiasaan tidur, dan seterusnya.
- e. **Hubungan pasien-dokter yang berkesinambungan** memberikan kesempatan untuk mengenali semua masalah pasien dan bagaimana masalah tersebut saling berhubungan serta memahami kepribadian dan keadaan hidup pasien. Jika dokter benar-benar bersedia mendengarkan pasien, kepercayaan dan saling pengertian akan tumbuh bersamaan dengan berkembangnya hubungan terapi. Pasien akan mengatakan bahwa hanya dengan berbicara kepada dokternya membuat mereka merasa lebih baik.

2.3 Rekam Medis

Peraturan Menteri Kesehatan Republik Indonesia Nomor 55 Tahun 2013 Tentang Penyelenggaraan Pekerjaan Perekam Medis, rekam medis merupakan berkas yang berisi catatan dan dokumen tentang identitas pasien, pemeriksaan, pengobatan, tindakan dan pelayanan lain kepada pasien pada fasilitas pelayanan kesehatan. Menurut PERMENKES Nomor 269 Tahun 2008, isi dari pada rekam medis untuk unit rawat jalan pada sarana pelayanan kesehatan sekurang-kurangnya memuat:

- a. Identitas pasien
- b. Tanggal dan waktu
- c. Hasil anamnesis, mencakup sekurang-kurangnya keluhan dan riwayat penyakit
- d. Hasil pemeriksaan fisik dan penunjang medis
- e. Diagnosis
- f. Rencana penatalaksanaan
- g. Pengobatan dan/atau tindakan
- h. Pelayanan lain yang telah diberikan kepada pasien
- i. Persetujuan tindakan bila diperlukan

Menurut Rustiyanto (2012), rekam medis merupakan keterangan tertulis ataupun yang terekam tentang identitas, anamnesa penentuan fisik laboratorium, diagnosa segala pelayanan dan tindakan medis yang diberikan kepada pasien, baik pemeriksaan atau pengobatan yang dirawat jalan, rawat inap sampai pelayanan gawat darurat. Sedangkan rekam medis elektronik adalah rekaman kesehatan yang

di komputerisasikan mulai dari mengumpulkan, mengolah, menganalisis dan mempresentasikan data yang berhubungan dengan kegiatan pelayanan kesehatan.

2.3.1 Tujuan Rekam Medis

Dibuatnya rekam medis bertujuan untuk menunjang tercapainya tertib administrasi dalam rangka upaya peningkatan pelayanan kesehatan di rumah sakit. Tertib administrasi merupakan salah satu faktor yang menentukan upaya pelayanan kesehatan rumah sakit. Maka dari itu dukungan pengolahan rekam medis yang baik dan benar akan membuat tertib administrasi di rumah sakit berhasil sebagaimana mestinya.

Pembuatan rekam medis di rumah sakit juga bertujuan untuk mendapatkan catatan atau dokumen dari pasien yang kuat dan akurat, mengenai riwayat kesehatan, riwayat penyakit dimasa lalu sampai sekarang, serta segala pengobatan yang pernah diberikan sebagai upaya peningkatan kesehatan di rumah sakit.

2.3.2 Kegunaan Rekam Medis

Secara umum, kegunaan rekam medis antara lain sebagai berikut:

1. Sebagai alat komunikasi antara dokter dengan tenaga ahlinya yang ikut ambil bagian didalam memberikan pelayanan pengobatan, perawatan terhadap pasien.
2. Sebagai dasar perencanaan pengobatan/perawatan yang akan diberikan kepada pasien.
3. Sebagai bukti tertulis atas segala tindakan pelayanan, perkembangan penyakit, dan pengobatan selama pasien berkunjung/dirawat di rumah sakit.

4. Sebagai bahan yang digunakan untuk analisa, penelitian, dan evaluasi terhadap kualitas pelayanan yang diberikan kepada pasien.
5. Melindungi kepentingan hukum bagi pasien, rumah sakit, dokter dan tenaga kesehatan lainnya.
6. Menyediakan data-data khusus guna untuk penelitian dan pendidikan.
7. Sebagai dasar didalam perhitungan biaya pembayaran pelayanan medik pasien.
8. Menjadi sumber ingatan yang harus didokumentasikan, serta sebagai bahan pertanggung jawaban dan laporan.

2.4 Dokter

Menurut undang-undang nomor 29 tahun 2004 tentang praktik kedokteran, dokter dan dokter gigi adalah dokter, dokter spesialis, dokter gigi, dan dokter gigi spesialis lulusan pendidikan kedokteran atau kedokteran gigi baik di dalam maupun di luar negeri yang diakui oleh Pemerintah Republik Indonesia sesuai dengan peraturan perundang-undangan. Dokter merupakan seorang ahli medis yang dapat melakukan pemeriksaan atau tindakan medis untuk menangani/menyembuhkan penyakit. Dokter juga memiliki kewajiban dalam melaksanakan praktik kedokteran, sebagaimana yang sudah diatur pada pasal 51 undang-undang nomor 29 tahun 2004, sebagai berikut:

- a. Memberikan pelayanan medis sesuai dengan standar profesi dan standar prosedur operasional serta kebutuhan medis pasien;
- b. Merujuk pasien ke dokter atau dokter gigi lain yang mempunyai keahlian atau kemampuan yang lebih baik, apabila tidak mampu melakukan suatu pemeriksaan atau pengobatan;

- c. Merahasiakan segala sesuatu yang diketahuinya tentang pasien, bahkan juga setelah pasien itu meninggal dunia;
- d. Melakukan pertolongan darurat atas dasar perikemanusiaan, kecuali bila ia yakin ada orang lain yang bertugas dan mampu melakukannya;
- e. Menambah ilmu pengetahuan dan mengikuti perkembangan ilmu kedokteran atau kedokteran gigi.

Selain itu, dokter juga memiliki hak dalam melaksanakan praktik kedokteran, sesuai aturan pada pasal 50 undang-undang nomor 29 tahun 2004, yaitu:

- a. Memperoleh perlindungan hukum sepanjang melaksanakan tugas sesuai dengan standar profesi dan standar prosedur operasional;
- b. Memberikan pelayanan medis menurut standar profesi dan standar prosedur operasional;
- c. Memperoleh informasi yang lengkap dan jujur dari pasien atau keluarganya;
- d. Menerima imbalan jasa.

2.5 Pasien

Menurut Kamus Besar Bahasa Indonesia (2015), pasien adalah penderita, yaitu orang sakit yang dirawat oleh dokter untuk menerima/mendapat perawatan medis. Sedangkan, menurut undang-undang nomor 29 tahun 2004 tentang praktik kedokteran, Pasien adalah setiap orang yang melakukan konsultasi masalah kesehatannya untuk memperoleh pelayanan kesehatan yang diperlukan baik secara langsung maupun tidak langsung kepada dokter atau dokter gigi. Adapun kewajiban pasien dalam menerima pelayanan pada praktik kedokteran, sesuai dengan aturan pada pasal 52 undang-undang nomor 29 tahun 2004:

- a. Memberikan informasi yang lengkap dan jujur tentang masalah kesehatannya;
- b. Mematuhi nasihat dan petunjuk dokter atau dokter gigi;
- c. Mematuhi ketentuan yang berlaku di sarana pelayanan kesehatan;
- d. Memberikan imbalan jasa atas pelayanan yang diterima.

Selain itu, pasien juga memiliki hak dalam pelayanan praktik kedokteran, sebagaimana diatur dalam pasal 52 undang-undang nomor 29 tahun 2004, yaitu:

- a. Mendapatkan penjelasan secara lengkap tentang tindakan medis;
- b. Meminta pendapat dokter atau dokter gigi lain;
- c. Mendapatkan pelayanan sesuai dengan kebutuhan medis;
- d. Menolak tindakan medis;
- e. Mendapatkan isi rekam medis.

2.6 Database

Menurut Raharjo (2011) *database* adalah kumpulan data yang terintegrasi dan diatur sedemikian rupa hingga data tersebut dapat dimanipulasi, diambil, dan dicari secara cepat. Isi dari *database* adalah metadata, yaitu data yang menjelaskan tentang struktur dari data itu sendiri. Data dalam *database* sendiri diklarifikasikan jenisnya dan disimpan didalam wadah tersendiri, yang disebut tabel. Tabel adalah suatu entitas yang tersusun atas kolom (*field*) dan baris (*record*). Di dalam *database* juga mengenal *constraint*, yaitu aturan atau batasan yang didefinisikan nilai atau data yang dapat disimpan di dalam *database*, antara lain melalui operasi *INSERT*, *UPDATE* dan *DELETE*. *Constraint* dibedakan menjadi empat, antara lain:

a. *Primary key*

Primary key adalah aturan untuk memastikan bahwa setiap baris data didalam suatu tabel bersifat unik (berbeda antara baris yang satu dengan yang lainnya), *primary key* diterapkan pada kolom tabel untuk dijadikan sebagai pembeda antar baris tabel.

b. *Foreign key*

Foreign key berguna untuk mendefinisikan kolom-kolom pada suatu tabel yang nilainya mengacu ke tabel lain. Dimana nilai yang diisikan ke dalam kolom *foreign key* pada suatu tabel harus diacu/diambil dari kolom *primary key* pada tabel lain. Karena jika memasukkan nilai ke dalam kolom *foreign key* dimana nilai tersebut tidak terdapat pada tabel yang diacu, maka *server database* akan menolak cara menampilkan pesan kesalahan. Aturan dalam pendefinisian *foreign key*:

- Satu tabel dapat memiliki lebih dari satu *foreign key*
- Kolom yang diacu harus didefinisikan sebagai *primary key* atau *unique*
- *Foreign key* tidak bersifat unik.

c. *Unique*

Unique sama seperti *primary key*, yaitu untuk memastikan bahwa setiap baris data yang terdapat dalam suatu tabel bersifat unik (tidak sama).

d. *Check*

Check berfungsi untuk membatasi nilai-nilai yang dapat dimasukkan ke dalam suatu kolom didalam tabel. Contohnya, kolom 'JENIS_KELAMIN' nilainya dapat kita batasi, misalnya hanya berupa 'PRIA' dan 'WANITA' saja. Selain nilai tersebut, *server database* akan menolak.

2.7 MySQL

Menurut Nugroho (2014), MySQL adalah *software* atau program aplikasi *database*, yaitu *software* yang dipakai untuk menyimpan data berupa informasi teks dan juga angka. Dengan menggunakan *database* MySQL, maka data yang tersimpan didalam *database* dapat diakses secara bersamaan menggunakan beberapa komputer yang berbeda, konsep ini sering disebut dengan sistem *multy user*. Sedangkan untuk konsep sistem *client/server*. *Database* MySQL diinstal pada sebuah komputer pusat (*server*) yang tersimpan diruang *server*, kemudian datanya diakses melalui komputer yang diinstal dengan program *client* yang letaknya berbeda dengan tempat *server*. *Database* MySQL adalah *software* yang tidak memiliki *interface* (tampilan antarmuka) visual. Perintah dasar MySQL adalah SQL (*Structure Query Language*) dimana dapat diakses menggunakan DOS (*Disk Operation System*) *Prompt* atau *Command prompt* perintah dasar MySQL disebut dengan perintah *Data Definition Statement* (DDS), didalamnya terdapat 3 perintah utama, yaitu:

- *Create* Perintah untuk membuat *database* baru, membuat tabel dan struktur datanya.
- *Alter* Perintah untuk memperbaiki nama *database*, nama tabel dan struktur datanya.
- *Drop* Perintah untuk menghapus nama *database* dan nama tabel.

Sedangkan untuk penampilan data, memperbaiki data dan menghapus data. Perintah SQL yang digunakan adalah perintah *Data Manipulation Statement* (DML), didalamnya ada 4 perintah utama, yaitu:

- *Insert* Perintah untuk menyimpan data atau memasukkan data ke dalam tabel.

- *Select* Perintah untuk menampilkan data atau mengambil data yang ada didalam tabel.
- *Update* Perintah untuk memperbaiki data atau mengubah data yang sudah ada didalam tabel.
- *Delete* Perintah untuk menghapus data yang ada didalam tabel.

Menurut Raharjo (2011) MySQL adalah *software* RDBMS (atau server *database*) yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak pengguna (*multi-user*) dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-threaded*).

Ada beberapa kelebihan pada MySQL:

- **Fleksibel:** MySQL dapat digunakan untuk mengembangkan aplikasi desktop maupun aplikasi *web*, baik menggunakan PHP, JSP, Java, Delphi, C++ maupun yang lainnya.
- **Performa Tinggi:** MySQL mempunyai mesin *query* dengan performa tinggi, yang bisa membuat proses transaksional menjadi lebih cepat.
- **Lintas Platform:** MySQL bisa digunakan pada platform (dalam hal ini Sistem Operasi) yang beragam, bisa menggunakan Microsoft Windows, Linux, atau UNIX. Dari hal tersebut, proses migrasi data (bila dibutuhkan) antar sistem operasi bisa dilakukan lebih mudah.
- **Gratis:** MySQL dapat digunakan atau bisa di unduh secara gratis tanpa biaya, tapi ada juga *software* MySQL yang bersifat komersial, yang tentunya juga memiliki tambahan kemampuan spesifik dan mendapat pelayanan dari *technical support*.

- Proteksi Data yang Handal: MySQL menyediakan mekanisme yang canggih untuk menangani perlindungan terhadap keamanan data, dengan menyediakan fasilitas manajemen *user*, enkripsi data, dan lain sebagainya.
- Komunitas Luas: MySQL memiliki komunitas yang luas dikarenakan penggunaanya banyak. Komunitas ini berguna sebagai pemberi solusi apabila mendapatkan permasalahan dalam penggunaan MySQL dengan cara melalui atau mengikuti forum.

2.8 PHP Hypertext Preprocessor

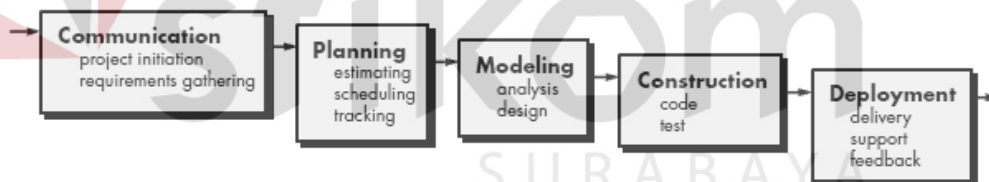
Menurut Nugroho (2013) *PHP Hypertext Preprocessor* (PHP) merupakan bahasa pemrograman berbasis *web* atau bahasa program yang digunakan untuk membuat aplikasi berbasis *web* (*Website*, blog atau aplikasi web). PHP merupakan bahasa program yang bisa dijalankan di sisi server atau sering disebut dengan “*side server language*”. Jadi program yang dibuat dengan kode PHP tidak bisa berjalan kecuali dia dijalankan pada server *web*, tanpa adanya server *web* yang terus berjalan dia tidak akan bisa dijalankan.

Karena berbasis *web*, maka aplikasi yang dibuat dengan PHP adalah berbasis *web*. Hasil dari PHP adalah berbentuk *web*, baik itu *website* (*web* pribadi, *web* portal berita, *web profile* perusahaan, dll). Hasilnya bisa dilihat dengan menjalankan *web browser*. PHP tidak ada aplikasi/*software* nyata yang bisa kita jalankan untuk menuliskan kodenya, butuh aplikasi editor untuk membuat kode program PHP, contohnya seperti editor Notepad++ atau editor lainnya. PHP adalah bahasa programnya, dalam membuat *website* ataupun aplikasi berbasis *web*, bukan hanya kode PHP saja yang kita butuhkan tetapi juga menggunakan kode HTML (*Hyper Text Markup Language*) untuk desain tampilan, yaitu

mengatur teks, tabel dan juga membuat form. Selain itu butuh CSS (*Cascading Style Sheets*) sebagai kode pemanis *web*, yang bisa jadi pengganti HTML. Jadi, dalam membuat *web* pasti menggunakan kode HTML, CSS dan PHP.

2.9 Tahapan Software Development Life Cycle (SDLC)

Menurut Pressman (2015) nama lain dari model *Waterfall* adalah model Air Terjun kadang dinamakan siklus hidup klasik (*classic life cycle*), dimana hal ini menyiratkan pendekatan yang sistematis dan berurutan (sekuensial) pada pengembangan perangkat lunak. Pengembangan perangkat lunak dimulai dari spesifikasi kebutuhan pengguna, berlanjut ke tahapan-tahapan perencanaan (*planning*), pemodelan (*modelling*), konstruksi (*construction*), serta penyerahan sistem perangkat lunak ke para pelanggan/pengguna (*deployment*), yang diakhiri dengan dukungan berkelanjutan pada perangkat lunak yang dihasilkan.



Gambar 2.1 Model *Waterfall*

Gambar 2.1 menunjukkan tahapan umum dari model proses *waterfall*. Model ini disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Akan tetapi, Pressman (2015) memecahkan model ini meskipun secara garis besar sama dengan tahapan-tahapan model *waterfall* pada umumnya.

Model ini merupakan model yang paling banyak dipakai dalam *Software Engineering*. Model ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap *communication*, *planning*, *modelling*, *construction*, dan *deployment*. Berikut ini adalah penjelasan dari tahap-tahap yang dilakukan didalam model *Waterfall* menurut Pressman (2015).

1. *Communication*

Tahap pertama, pihak pengembang akan melakukan pengumpulan data kebutuhan-kebutuhan pelanggan/pengguna. Pada tahap ini pengembang dapat mengetahui sistem seperti apa yang akan dibuat.

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. *Modelling*

Pada proses *modeling* ini menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat/dilakukan *coding*. Proses ini berfokus pada rancangan struktur data, arsitektur *software*, representasi *interface*, dan detail (algoritma) prosedural.

4. *Construction*

Construction merupakan proses membuat kode (*code generation*). Coding atau pengkodean merupakan penerjemah desain dalam sebuah bahasa yang bisa dikenali oleh komputer. *Programmer* akan menerjemahkan transaksi yang diminta oleh *user*/pengguna. Tahapan inilah yang merupakan tahapan secara

nyata dalam mengerjakan suatu *software*, artinya *user*/pengguna komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai, maka akan dilakukan testing terhadap sistem yang telah dibuat. Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap sistem tersebut, kemudian diperbaiki sesuai dengan kesalahan yang ditemukan dan kebutuhan yang diinginkan.

5. *Deployment*

Setelah semua tahap selesai dan perangkat lunak dinyatakan tidak terdapat kesalahan, pada tahap ini dilakukan implementasi (instalasi), pemeliharaan perangkat lunak dan *feedback* dari pelanggan.

2.10 *Testing*

Pengertian *testing* menurut Nidhra & Dondeti (2012), pengujian *software* merupakan suatu prosedur atau teknik yang digunakan untuk verifikasi dan validasi kualitas suatu *software*, dimana *software* akan dieksekusi untuk menemukan kesalahan yang ada didalamnya. Sedangkan menurut Quadri & Farooq (2010), pengujian *software* adalah proses verifikasi dan validasi apakah sebuah aplikasi *software* atau program memenuhi persyaratan bisnis dan persyaratan teknis yang mengarahkan desain dan pengembangan serta cara kerjanya seperti yang diharapkan, juga untuk mengidentifikasi kesalahan yang penting yang digolongkan berdasarkan tingkat *severity* pada aplikasi yang harus diperbaiki.

Dari pendapat-pendapat tersebut dapat disimpulkan bahwa pengujian *software* merupakan proses verifikasi dan validasi sebuah *software* apakah sudah

memenuhi kebutuhan dan untuk mengidentifikasi kesalahan-kesalahan yang ditemukan saat eksekusi program.

2.10.1 Behavioral (*Black-Box*) Testing

Menurut Nidhra & Dondeti (2012), *black-box testing* juga disebut dengan *functional testing*, sebuah teknik pengujian fungsional yang merancang *test case* berdasarkan informasi dari spesifikasi. Menurut Pressman (2015), pengujian *Black-Box* berfokus pada persyaratan fungsional perangkat lunak. Pengujian *Black-Box* memungkinkan perekrut perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.

Pengujian *Black-Box* berusaha menemukan kesalahan dalam kategori:

1. Fungsi-fungsi yang tidak benar atau hilang,
2. Kesalahan *interface*,
3. Kesalahan dalam struktur data atau akses *database* eksternal,
4. Kesalahan kinerja,
5. Inisialisasi dan kesalahan terminasi.