

BAB II

LANDASAN TEORI

Landasan teori dan tinjauan pustaka yang memiliki relevansi terhadap masalah yang akan dibahas perlu diberikan dalam penyusunan laporan Tugas Akhir ini. Landasan teori didapatkan dari beberapa *literature* yang mempublikasikan pendapat beberapa ilmuwan yang digunakan sebagai penunjang dalam pembahasan masalah. Tinjauan pustaka didapatkan dari beberapa penelitian yang sudah muncul sebelumnya.

2.1. Tinjauan Pustaka

Lestari, dkk (2011) menjelaskan bahwa dengan adanya sistem komputerisasi proses pelayanan pada rumah sakit bersalin Graha Rap Tanjung Balai Karimun lebih cepat dan akurat, proses pencarian data lebih mudah, serta sistem dapat membuat laporan secara otomatis. Tampilan sistem informasi rekam medis rumah sakit bersalin Graha Rap Tanjung Balai Karimun disajikan pada Gambar 2.1.



Gambar 2.1. Tampilan Sistem Informasi Rekam Medis Rumah Sakit Bersalin Graha Rap Tanjung Balai Karimun

Mukti, dkk (2013) menjelaskan bahwa sistem informasi pelayanan rawat jalan berbasis komputer dapat mempercepat dan mempermudah pelayanan rawat jalan, karena semua kegiatan pelayanan rawat jalan tersedia di dalam sistem. Segi keamanan,

ketepatan, dan penghematan waktu, sistem pelayanan rawat jalan berbasis komputer menunjukkan keunggulan dalam pengolahan data, penyimpanan, dan penyajian data dibandingkan dengan sistem lama.

Puspitasari, dkk (2013) menjelaskan bahwa sistem informasi ini dapat mempermudah petugas Puskesmas Pringkuku Kabupaten Pacitan dalam mengelola data rekam medis pasien sehingga dapat meminimalisir kesalahan yang terjadi jika data rekam medis tersebut dibutuhkan ataupun dicari. Tampilan aplikasi *web* rekam medis disajikan pada Gambar 2.2.

UNIVERSITAS SAHID USAHID

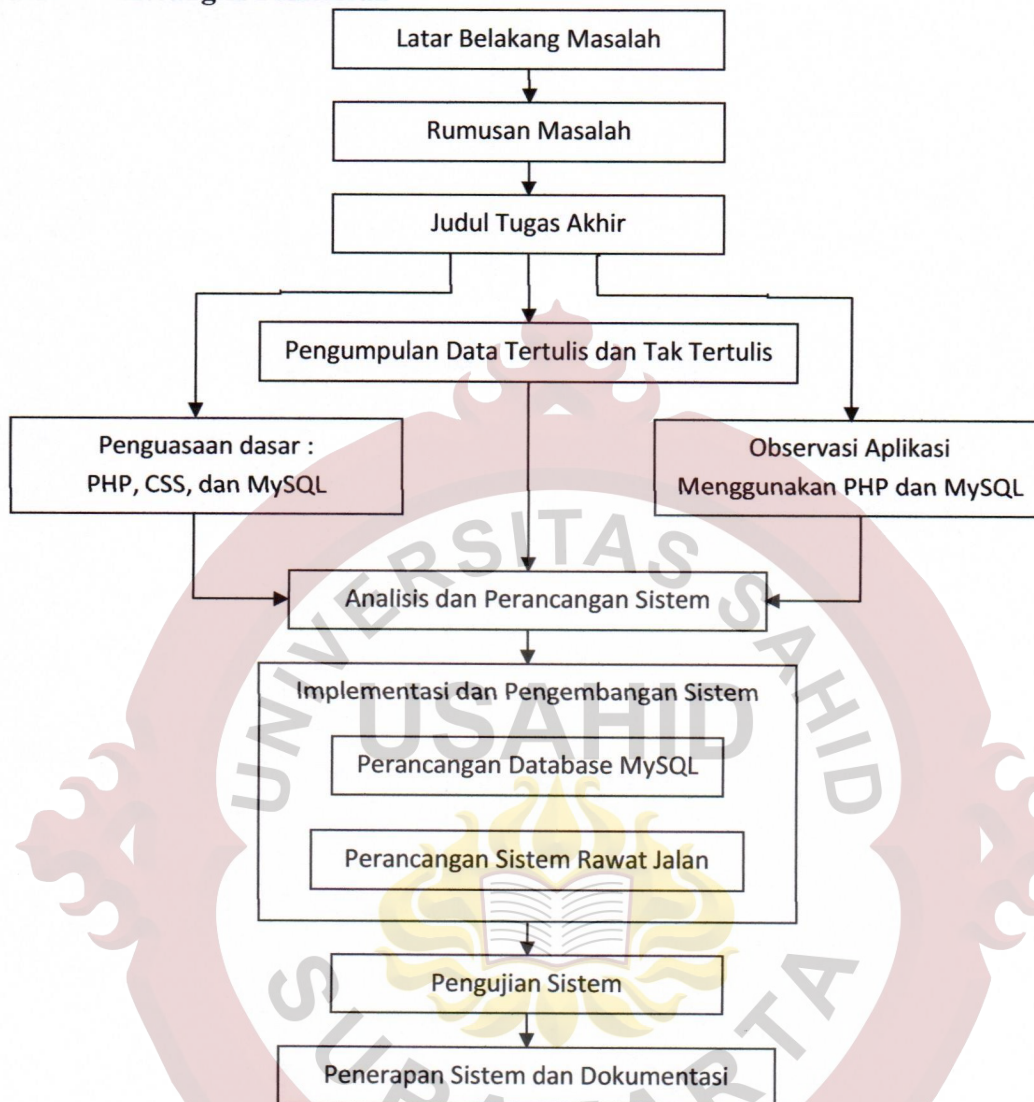
No	Nama Lengkap	Alamat	Tanggal & Jam Masuk	Keluhan	Riwayat Penyakit	No. Rawat	Umur
1	Adi Pratomo	Jember	12 April 2018	Demam	Diabetes	0000000001	35
2	Budi Santoso	Jember	12 Juli 2018	Demam	Diabetes	0000000002	40
3	Citra Dewi	Jember	12 Agustus 2018	Demam	Diabetes	0000000003	30
4	Dani Kusuma	Jember	12 Februari 2018	Demam	Diabetes	0000000004	38
5	Eka Sari	Jember	12 Januari 2018	Demam	Diabetes	0000000005	32
6	Fani Kusuma	Jember	12 Juli 2018	Demam	Diabetes	0000000006	35
7	Gani Kusuma	Jember	12 Agustus 2018	Demam	Diabetes	0000000007	38
8	Hani Kusuma	Jember	12 Agustus 2018	Demam	Diabetes	0000000008	35
9	Iani Kusuma	Jember	12 Agustus 2018	Demam	Diabetes	0000000009	35
10	Jani Kusuma	Jember	12 Agustus 2018	Demam	Diabetes	0000000010	35

Gambar 2.2. Tampilan Aplikasi *Web* Rekam Medis

Widya dan Migunani (2014) menjelaskan bahwa sistem informasi rekam medis pada rumah perawatan *psiko-neuro-geriatric* Puri Saras Semarang dapat meminimalisir kehilangan data rekam medis pasien yang disebabkan oleh penyimpanan data yang masih di simpan dalam rak, mempermudah dan mempercepat pekerjaan petugas administrasi, serta sistem dapat mencetak data yang dapat digunakan sebagai cadangan data ketika komputer mengalami kerusakan.

Syukron dan Hasan (2015) menjelaskan bahwa sistem informasi rawat jalan merupakan sistem yang baik untuk digunakan karena dapat mewujudkan pelayanan kesehatan yang bermutu, efektif, dan efisien serta dapat meningkatkan kinerja Puskesmas Winong. Sistem rawat jalan ini juga dapat mempermudah kerja bagian administrasi dalam membuat laporan.

2.2. Kerangka Pemikiran



Gambar 2.3. Diagram Kerangka Pemikiran

Keterangan Diagram Kerangka Pemikiran (Gambar 2.3) :

1. Latar Belakang Masalah

Identifikasi adanya masalah yang menjadi sebuah permasalahan pada klinik Cahaya Sehat Dawung Surakarta. Permasalahan yang dibahas merupakan permasalahan yang menjadi kendala bagi klinik dan dari pihak klinik belum memiliki solusi untuk menyelesaikan masalah tersebut.

2. Rumusan Masalah

Solusi yang baik untuk menyelesaikan sebuah permasalahan yang ada pada Klinik Cahaya Sehat Dawung Surakarta. Solusi yang diberikan dapat berupa sebuah sistem informasi.

3. Judul Tugas Akhir

Judul yang sekiranya tepat untuk menangani permasalahan yang ada dan di ambil pada Klinik Cahaya Sehat Dawung Surakarta.

4. Pengumpulan Data Tertulis dan Tak Tertulis

Semua data yang dibutuhkan dikumpulkan, baik melalui *interview* dengan petugas, observasi atau studi literatur di Klinik Cahaya Sehat Dawung Surakarta.

5. Penguasaan Dasar (PHP, CSS, dan MySQL)

Aplikasi sederhana dibuat sebagai percobaan dengan tujuan agar dapat lebih menguasai bahasa pemrograman PHP, CSS, dan *database* MySQL sehingga hasilnya lebih maksimal.

6. Observasi Aplikasi

Aplikasi yang sudah ada dilakukan pengamatan, baik dari karya ilmiah, buku atau internet sehingga dapat dijadikan referensi untuk membangun sebuah aplikasi.

7. Analisis dan Perancangan Sistem

Sistem yang akan di bangun dianalisis dan dirancang seperti apa, bagaimana desainnya, dan apa saja isinya, sehingga sistem ini dapat membantu memecahkan permasalahan yang ada pada Klinik Cahaya Sehat Surakarta.

8. Implementasi dan Pengembangan Sistem

a. Perancangan *database* MySQL

Database dari data-data yang telah didapatkan dibuat sesuai dengan kebutuhan sistem.

b. Perancangan aplikasi pengolahan data

Aplikasi dibuat dengan dasar *database* yang telah selesai dibuat.

9. Pengujian Sistem

Pengujian sistem ini dilakukan bertujuan untuk mengetahui jika ternyata

masih ada kesalahan ataupun kekurangan pada sistem yang dibuat.

10. Penerapan Sistem dan Dokumentasi

Sistem telah siap digunakan setelah melewati tahap pengujian dan membuat dokumentasi dari keseluruhan kegiatan penyusunan tugas akhir.

2.3. Landasan Teori

2.3.1. Pengertian Rawat Jalan

Surat Keputusan Menteri RI No.560/MENKES/SK/IV/2003 tentang tarif per jam rumah sakit bahwa Pelayanan Rawat Jalan adalah pelayanan pasien untuk observasi, diagnosis, pengobatan, rehabilitasi medik dan pelayanan kesehatan lainnya tanpa menginap di rumah sakit.

2.3.2. Pengertian Klinik

Peraturan Menteri Kesehatan Republik Indonesia Nomor 028/MENKES/PER/I/2011 tentang klinik, Klinik adalah fasilitas pelayanan kesehatan yang menyelenggarakan pelayanan kesehatan perorangan yang menyediakan pelayanan medis dasar dan/atau spesialisik, diselenggarakan oleh lebih dari satu jenis tenaga kesehatan dan dipimpin oleh seorang tenaga medis.

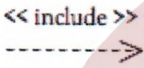
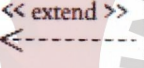
2.3.3. Analisis dan Perancangan Berbasis Objek

Menurut Hermawan (2004), *Object Oriented Programming* atau Pemrograman Berorientasi Objek adalah sebuah rancangan dalam pemrograman yang diutamakan pada penciptaan kelas yang merupakan abstraksi/*blueprint/prototype*. Analisis dan perancangan metodologi berorientasi objek menggunakan diagram *Unified Modeling Language* (UML).

2.3.3.1. Use Case Diagram

Use case Diagram menjelaskan apa yang dilakukan oleh sistem yang akan di bangun dan siapa yang akan berinteraksi dengan sistem. Sebuah *use case* digambarkan sebagai *elips* horizontal dalam suatu diagram UML *use case*. Simbol yang digunakan pada *use case diagram* ditampilkan pada Tabel 2.1.

Tabel 2.1 Simbol *Use case Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Segala sesuatu yang berinteraksi dengan sistem aplikasi komputer, bisa berupa orang atau perangkat keras, atau obyek lain dalam sistem. <i>Actor</i> memberikan informasi dan atau memerintahkan sistem melakukan sesuatu.
2		<i>Association</i>	Jalur komunikasi yang menghubungkan antara <i>actor</i> dengan <i>use case</i> .
3		<i>Use Case</i>	Urutan aksi-aksi yang dikerjakan sistem untuk menghasilkan suatu hasil yang terukur bagi suatu aktor.
4		<i>Include</i>	Relasi <i>use case</i> tambahan sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.
5		<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan tersebut.

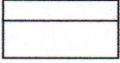
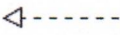
2.3.3.2. *Class Diagram*

Class diagram memperlihatkan hubungan antar *class* yang di bangun dan menunjukkan bagaimana *class* tersebut saling berkolaborasi untuk mendapatkan suatu tujuan tertentu. Simbol yang digunakan pada *class diagram* ditampilkan pada Tabel 2.2.

2.3.3.3. *Activity Diagram*

Activity diagram menjelaskan aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*. Selain itu juga kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas. Simbol yang digunakan pada *activity diagram* ditampilkan pada Tabel 2.3.

Tabel 2.2 Simbol *Class* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan antar elemen yang lebih umum ke elemen yang lebih spesifik (<i>sub class</i>), sehingga <i>class</i> yang lebih spesifik akan menurunkan atribut dan operasi.
2		<i>Class</i>	Kumpulan dari objek yang memiliki atribut serta operasi yang sama.
3		<i>Realization</i>	Hubungan bahwa elemen yang ada dibagian tanpa arah panah akan merealisasikan apa yang dinyatakan oleh elemen yang ada pada bagian depan panah.
4		<i>Dependency</i>	Menjelaskan hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
5		<i>Association</i>	Menggambarkan navigasi antar class, berapa banyak obyek lain yang bisa berhubungan dengan satu obyek.

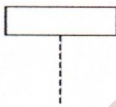
Tabel 2.3 Simbol *Activity* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Menggambarkan sebuah proses/kegiatan dari sebuah aktifitas yang terjadi.
2		<i>Initial Node</i>	Menjelaskan bagaimana objek dibentuk atau diawali.
3		<i>Activity Final Node</i>	Menjelaskan bagaimana objek dibentuk dan kemudian dihancurkan.
4		<i>Fork Node</i>	Sebuah aliran yang pada tahap tertentu berubah menjadi beberapa aliran.
5		<i>Decision node</i>	Penggambaran sebuah pilihan dalam mengambil sebuah keputusan.
6		<i>Join Node</i>	Menunjukkan bahwa beberapa aliran masukan tertentu menjadi satu aliran.

2.3.3.4. Sequence Diagram

Sequence diagram menggambarkan setiap obyek yang terlibat pada sebuah *use case* menggunakan garis putus-putus vertikal, kemudian *message* yang dikirim oleh obyek ditunjukkan dengan berupa garis horisontal secara kronologis dari atas ke bawah. Simbol yang digunakan pada *sequence diagram* ditampilkan pada Tabel 2.4.

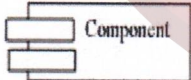
Tabel 2.4 Simbol *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Garis titik-titik yang terhubung dengan objek. Sepanjang <i>lifeline</i> terdapat <i>activation</i> yang mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
2		<i>Message</i>	Simbol dari komunikasi antar <i>class</i> yang memuat informasi-informasi tentang aktifitas yang terjadi.

2.3.3.5. Component Diagram

Component diagram menggambarkan *software* pada suatu sistem dan penerapan *software* dari satu ataupun lebih *class*. Simbol yang digunakan pada *component diagram* ditampilkan pada Tabel 2.5.

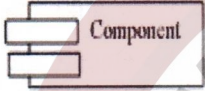
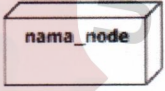
Tabel 2.5 Simbol *Component Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Component</i>	Bertujuan untuk melambangkan sebuah entitas <i>software</i> pada sebuah sistem dan digambarkan sebagai sebuah kotak segiempat dengan tambahan dua kotak kecil sejajar yang menempel dibagian kiri.
2		<i>Dependency</i>	Bertujuan untuk menggambarkan relasi antara dua komponen. Notasi yang dipakai adalah tanda panah putus-putus yang diarahkan kepada komponen tempat sebuah komponen itu bergantung.

2.3.3.6. Deployment Diagram

Deployment diagram bertujuan untuk menunjukkan tata letak suatu sistem secara fisik, selain itu dapat digunakan untuk menampilkan bagian-bagian *software* yang terdapat pada *hardware* dan juga digunakan untuk menerapkan suatu sistem dan hubungan antara komponen *hardware*. *Deployment diagram* menunjukkan letak *software* pada *hardware* yang digunakan oleh suatu sistem. Simbol yang digunakan pada *deployment diagram* ditampilkan pada Tabel 2.6.

Tabel 2.6 Simbol *Deployment Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Component</i>	Menjelaskan bahwa komponen-komponen yang ada diletakkan di dalam node untuk menentukan keberadaan posisi mereka dalam sebuah sistem.
2		<i>Node</i>	Bertujuan untuk menggambarkan bagian <i>hardware</i> yang terdapat dalam sebuah sistem. Lambang notasi untuk node digambarkan sebagai sebuah kubus 3 dimensi.
3		<i>Association</i>	Notasi <i>association</i> digambarkan sebagai sebuah garis yang menghubungkan dua node yang bertujuan untuk mengindikasikan jalur komunikasi antara komponen-komponen <i>hardware</i> .

2.3.4. Pengujian *Blackbox*

Pressman (2002), Pengujian *blackbox* merupakan pengujian yang berfokus kepada persyaratan fungsional perangkat lunak. Pengujian *blackbox* memungkinkan perekayasa perangkat lunak menghasilkan serangkaian kondisi masukan yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *blackbox* bukan merupakan alternatif dari metode *whitebox*, akan tetapi merupakan pendekatan komplementer yang kemungkinan besar dapat mengungkap kesalahan daripada metode *whitebox*.

Pengujian *blackbox* berusaha menemukan kesalahan dalam kategori sebagai berikut :

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.

Pengujian *blackbox* berbeda dari pengujian *whitebox* yang dilakukan pada awal proses pengujian, pengujian *blackbox* cenderung dilakukan selama tahap akhir pengujian. Oleh karena *blackbox testing* tidak mengabaikan struktur *control*, maka perhatian difokuskan pada domain informasi. Pengujian ini didesain untuk menjawab pertanyaan-pertanyaan sebagai berikut :

1. Bagaimana validitas fungsional diuji ?
2. Kelas masukan apa yang akan membuat *test case* menjadi baik ?
3. Apakah sistem sangat sensitif terhadap nilai masukan tertentu ?
4. Bagaimana batas-batas dari data yang terisolasi ?
5. Kecepatan data dan volume data apa yang dapat mentolerir sistem ?
6. Efek apa yang akan muncul dari kombinasi data tertentu terhadap operasi sistem ?