

## **BAB II**

### **LANDASAN TEORI DAN TINJAUAN PUSTAKA**

#### **2.1 Teori Yang Berhubungan Dengan Sistem**

##### **2.1.1 Pengertian E-Medic**

E-Medic atau yang sering dikenal dengan Electronic Medical Record merupakan kegiatan mengkomputerisasikan isi rekam medis dan proses yang berhubungan dengannya. Pada awalnya rekam medis di Indonesia dikenal dengan istilah rekam medis yang sampai saat ini pun sebagian Rumah Sakit di Indonesia masih menggunakan istilah yang sama.

Rekam Medis Kesehatan menurut Lampiran SK PB IDI No 315/PB/A.4/88 adalah rekaman dalam bentuk tulisan atau gambaran aktivitas pelayanan yang diberikan oleh pemberi pelayanan medis/kesehatan kepada seorang pasien. Berdasarkan SK Menteri Kesehatan Nomor:269/Menkes/PER/III/2008 tentang rekam medis adalah berkas yang berisikan catatan dan dokumen tentang identitas pasien, pemeriksaan, pengobatan, tindakan dan pelayanan lain yang telah diberikan kepada pasien.

Berdasarkan pengertian E-Medic maupun rekam Medis diatas dapat disimpulkan bahwa keduanya memiliki arti yang sama yaitu berkas yang berisikan catatan atau rekaman tentang identitas pasien, pemeriksaan, pengobatan, tindakan medis dan pelayanan medis lainnya terhadap pasien.

### 2.1.2 Pengertian Sistem Informasi

Menurut I Putu Agus Eka Pratama (2014 : 7) Sistem didefinisikan sebagai sekumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan suatu tugas bersama-sama guna tercapainya tujuan tertentu. Informasi merupakan hasil pengolahan data dari satu atau berbagai sumber, yang kemudian diolah, sehingga memberikan nilai, arti, dan manfaat (I Putu Agus Eka Pratama, 2014 : 8). Sedangkan Sistem informasi menurut Abdul Kadir (2014:8) sistem informasi mencakup sejumlah komponen (manusia, komputer, teknologi informasi, dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi), dan dimaksudkan untuk mencapai suatu sasaran atau tujuan.

Berdasarkan pengertian sistem dan informasi yang telah dijelaskan diatas, maka dapat dinyatakan bahwa Sistem Informasi merupakan gabungan dari empat bagian utama (I Putu Agus Eka Pratama, 2014 : 7). Ke empat bagian utama tersebut mencakup perangkat lunak, perangkat keras, infrastruktur, dan sumber daya manusia yang terlatih.

### 2.1.3 Komponen Sistem Informasi

Menurut I Putu Agus Eka Pratama (2014:10), sistem informasi merupakan gabungan dari empat bagian utama. Ke empat bagian utama tersebut mencakup perangkat lunak (*Software*), perangkat keras (*Hardware*), infrastruktur, dan Sumber Daya Manusia (SDM) yang terlatih. Komponen-komponen Pada Sistem Informasi menurut I Putu Agus Eka Pratama (2014: 11), yaitu :

1. Input (Masukkan)

2. Output (Keluaran)
3. Software (Perangkat Lunak)
4. Hardware (Perangkat Keras)
5. Database (Basis Data)
6. Kontrol dan Prosedur
7. Teknologi dan Jaringan Komputer

Elemen penting pada sistem informasi menurut I Putu Agus Eka Pratama (2014: 15), yaitu :

1. Perangkat Keras
2. Perangkat Lunak
3. Pengguna
4. Prosedur
5. Basis Data
6. Komunikasi

## **2.2 Unsur-Unsur Dalam Perancangan Website**

### **2.2.1 Pengertian Website**

Menurut Andrea Adelheid (2015:11) website adalah halaman informasi yang di sediakan melalui jalur internet sehingga bisa diakses dimanapun selama anda terkoneksi dengan jaringan internet. Sedangkan menurut Supriyanto (2005 : 355) website adalah kumpulan dokumen-dokumen agar dapat ditampilkan pada program *browser* yang umumnya digunakan adalah HTML (*Hypertext Markup Language*) dan bahasa dari WWW (*World Wide Web*) yang digunakan menyusun dokumen tersebut.

Berdasarkan kategorinya, website digolongkan menjadi 3 (tiga). Menurut Andrea Adelheid (2015: 11), yaitu :

- a. Website Statis adalah website yang mempunyai halaman tidak berubah.
- b. Website Dinamis adalah website yang secara struktur diperuntukkan untuk update sesering mungkin.
- c. Website Interaktif adalah website yang saat ini memang sedang booming. Salah satu contoh website interaktif adalah blog dan forum.

### **2.2.2 Pengertian PHP (*PHP Hypertext Preprocessor*)**

Menurut Agus Saputra dan Feni Agustin (2013:2) PHP (*PHP Hypertext Preprocessor*) merupakan suatu bahasa pemrograman yang difungsikan untuk membangun suatu *website* dinamis. PHP menyatu dengan kode HTML, maksudnya adalah beda kondisi. HTML digunakan sebagai pembangun atau pondasi dari kerangka *layoutweb*, sedangkan PHP difungsikan sebagai prosesnya, sehingga dengan adanya PHP tersebut sebuah *web* akan mudah di *maintenance*.

Sedangkan menurut M. Rudyanto Arief (2011:43) PHP (*PHP Hypertext Preprocessor*) adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman *web* yang dinamis. Karena PHP merupakan *server-side scripting* maka *sintaks* dan perintah-perintah PHP akan dieksekusi di *server* kemudian hasilnya dikirim ke *browser* dalam format HTML.

### 2.2.3 Pengertian Database

Menurut I Putu Agus Eka Pratama (2014:12) komponen basis data berfungsi untuk menyimpan data dan informasi kedalam satu atau beberapa tabel. Setiap tabel memiliki *field* masing-masing. Setiap tabel memiliki fungsi penyimpanan masing-masing, serta antartabel juga terjadi relasi (hubungan).

Sedangkan menurut Rosa A.S. dan M. Shalahuddin (2013:43) basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Dari definisi-definisi tersebut maka dapat disimpulkan bahwa basis data adalah sebuah kumpulan data-data yang tersimpan, agar nantinya dapat diolah sebaik mungkin untuk dimanfaatkan oleh pengguna.

### 2.2.4 Pengertian MySql

Menurut M. Rudyanto Arief (2011:151) *MySQL* adalah jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *web* yang menggunakan *database* sebagai sumber dan pengolahan datanya. Kepopuleran *MySQL* antara lain karena *MySQL* menggunakan *SQL* sebagai bahasa dasar untuk mengakses *database*-nya sehingga mudah untuk digunakan, kinerja *query* cepat dan mencukupi untuk kebutuhan *database* perusahaan-perusahaan skala menengah-kecil.

Sedangkan menurut Muhammad Sadeli (2014:10) *MySQL* adalah *database* yang menghubungkan *script php* menggunakan perintah *query* dan *escaps character* yang sama dengan *php*. *MySQL* mempunyai tampilan *client* yang mempermudah anda dalam mengakses *database* dengan kata sandi untuk mengijinkan proses yang bisa anda lakukan.

### 2.2.5 Pengertian DBMS (*Database Management System*)

Menurut Rosa A.S. dan M. Shalahuddin (2014:44) DBMS (*Database Management System*) atau dalam bahasa Indonesia sering disebut sebagai sistem manajemen basis data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data. Karena pentingnya data bagi suatu organisasi/perusahaan, maka hampir sebagian besar perusahaan memanfaatkan DBMS dalam mengelola data yang mereka miliki.

Sedangkan menurut M. Rudyanto Arief (2011:20) DBMS (*Database Management System*) adalah sebuah aplikasi yang menjembatani user dan data-data di dalam *database*. Dengan menggunakan DBMS *user* mampu mengolah data-data di dalam *database* secara mudah dan cepat. Arsitektur DBMS ada yang *stand alone* dan ada yang *client-server*. Untuk aplikasi *web*, jenis DBMS yang digunakan adalah yang berarsitektur *client-server*.

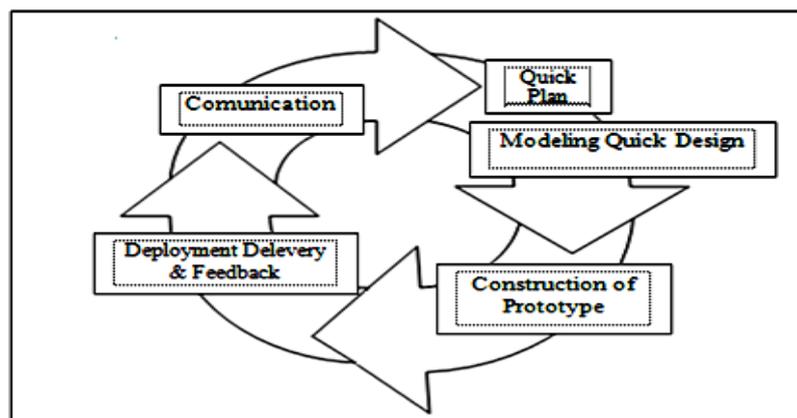
## 2.3 Metode *Prototype*

Menurut Roger S. Pressman, Ph.D (2012 : 51), *Prototype* seringkali pelanggan mendefinisikan sejumlah sasaran perangkat lunak secara umum, tetapi tidak biasa mengidentifikasi spesifikasi kebutuhan yang rinci untuk fungsi-fungsi dan fitur-fitur yang nantinya akan dimiliki perangkat lunak yang akan dikembangkan.

Dalam kasus yang lain, perkembangan perangkat lunak mungkin merasa tidak pasti tentang efisiensi suatu algoritma yang akan digunakan dalam perkembangan perangkat lunak, atau juga sistem operasi yang akan digunakan

atau merasa tidak pasti akan bentuk interaksi manusia-komputer yang akan digunakan. Dalam kasus-kasus seperti ini dan banyak dalam situasi yang lain, paradigma pembuatan *prototype* (*prototyping*) mungkin menawarkan pendekatan yang paling baik.

Meskipun pembuatan *prototype* dapat digunakan sebagai model proses yang berdiri sendiri, pembuatan *prototype* lebih umum digunakan sebagai teknik yang dapat diimplementasikan didalam hal ini, tidak terlalu peduli dengan dimana ia diterapkan, paradigma pembuatan *prototype* sering kali membantu tim perkembangan perangkat lunak *stakeholder* untuk memahami lebih baik apa yang akan dikembangkan saat *spesifikasi* kebutuhan belum jelas. Berikut gambar metode *prototype*:



**Sumber :** Roger S. Pressman, Ph. D (2012 : 51)

**Gambar 2.1** Metode *Prototype*

## 2.4 Pemodelan Sistem

### 2.4.1 Pengertian UML (*Unified Modeling Language*)

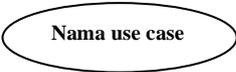
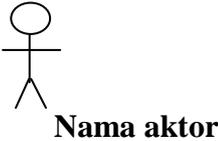
Menurut Rosa & Shalahuddin (2014 : 133), UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman

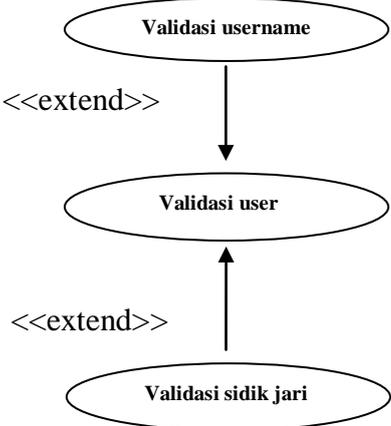
berorientasi objek. UML (*Unified Modeling Language*) terdapat 3 *diagram* yang dikelompokkan berdasarkan sifatnya, yaitu :

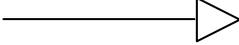
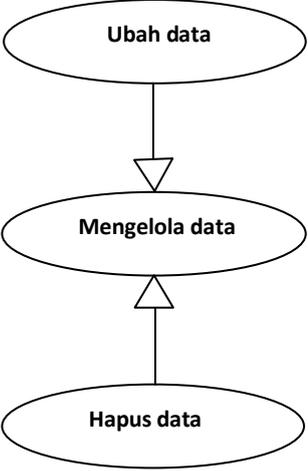
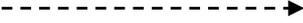
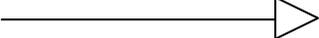
#### 2.4.1.1 *Use Case Diagram*

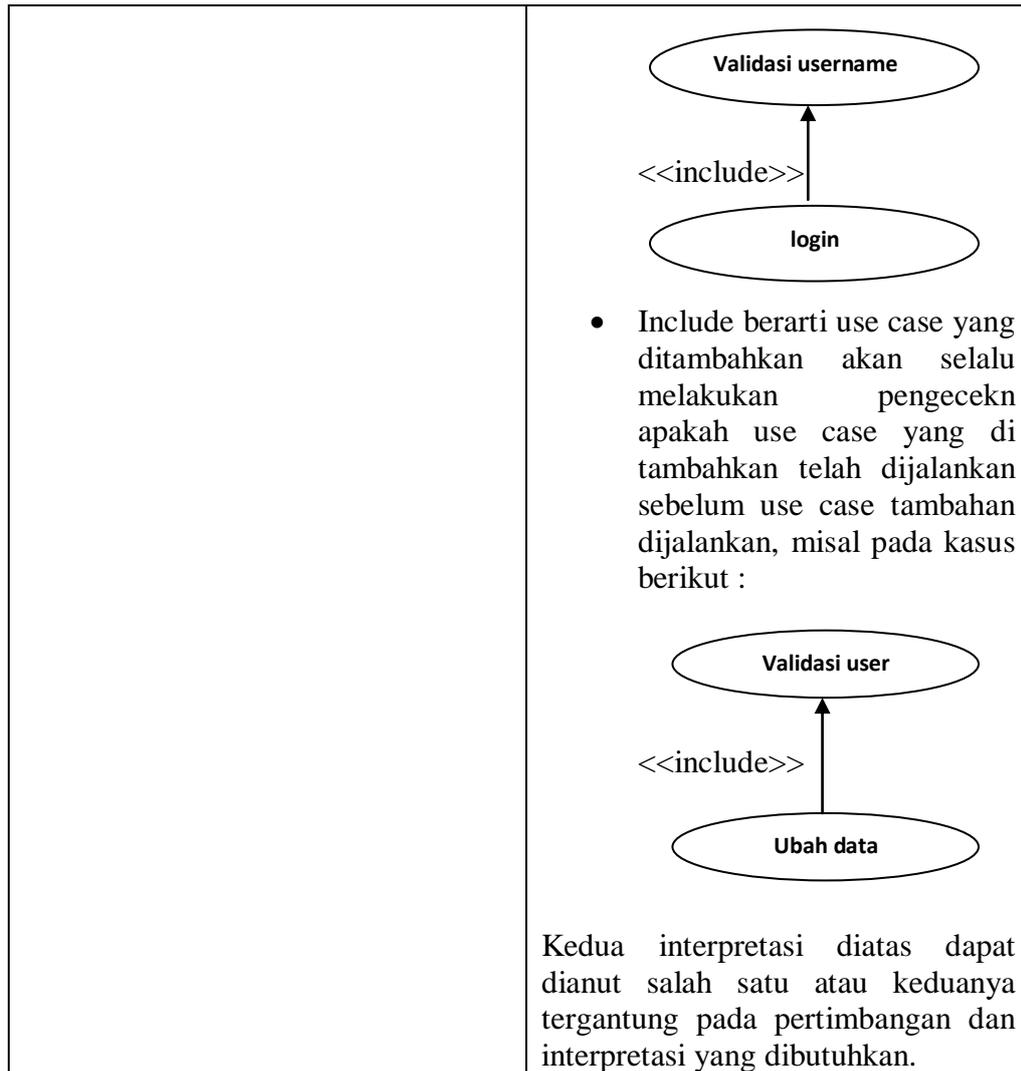
*Use Case diagram* merupakan pemodelan untuk melakukan sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut adalah simbol-simbol yang ada pada diagram *use case* :

**Tabel. 2.1 Simbol-simbol *Use Case***

<b>Simbol</b>	<b>Deskripsi</b>
Use case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>
Aktor / <i>actor</i> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor
Asosiasi / <i>association</i> 	Komunikasi antara aktor dan use case yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor

<p>Ektensi / <i>extend</i></p> <p>&lt;&lt;extend&gt;&gt;</p> <p>-----▶</p>	<p>Relasi <i>use case</i> tambahan kesebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal</p>  <pre>graph TD; A([Validasi username]) -- "&lt;&lt;extend&gt;&gt;" --&gt; B([Validasi user]); C([Validasi sidik jari]) -- "&lt;&lt;extend&gt;&gt;" --&gt; B;</pre> <p>Arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya</p>
----------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>Generalisasi / <i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya :</p>  <p>Arah panah mengarah pada use case yang menjadi generalisainya (umum)</p>
<p>Menggunakan / <i>include / uses</i></p> <p>&lt;&lt;include&gt;&gt;</p>  <p>&lt;&lt;uses&gt;&gt;</p> 	<p>Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini</p> <p>Ada dua sudut pandang yang cukup besar mengenai include di use case</p> <ul style="list-style-type: none"> <li>• Include berarti use case yang ditambahkan akan selalu dipanggil saat use case tambahan dijalankan, misal pada kasus berikut :</li> </ul>

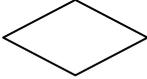
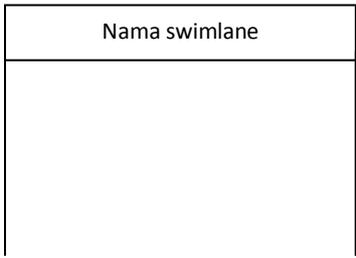
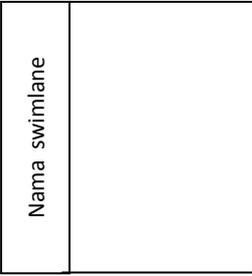


**Sumber :** Rosa dan Shalahuddin (2014 : 135)

#### 2.4.1.2 Activity Diagram

Diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan yang dilakukan oleh aktor. Jadi aktivitas yang dapat dilakukan oleh sistem. Berikut adalah simbol-simbol yang ada pada diagram aktivitas :

Tabel. 2.2 Simbol-simbol *Activity Diagram*

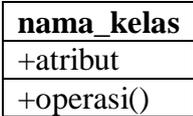
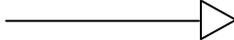
Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Swimlane  Atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber : Rosa dan Shalahuddin (2014 : 137)

### 2.4.1.3 Class Diagram

*Class diagram* atau diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Berikut adalah simbol-simbol yang ada pada diagram kelas :

**Tabel. 2.3 Simbol-simbol Class Diagram**

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka / <i>Interface</i>  <b>Nama_interface</b>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi / <i>association</i> 	Relasi antarkelas dengan makna umum koma asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah / <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antar kelas
Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian ( <i>whole-part</i> )

**Sumber :** Rosa dan Shalahuddin (2014 : 138)

## 2.5 Pengujian Perangkat Lunak

Menurut Pressman (2012:550) pengujian adalah serangkaian kegiatan yang dapat direncanakan di muka dan dilakukan secara sistematis. Ada banyak strategi pengujian yang dapat digunakan untuk menguji perangkat lunak, seperti pengujian atas-ke-bawah, pengujian bawah-ke-atas, pengujian regresi, pengujian asap, pengujian *sandwich*, pengujian kotak putih, dan pengujian kotak hitam. (Pressman 2012:560). Dalam penelitian skripsi ini penulis akan menggunakan satu pengujian, yaitu :

### 2.5.1 Pengujian Kotak Hitam (Black Box Testing)

Menurut Pressman (2012:597) *black box testing* (pengujian kotak hitam) juga disebut pengujian perilaku, berfokus pada persyaratan fungsional perangkat lunak. Artinya, teknik pengujian kotak hitam memungkinkan anda untuk membuat beberapa kumpulan kondisi masukan yang sepenuhnya akan melakukan semua kebutuhan fungsional untuk program. Pengujian kotak hitam berupaya untuk menemukan kesalahan dalam kategori berikut : (1) fungsi yang salah atau hilang, (2) kesalahan antar muka, (3) kesalahan dalam struktur data atau akses basis data eksternal , (4) kesalahan perilaku atau kinerja dan, (5) kesalahan inisialisasi dan penghentian.

## 2.6 Tinjauan Pustaka

Berikut adalah penelitian terdahulu tentang sistem informasi rekam medis :

1. Gunawan Susanto 2013 (Sistem Informasi Rekam Medis pada Rumah Sakit Umum Daerah (RSUD) Pacitan Berbasis Web Basis).

Hasil penelitian ini adalah dapat membangun sistem informasi yang meliputi pengolahan data seperti pasien, perawat poli atau ruangan, ICD 10, kasus penyakit, petugas rekam medis dan laporan. Pengembangan sistem informasi rekam medis tersebut ditujukan untuk mendukung ketersediaan data informasi bagi manajemen dan pelaksanaan layanan serta pengembangan jaringan informasi kesehatan. Sistem tersebut dibangun dengan teknologi komputer berbasis web. Sistem informasi rekam medis tersebut dibangun dengan menggunakan bahasa pemrograman PHP dengan didukung basis data MySQL.

2. Fahmi Hakam 2015 (Perancangan Sistem Informasi Rekam Medis Di Klinik Griya Medika Menggunakan Metode Prototyping ).

Hasil Penelitian ini adalah dapat mengembangkan sistem, dengan membuat rancangan sistem informasi rekam medis. Dari hasil identifikasi user, peneliti mulai merancang data flow diagram (DFD), entity relationship diagram (ERD), basis data dan interface sistem, serta bentuk cetak laporan yang dihasilkan. Dalam sistem ini output yang dihasilkan berupa cetakan laporan yang sudah sesuai dengan keinginan pengguna, seperti laporan sebaran penyakit, laporan data pasien dan laporan kunjungan pasien.

3. M. Agung Wijaya 2012 (Sistem informasi rekam medis terpadu pada rumah sakit pelabuhan Palembang berbasis web menggunakan metode TCSD (task centered system Design).

Hasil penelitian ini adalah untuk membangun Sistem informasi rekam medis terpadu pada rumah sakit pelabuhan Palembang berbasis web menggunakan metode TCSD (task centered system Design) yang dapat membantu kinerja

pihak rumah sakit khususnya unit rekam medis dalam pengolahan data agar menjadi lebih efektif dan efisien, dan juga menghindari terjadinya redudansi data.

4. Endang Lestari 2011 (Sistem Informasi Rekam Medik Pada Rumah Sakit Bersalin Graha Rap Tanjung Balai Karimun).

Hasil dari penelitian ini adalah dapat mempercepat pemrosesan seluruh dokumen pasien pada Rumah Sakit Bersalin Graha Rap Tanjung Balai Karimun. Serta mampu mengatasi terjadinya redudansi data, kehilangan kartu rekam medik pada sistem lama sehingga dapat mengurangi kesalahan-kesalahan serta mempercepat dalam pengolaan data pasien. Sistem informasi rekam medik yang akan dibuat menggunakan metode FAST. Dalam sistem ini implementasi yang dihasilkan berupa halaman pendataan pasien, registrasi pasien, input diagnosa, input hasil pemeriksaan, halaman hasil pemerisaan, halaman daftar antrian, halaman data pasien RAP, dan halaman pembayaran.

5. Yuyun Puspitasari (Sistem Informasi Rekam medis Pasien Rawat Jalan Pada Puskesmas Pringkuku Kabupaten Pacitan).

Hasil dari penelitian ini adalah sistem informasi rekam medis ini dibuat dengan menggunakan bahasa pemrograman PHP dan database MySQL. sistem informasi rekam medis pasien rawat jalan ini diharapkan dapat memudahkan pencarian data pasien, memudahkan dalam pembuatan laporan dan menghasilkan data yang akurat, serta dapat meminimalisir kesalahan dalam pengolahan data rekam medis pasien rawat jalan.

Berdasarkan tinjauan pustaka tersebut maka penulis akan membuat Sistem Informasi *E-Medic* Pada Rumah Sakit Umum Daerah Lahat (RSUD Lahat)

Berbasis Web Dengan Menggunakan PHP Dan Android yang dapat mempermudah pengolahan data pasien, data obat, jadwal dokter, data poliklinik, data rekam medis, cetak kartu berobat (KRM), serta diagram tahunan penyakit pasien. Sistem informasi ini dibuat dengan menggunakan metode *Prototype*.