

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

2.1.1 Kosep Dasar Perancangan

Menurut Syahbana (2013:44) “Perancangan adalah suatu bagian dari metodologi pengembangan pembangunan suatu perangkat lunak yang dilakukan setelah tahapan analisis yang dimaksudkan untuk memberikan gambaran secara terperinci”.

2.1.2 Konsep Dasar Sistem Informasi

1. Sistem

Menurut Kristanto (2008:1) “sistem merupakan kumpulan elemen-elemen yang saling terkait dan bekerja sama untuk memproses masukan (*input*) yang ditunjukkan kepada sistem tersebut dan mengubah masukan tersebut sampai menghasilkan keluaran (*output*) yang diinginkan”.

2. Informasi

Menurut Paryati dan Ardhana (2008:17) informasi “merupakan kumpulan data yang diproses dan diolah menjadi data yang memiliki arti bagi penerimanya yang menggambarkan suatu kejadian-kejadian nyata dan dapat digunakan sebagai alat bantu untuk pengambilan suatu keputusan”.

3. Sistem Informasi

Menurut Sutjiadi (2014:36) “Sistem Informasi adalah kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi itu

untuk mendukung operasi dan manajemen”. Dalam arti yang sangat luas, istilah sistem informasi yang sering digunakan merujuk kepada interaksi antara orang, proses algoritmik, data, dan teknologi. Dalam pengertian ini, istilah ini digunakan untuk merujuk tidak hanya pada penggunaan organisasi teknologi informasi dan komunikasi (TIK), tetapi juga untuk cara di mana orang berinteraksi dengan teknologi ini dalam mendukung proses bisnis.

2.1.3 Pengertian Pelayanan

Menurut Sinambela (2006:5) “pelayanan publik diartikan sebagai pemberi pelayanan (melayani) keperluan orang atau masyarakat yang mempunyai kepentingan pada organisasi itu sesuai dengan aturan pokok dan tatacara yang ditetapkan”.

Dalam keputusan Menteri Pendayagunaan Aparatur negara No. 63 tahun 2003, pelayanan publik dibagi berdasarkan 3 kelompok, yaitu:

1. Kelompok Pelayanan Administratif, yaitu bentuk pelayanan yang menghasilkan berbagai macam dokumen resmi yang dibutuhkan oleh masyarakat atau publik. Misalnya status kewarganegaraan, kepemilikan, dan lain-lain. Dokumen-dokumen ini antara lain KTP.
2. Kelompok Pelayanan Barang, yaitu pelayanan yang menghasilkan berbagai bentuk/jenis barang yang digunakan publik. Misalnya penyediaan tenaga listrik, air bersih, dan lain-lain.

3. Kelompok Pelayanan Jasa, yaitu pelayanan yang menghasilkan berbagai bentuk jasa yang dibutuhkan publik. Misalnya pendidikan, pelayanan kesehatan, penyelenggaraan transportasi, dan lain-lain.

2.1.4 Pengertian Pasien

Menurut Undang-Undang Nomor 44 Tahun 2009 tentang Rumah Sakit, Pasal 1 Angka (4) dikatakan bahwa “Pasien adalah setiap orang yang melakukan konsultasi masalah kesehatannya untuk memperoleh pelayanan kesehatan yang diperlukan, baik secara langsung maupun tidak langsung di Rumah Sakit.”

2.1.5 Konsep Dasar WEB

Web menurut Arief (2011:7) adalah “salah satu aplikasi yang berisikan dokumen-dokumen *multimedia* (teks, gambar, suara, video) di dalamnya yang menggunakan protokol *HTTP* (*hypertext transfer protocol*) dan untuk mengaksesnya menggunakan perangkat lunak yang disebut *browser*”.

Maka harus tersedia komponen penunjangnya, adalah sebagai berikut :

1. Bahasa Pemrograman / *Scripting Language*

Secara garis besar ada 2 kategori *script* yang dibutuhkan untuk membuat aplikasi *web* :

- a. *Client side scripting*, contohnya adalah: *Hypertext Markup Language* (*HTML*), *HTML5*, *XHTML*, *Cascading Style Sheet* (*CSS*), *JavaScript*, *VBScript*, *jQuery*.
- b. *Server side scripting*, contohnya adalah: *Active Server Pages* (*ASP*), *PHP: Hypertext Preprocessor* (*PHP*), *Java Server Pages* (*JPS*).

2. *Web Editor*

Merupakan program aplikasi yang berfungsi untuk mengetikkan perintah-perintah dokumen *web* baik *client side scripting* maupun *server side scripting*.

3. *Web Browser*

Merupakan program yang berfungsi untuk menampilkan dokumen-dokumen *web* dalam format *HTML*. Bagaimana halaman *web* yang ditampilkan sangat tergantung pada *web engine* yang digunakan oleh masing-masing *browser*.

4. *Web Server*

Web server adalah program aplikasi yang memiliki fungsi sebagai tempat menyimpan dokumen-dokumen *web*.

5. *Database Server*

Database server adalah program yang digunakan untuk menyimpan data yang akan diolah dihalaman *web*.

Menurut Hidayat (2010:1) *Website* adalah kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar, diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman. Jenis-jenis *web* berdasarkan sifat atau *stylenya*.

1. *website dinamis*, merupakan sebuah *website* yang menyediakan *konten* atau isi yang selalu berubah –ubah setiap saat. Bahasa pemrograman yang digunakan antara lain *php,asp, .net* dan memanfaatkan *datatbase mysql* atau *mssql*.

2. *website statis* , merupakan *website* yang *kontennya* jarang diubah. bahasa pemrograman yang digunakan adalah html dan belum memanfaatkan *database*.

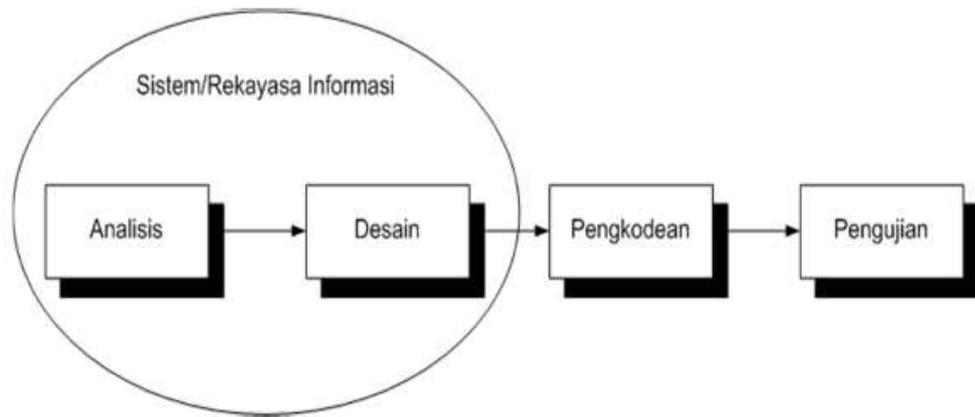
2.1.6 Pengertian Klinik

Berdasarkan peraturan Menteri Kesehatan Republik Indonesia Nomor 028/Menkes/Per/I/2011, pengertian klinik adalah fasilitas pelayanan kesehatan yang menyelenggarakan pelayanan kesehatan perorangan yang menyediakan pelayanan medis dasar dan spesialisik, diselenggarakan oleh lebih dari satu jenis tenaga kesehatan dan dipimpin oleh seorang tenaga medis.

2.1.7 Konsep Dasar Model Pengembangan sistem

Menurut Rosa dan Shalahuddin (2013:26) “SDLC adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik)”.

Menurut Rosa dan Shalahudiin (2013:28) menyimpulkan bahwa: Model SDLC air terjun(*waterfall*) sering juga disebut model sekuensial linier(*Sequential linear*) atau alur hidup klasik(*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung atau (*support*).



Sumber: Rosa dan Shalahudin (2013:29)

Gambar II.1. ilustrasi model waterfall

1. Analisis kebutuhan perangkat lunak

Menurut Rosa dan Shalahuddin(2013:29) “Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan”.

2. Desain

Menurut Rosa dan Shalahuddin (2013:29) “Desain perangkat lunak adalah proses multilangkah yang fokus pada desain pembuatan program perangkat lunak, representasi antarmuka, dan prosedur pengkodean”. Tahap ini mentranslasi ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. Pembuatan kode program

Menurut Rosa dan Shalahuddin (2013:29) “Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain”.

4. Pengujian

Menurut Rosa dan Shalahuddin (2013:30) “Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan”.

5. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Menurut Rosa dan Shalahuddin (2013:30) menyimpulkan bahwa: Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.1.8 Konsep Dasar Pemrograman

Menurut Rosa dan shalahudin (2013:67) ”pemrograman terstruktur adalah konsep atau paradigma atau sudut pandang pemrograman yang membagi-bagi program berdasarkan fungsi-fungsi atau prosedur-prosedur yang dibutuhkan program komputer”. Modul-modul (pembagian program) biasanya dibuat dengan mengelompokkan fungsi-fungsi dan prosedur-prosedur yang diperlukan sebuah proses tertentu.

Fungsi-fungsi dan prosedur-prosedur ditulis secara sekuensial atau terurut dari atas kebawah sesuai dengan kebergantungan antar fungsi atau prosedur. Pemodelan pada pemrograman terstruktur lebih fokus bagaimana memodelkan

data dan fungsi-fungsi atau prosedur-prosedur yang harus dibuat. Jenis paradigma pemrograman yang digunakan dapat dideteksi dari bahasa pemrograman apa yang akan untuk membuat program, baru setelah itu ditentukan paradigma pemrograman apa yang akan digunakan.

2.1.9 UML (*Unified Modeling Language*)

Menurut Rosa dan Shalahuddin (2013:133) “UML adalah salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”.

Ada beberapa diagram dalam UML (*Unified Modelling Language*) antara lain:

1. *Use Case Diagram*

Menurut Rosa dan Shalahuddin (2013:155) “*use case* atau *diagram use case* merupakan pemodelan untuk kelakuan (*behavior*) 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. Syarat penamaan *use case* adalah nama didefinisikan sesimpel mungkin dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisikan apa yang disebut aktor dan *use case*.

- a. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi

yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.

- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan unit atau aktor.

2. *Activity Diagram*

Menurut Rosa dan Shalahuddin (2013:161) “diagram aktifitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”. Yang perlu diperhatikan disini adalah bahwa diagram aktifitas menggambarkan aktifitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktifitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

- a. Rancangan proses bisnis dimana setiap urutan aktifitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
- b. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktifitas dianggap memiliki sebuah rancangan antarmuka tampilan
- c. Rancangan pengujian dimana setiap aktifitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya
- d. Rancangan menu yang ditampilkan pada perangkat lunak

3. *Package Diagram*

Menurut Rosa dan Shalahuddin (2013:153) “*Package Diagram* menyediakan cara mengumpulkan elemen-elemen yang saling terkait dalam

UML. Hampir semua diagram dalam UML dapat dikelompokkan menggunakan *Package Diagram*”.

4. *Component Diagram*

Menurut Rosa dan Shalahuddin (2013:148) “diagram komponen atau *Component Diagram* dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada di dalam sistem”. Diagram komponen juga dapat digunakan untuk memodelkan hal-hal berikut:

- a. *Source code* program perangkat lunak
- b. Komponen *executable* yang dilepas ke *user*.
- c. Basis data secara fisik
- d. Sistem yang harus beradaptasi dengan sistem lain
- e. *Framework* sistem

5. *Deployment Diagram*

Menurut Rosa dan Shalahuddin (2013:154) “Diagram *deployment* atau *deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi”. Diagram *deployment* juga dapat digunakan untuk memodelkan hal-hal berikut:

- a. sistem tambahan (*embedded system*) yang menggambarkan rancangan *device, node, dan hardware*.
- b. Sistem *client/server*
- c. Sistem terdistribusi murni
- d. Rekayasa ulang aplikasi

2.1.10 ERD (*Entity Relationship Diagram*)

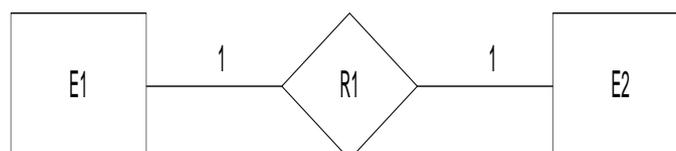
Menurut Raharjo (2015:47) “ERD merupakan salah satu alat bantu (berupa gambar) dalam model database relasional yang berguna untuk menjelaskan hubungan atau relasi antar tabel yang terdapat di dalam database. Dalam ERD kita juga dapat melihat daftar kolom yang menyusun masing-masing tabel”.

Menurut Rosa dan Shalahudin (2013:50) “Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram* (ERD)”. ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD.

ERD biasanya memiliki hubungan *binary* (satu relasi menghubungkan dua buah entitas). Beberapa metode perancangan ERD menoleransi hubungan relasi *ternary* (satu relasi menghubungkan tiga buah relasi) atau *N-ary* (satu relasi menghubungkan banyak entitas), tapi banyak metode perancangan ERD yang tidak mengizinkan hubungan *ternary* atau *N-ary*. Berikut adalah contoh bentuk hubungan relasi dalam ERD.

1. *Binary*

Tingkat hubungan satu kesatu, dinyatakan dengan satu kejadian pada entitas pertama.

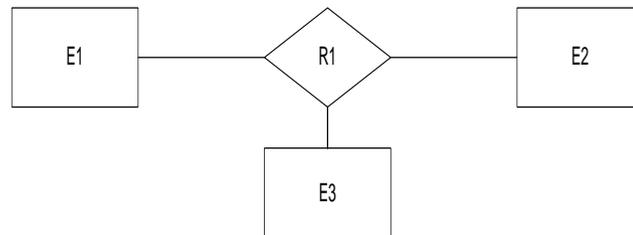


Sumber: Rosa dan Shalahudin (2013:52)

Gambar II.2. Binary

2. *Ternary*

Tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu tergantung dari arah mana hubungan tersebut dilihat.

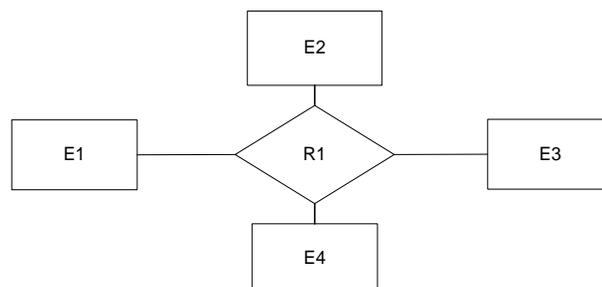


Sumber: Sumber: Rosa dan Shalahudin (2013:52)

Gambar II.3. Ternary

3. *N-ary*

Tingkat hubungan banyak ke banyak terjadi tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya. Hubungan antara file pertama dengan file kedua adalah banyak berbanding banyak. Relasi antar keduanya diwakili dengan tanda panah ganda disetiap sisinya untuk menunjukkan hubungan banyak tersebut.



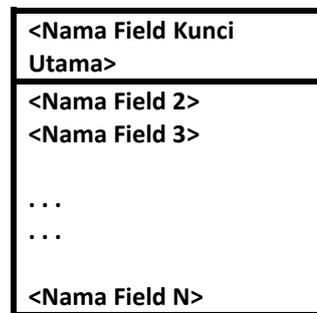
Sumber: Rosa dan Shalahudin (2013:52)

Gambar II.4. N-ary

2.1.11 LRS (*Logical Record Structure*)

Menurut Frieyadie (2007:13) “*Logical Record Structure* merupakan hasil dari pemodelan *Entity Relationship* (ER) beserta atributnya sehingga bisa terlihat hubungan-hubungan antar entitas”. Sebelum *table* dibentuk dari *field* atau atribut entitas secara fisik atau *level internal*, maka harus dibuatkan suatu bentuk *Relational* model yang dibuat secara *logic* atau *level external* dan konsep, dari pernyataan tersebut dibutuhkan yang disebut dengan *Logical Relationship Structure* (LRS).

<Nama Tabel>



Sumber : Frieyadie (2007:14)

Gambar II.5. (Logical Record Structure)

2.1.12 *Intranet*

Menurut Turban (2008:124) “*intranet* merupakan jaringan perusahaan yang berfungsi dengan teknologi *internet* seperti *browser* dan *search engine*, menggunakan protokol *internet*”.

Intranet digunakan untuk mempercepat proses-proses pertukaran informasi maupun data. Untuk skala kecilnya, *intranet* dapat dioperasikan dalam sebuah *local Area Network* (LAN), dan untuk skala yang lebih besar, *intranet* dapat dioperasikan pada *Wide Area Network* (WAN).

2.1.13 MySQL

Menurut Susanto dan Sukadi (2011:21) “MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial”. MySQL merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian *database* untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah. Keandalan suatu database (DBMS) dapat diketahui dari cara kerja optimizernya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya.

MySQL memiliki beberapa keistimewaan, antara lain:

1. Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti *Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga*, dan lainnya.
2. *Open Source*. MySQL didistribusikan secara *open source*, dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.
3. '*Multiuser*'. MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan.
4. '*Performance tuning*'. MySQL memiliki kecepatan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. Keamanan. MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.
6. Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
7. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, *Unix socket* (UNIX), atau *Named Pipes* (NT).
8. Antar Muka. MySQL memiliki *interface* (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
9. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.

Menurut Syukron dan Hasan dalam Nugroho (2008:91) “*MySQL(My Structured Query Language)* adalah sebuah program pembuat dan pengelola *database* atau yang sering disebut dengan *DBMS (Database Management System)*”. *Database* berfungsi sebagai penampung data yang akan dimasukkan melalui *form website*. Selain itu dapat juga dibalik dengan menampilkan data yang tersimpan dalam *database* ke dalam halaman *website*. *MySQL* merupakan penyimpanan data yang *fleksibel* dan cepat *aksesnya* sangat dibutuhkan dalam sebuah *website* yang *interaktif* dan *dinamis*.

2.1.14 PHP (*PHP Hypertext Preprocessor*)

Menurut Sutjiadi (2014:37) “PHP: *Hypertext Preprocessor* adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML sejatinya dirancang untuk menghasilkan halaman-halaman *web* yang dinamis. Arti dari *web* dinamis sendiri adalah halaman web yang sumbernya menggunakan *database* sehingga dapat diperbarui secara berkala”. Berbeda dengan halaman HTML standar yang apabila ingin diperbarui, maka seluruh isi halaman HTML tersebut harus diunggah.

Menurut Winarno dan Zaki (2013:73) “PHP adalah sebuah bahasa pemrograman *scripting* untuk membuat halaman web yang dinamis. Maksud dari *scripting* adalah pembuatannya menggunakan *editor* teks biasa, seperti *Notepad*, *Notepad++*, dan lainnya”.

Walaupun dikenal sebagai bahasa untuk memuat halaman *web*, tapi PHP sebenarnya juga dapat digunakan untuk membuat aplikasi *command line* dan juga GUI. Cara kerja PHP adalah dengan menyelipkannya di antara kode HTML (*hypertext markup language*). *File* HTML murni yang berekstensi *.htm* atau *.html* akan langsung diperoleh dari *server*. Sementara *file* HTML yang memiliki script PHP umumnya berekstensi *.php*. Karenanya *script .php* akan diolah/di-*parsing* terlebih dahulu di PHP *parser* yang ada di *server* untuk menghasilkan tampilan kode HTML murni. HTML adalah dokumen sederhana, berfungsi untuk menampilkan teks dan juga halaman gambar.

2.1.15 HTML (*hypertext markup language*)

Menurut Ardhana (2012:42) HTML “merupakan suatu bahasa yang dikenali oleh *web browser* untuk menampilkan informasi seperti teks, gambar, suara, animasi bahkan video. Kode HTML memiliki aturan dan struktur penulisan tersendiri yang disebut tag HTML”. *Tag* adalah kode yang digunakan untuk memoles (*mark-up*) teks ASCII (*American Standard Code Of Information Interchange*) menjadi *file* HTML. Setiap *tag* diapit dengan tanda kurung runcing. Ada *tag* pembuka yaitu <HTML> dan ada *tag* penutup yaitu </HTML> yang ditandai dengan tanda garis miring (*slash*) di depan awal tulisannya.

Tag diatas memberikan faidah bahwa yang akan ditulis diantara kedua *tag* tersebut adalah isi dokumen HTML. Perlu anda ketahui bahwa tag-tag html dapat ditulis dengan huruf besar ataupun hurup kecil. Atrinya, penulisan <HTML> atau <html> atau <Html> sama saja hasilnya. Namun perlu selalu diingat bahwa penulisan tag yang salah meskipun hanya satu karakter akan berpengaruh terhadap dokumen HTML anda, bahkan bisa berakibat dokumen HTML anda tidak bisa ditampilkan dalam *browser*.

Contoh kode program:

```
<html>

    <head>

// <title> .....</title> berfungsi untuk menampilkan judul pada website.

    <title>.....</title>

    </head>

<body>

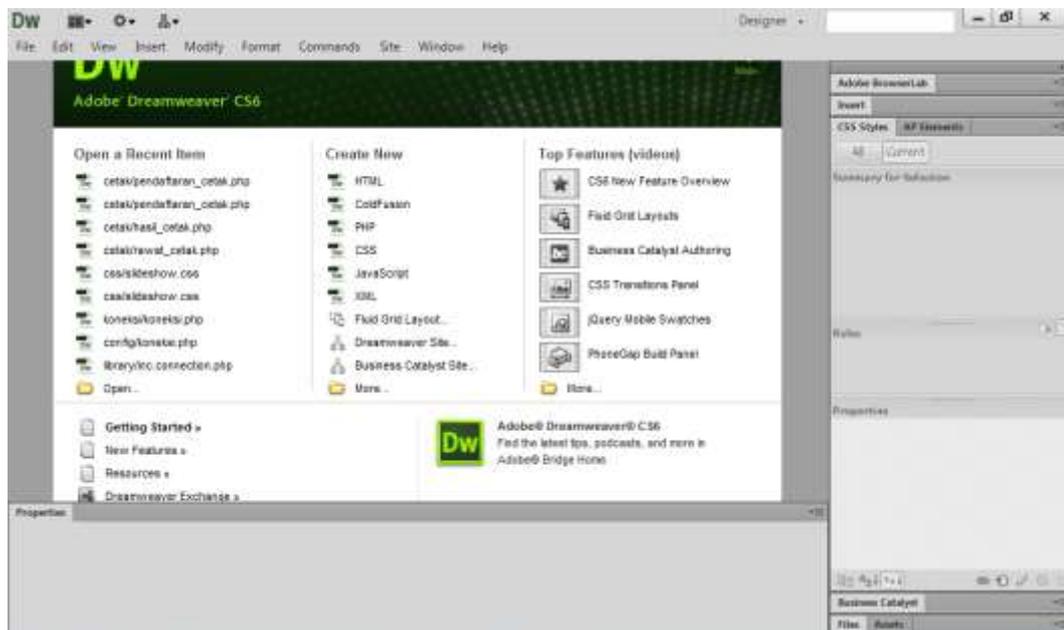
.....
```

```
</body>
```

```
</html>
```

2.1.16 Dreamweaver

Menurut MADCOMS (2016:30) “*Dreamweaver* adalah *software* aplikasi desain web visual yang biasa di kenal dengan istilah WYSIWYG (*what you see is what you get*), intinya anda tidak harus berusaha dengan tag-tag HTML untuk membuat sebuah situs”. *Dreamweaver* tidak hanya dapat digunakan oleh para *desainer web*, namun juga dapat digunakan oleh *programmer* untuk membangun halaman interaktif karena *dreamweaver* mendukung pula PHP, *ColdFusion*, ASP.NET dan lain-lain.



Sumber: MADCOMS (2016:30)

Gambar II.6 Tampilan Awal Dreamweaver

2.1.17 XAMPP

Menurut Riyanto (2010:1) “XAMPP merupakan paket PHP dan MySQL berbasis *open source*, yang dapat digunakan sebagai *tool* pembantu pengembangan aplikasi berbasis PHP. XAMPP mengombinasikan beberapa paket perangkat lunak berbeda ke dalam satu paket”. XAMPP menyediakan antarmuka *control panel* tersendiri yang dapat digunakan untuk menjalankan seluruh layanannya.

Menurut MADCOMS (2016:148) “Xampp adalah sebuah paket kumpulan software yang terdiri dari Apache, MySQL, PhpMyAdmin, PHP, Perl, Filezilla, dan lain-lain”. Xampp berfungsi untuk memudahkan instalasi lingkungan PHP, dimana biasanya lingkungan pengembangan *web* memerlukan PHP, Apache, MySQL dan PhpMyAdmin serta software-software yang terkait dengan pengembangan *web*.

2.1.18 CSS (*Cascading Style Sheet*)

Menurut MADCOMS (2016:46) “ CSS (*cascading Style Sheet*) adalah kumpulan kode-kode yang berurutan dan saling berhubungan untuk mengatur format / tampilan suatu HTML”.

Dengan CSS maka akan memungkinkan anda untuk mendesain (*style*) tampilan dokumen (PHP dan HTML) dengan memisahkan isi dari dokumen HTML dengan kode untuk menampilkan (CSS). Di dalam dunia desain *web*, *layout* atau mengatur letak *web*, *format* huruf, dan lainnya yang ada dalam sebuah *web* menjadi bagian yang amat penting bahkan dapat jadi yang paling lama

dibandingkan dengan bagian desain *web* lainnya. karena jika anda membuat *layout* yang salah, maka *web* anda juga akan salah diterjemahkan oleh *browser*.

jika anda memiliki banyak *file* HTML, anda hanya perlu satu *file* CSS, sehingga ketika anda mengganti jenis huruf pada CSS maka semua *file* HTML yang berhubungan dengan *file* CSS tersebut akan berubah.

2.2 Penelitian Terkait

Hasil penelitian sistem pendaftaran di klinik dr Sri Widatik dengan analisis PIECES menunjukkan bahwa petugas masih kesulitan mencari nomor rekam medis, informasi yang dihasilkan belum lengkap, penggunaan kertas yang semakin lama digunakan akan mengalami perubahan bentuk, orang yang tidak diberi kewenangan bisa mendaftarkan pasien, kemungkinan memerlukan tambahan karyawan untuk proses pendaftaran, pasien terlalu lama menunggu petugas untuk menemukan rekam medis pasien. Dari hasil analisis sistem lama maka dibuatlah perancangan sistem baru dengan kelebihan dan kelemahannya. Dengan berkembangnya *evidence based medicine* dimana pelayanan medis yang berbasis data sangatlah diperlukan maka kegiatan pendaftaran tersebut dapat dipenuhi dengan tersedianya alat bantu yang dapat memudahkan pekerjaan, salah satunya dengan menggunakan komputerisasi. (Putra, Rohmadi, dan Mulyono, 2013:19).

Dalam upaya peningkatan kualitas pelayanan kesehatan yang lebih baik ditingkat puskesmas pada khususnya. Maka diperlukan sebuah konsep atau sistem yang baik yang akan digunakan, sehingga nantinya dapat terwujud suatu pelayanan kesehatan yang bermutu, efektif dan efisien serta dapat meningkatkan kinerja dari puskesmas itu sendiri. Salah satu contohnya adalah penggunaan sistem informasi yang sudah terkomputerisasi pada puskesmas. Dengan penggunaan sistem komputerisasi, diharapkan dapat meningkatkan kualitas administrasi puskesmas. Dari hasil pengamatan yang telah dilakukan, ternyata masih terdapat beberapa kendalakendala yang dihadapi, seperti halnya pada bagian registrasi pasien rawat jalan yang mengalami kendala pada saat mengidentifikasi pasien yang akan berobat dan juga pada bagian administrasi dalam membuat laporan kesehatan. (Syukron dan Hasan,2015:28).