

BAB 2

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Sebagai dasar dalam penelitian, penulis menggunakan dua jurnal penelitian yaitu “Perancangan Sistem Informasi Pencatatan Pengiriman Barang di PT. TIKI Jalur Nugraha Eka Kurir Cabang Bandung ” yang disusun oleh Eka Triyanto¹, Widya Revina, S.Kom² [1] dan “ Perancangan Web Jasa Pengiriman Barang “yang disusun oleh Harry Dhika¹, Lukman², Aswin Fitriansyah³ pada tahun 2015 [2].

Pada jurnal “Perancangan Sistem Informasi Pencatatan Pengiriman Barang di PT. TIKI Jalur Nugraha Eka Kurir Cabang Bandung”. Dalam penelitiannya penulis melihat ada masalah dalam hal seperti penyimpanan data pengiriman barang belum dibuatkan berdasarkan kategori dan kualifikasi seperti barang yang berasuransi (HVS) dan barang yang non asuransi (NON HVS). Pada penelitian ini penulis merancang dan membangun sebuah sistem informasi dengan menggunakan metode pengembangan sistem SSAD (Structured System Analysis and Design) dan dengan pendekatan SDLC model waterfall. Bahasa pemrograman adalah VB.Net dan database yang digunakan adalah SQL Server 2005, SAP crystal report untuk merancang dan menghasilkan laporan.. Dan menghasilkan sebuah sistem yang mampu melakukan pencatatan pengiriman barang yang kemudian akan tersimpan berdasarkan kategor dan dapat memudahkan pencatata barang pada saat akan melakukan packing.[1].

Pada jurnal “ Perancangan Web Jasa Pengiriman Barang”. Dalam penelitian ini penulis meliat masalah yang terjadi yaitu web pengiriman barang yang ada masih memiliki kekurangan. Pada penelitian ini penulis merancang dan membangun sebuah sistem dengan metode kausal. Peneliti menghasilkan sebuah perancangan yang memiliki fasilitas SMS gateway dan Mendapatkan informasi mengenai perjalanan barang yang dikirim dari daerah yang satu ke daerah yang lainnya

sehingga pihak penyedia layanan atau pemilik barang dapat menggunakan informasi tersebut untuk kepentingan bisnis. [2].

| No | Nama Peneliti dan Tahun | Masalah | Solusi | Hasil |
|----|---|---|--|--|
| 1. | Eka Triyanto ¹ , Widya Revina, S.Kom ² | Penyimpanan data pengiriman barang belum dibuatkan berdasarkan kategori dan kualifikasi seperti barang yang berasuransi (HVS) dan barang yang non asuransi (NON HVS). | Membangun sebuah sistem informasi mengenai pencatatan pengiriman barang. | Pencatatan pengiriman barang akan tersimpan berdasarkan kategor dan dapat memudahkan pencatata barang pada saat akan mellakukan packing. |
| 2. | Harry Dhika ¹ , Lukman ² , Aswin Fitriansyah ³ , 2015 | Perlunya pengembangan fitur pada web jasa pengiriman barang. | Menganalisa web jasa pengiriman dan merancang web pengiriman barang yang lebih baik. | Perancangan memiliki fasilitas SMS gateway dan Mendapatkan informasi mengenai perjalanan barang yang dikirim dari daerah yang satu |

| No | Nama Peneliti dan Tahun | Masalah | Solusi | Hasil |
|----|-------------------------|---------|--------|---|
| | | | | ke daerah yang lainnya sehingga pihak penyedia layanan atau pemilik barang dapat menggunakan informasi tersebut untuk kepentingan bisnis. |

2.2 Landasan Teori

2.2.1 *Tracing*

Tracing memiliki arti penelusuran, pengiusut, atau pelacakan, dalam hal pengiriman barang *tracing* diartikan sebagai penelusuran jejak yang dilakukan untuk mengetahui posisi suatu objek, seperti pengecekan lokasi barang kiriman [2].

2.2.2 Konsep Dasar Sistem

2.2.2.1 Pengertian Sistem

Beberapa ahli mengemukakan pendapatnya tentang pengertian sistem, diantaranya:

1. Menurut pendapat Law, sistem merupakan sekelompok komponen yang beroperasi secara bersama untuk mencapai tujuan tertentu atau sekumpulan

entitas yang bertindak dan berinteraksi bersama untuk memenuhi tujuan akhir yang logis [3].

2. Menurut pendapat McLeod, sistem adalah sekelompok elemen-elemen yang terintegrasi dengan tujuan yang sama [3].

Dari pendapat pakar di atas, pengertian umum mengenai sistem dapat disimpulkan sebagai sekumpulan elemen-elemen yang saling bekerja sama untuk mencapai tujuan sistem.

2.2.2.2 Klasifikasi Sistem

Sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda-beda. Oleh karena itu sistem dapat di klasifikasikan dari beberapa sudut pandang[3].

1. Sistem abstrak dan sistem fisik

Sistem abstrak adalah sistem yang berupa ide atau pikiran yang tidak tampak seperti fisik. Sedangkan sistem fisik adalah sistem yang ada secara fisik, seperti sistem komputer, sistem administrasi dan lain sebagainya.

2. Sistem alamiah dan sistem buatan manusia

Sistem alamiah adalah sistem yang terjadi karena proses alami tidak dibuat manusia, seperti terjadinya siang dan malam. Sedangkan sistem buatan manusia merupakan sistem yang melibatkan manusia dengan mesin.

3. Sistem deterministik dan sistem probabilistik

Sistem yang beroperasi dengan tingkah laku yang dapat diprediksi disebut sistem deterministik. Sedangkan sistem yang kondisi masa depannya tidak dapat diprediksi disebut sistem probabilistik.

4. Sistem terbuka dan sistem tertutup

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luar. Sistem ini bekerja secara otomatis tanpa ada campur tangan pihak luar. Sedangkan sistem terbuka adalah sistem yang berhubungan dan dipengaruhi lingkungan luarnya.

2.2.2.3 Karakteristik Sistem

Pada umumnya sistem terdiri dari input, proses, dan output. Hal tersebut merupakan konsep sebuah sistem yang sangat sederhana. Selain itu sebuah sistem juga memiliki karakteristik atau sifat-sifat tertentu, yang mencirikan bahwa hal tersebut dapat dikatakan sebuah sistem. Adapun karakteristik yang dimaksud adalah sebagai berikut[3]:

1. Komponen sistem (Components)

Sistem merupakan komponen yang saling berinteraksi dan bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Subsistem sendiri memiliki sifat-sifat sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. Batasan Sistem (Boundary)

Ruang lingkup sistem merupakan daerah yang membatasi antar sistem dengan sistem lain atau sistem dengan lingkungan luarnya. Batasan sistem ini dipandang sebagai suatu kesatuan yang tidak dapat dipisah-pisahkan.

3. Lingkungan Luar Sistem (Environment)

Bentuk apapun yang berada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem.

4. Penghubung Sistem (Interface)

Media penghubung sistem dengan subsistem disebut penghubung sistem atau interface. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lain.

5. Masukan Sistem (Input)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan dan sinyal.

6. Keluaran sistem (Output)

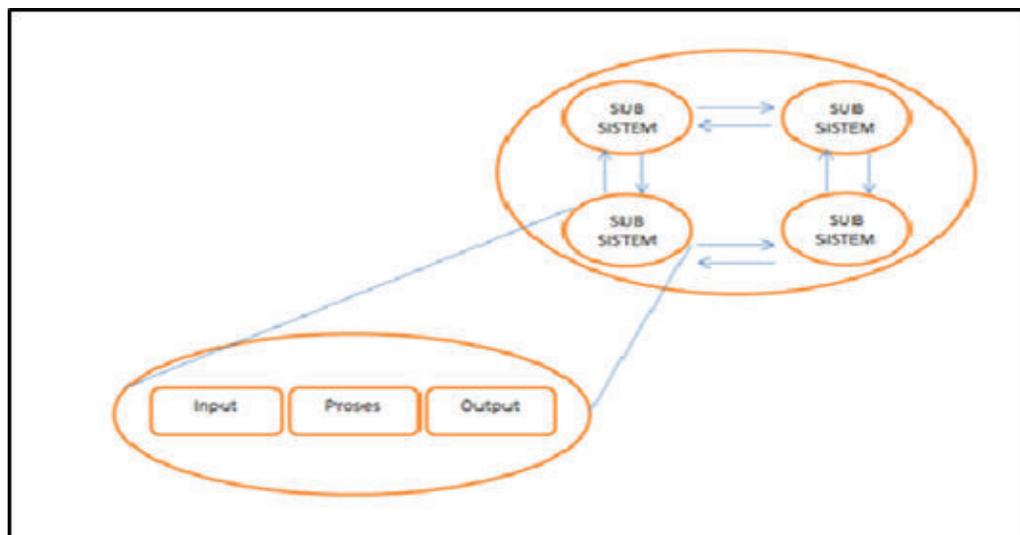
Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Seperti contoh sistem informasi, keluaran yang dihasilkan adalah informasi, dimana informasi dapat digunakan sebagai masukan untuk pengambilan keputusan.

7. Pengolahan sistem (Procces)

Suatu sistem mempunyai suatu proses yang dapat mengubah masukan menjadi keluaran. Sebagai contoh, sistem akuntansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan.

8. Sasaran sistem (objective)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik. Dimana suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang direncanakan.



Gambar 2.1 Karakteristik Sistem [3]

2.2.3 Konsep Dasar Informasi

2.2.3.1 Pengertian Informasi

Menurut McLeod, informasi merupakan data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan [3].

Informasi merupakan kumpulan data yang diolah menjadi bentuk yang lebih berguna dan berarti bagi yang menerima. Tanpa informasi, suatu sistem tidak akan berjalan dan akhirnya bisa mati. Suatu organisasi tanpa adanya informasi, maka organisasi tersebut tidak bisa berjalan dan beroperasi [3].

2.2.3.2 Pengertian Sistem Informasi

Sistem informasi (*information system*) merupakan kombinasi teratur dari orang (*brainware*), perangkat keras (*hardware*), perangkat lunak (*software*), jaringan komunikasi, dan sumber data yang mengumpulkan, mengubah, dan menyebarkan informasi dalam organisasi [3].

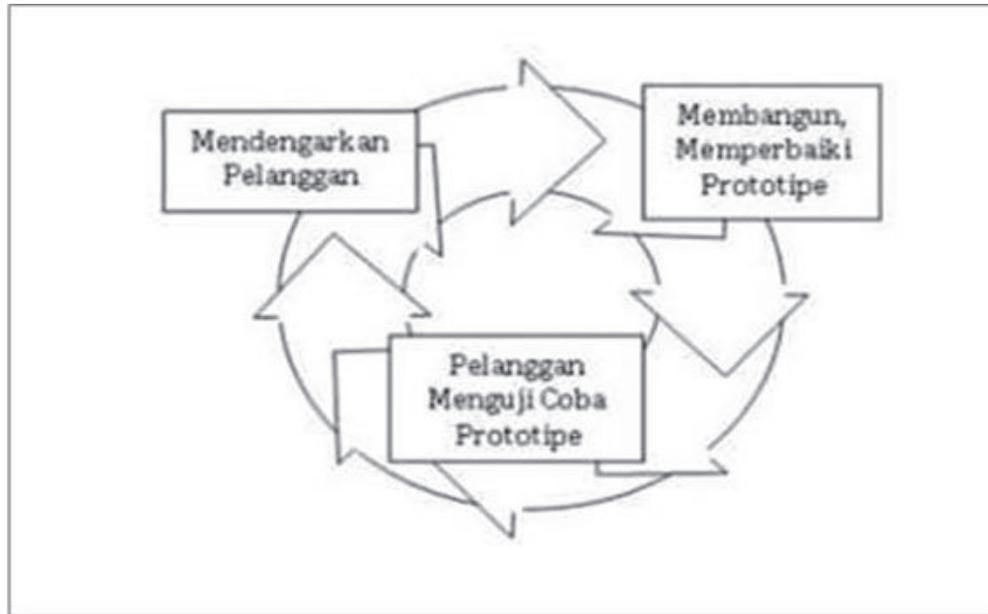
Sistem informasi merupakan hal yang dibutuhkan organisasi untuk pengolahan transaksi harian yang mendukung fungsi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi, untuk bias menyediakan laporan kepada pihak luar sesuai yang diperlukan [3].

2.2.4 Metode *Prototype*

2.2.4.1 Pengertian *Prototype*

Metode *prototype* merupakan sebuah metode pengembangan sistem dimana inti tahapannya adalah komunikasi, pembuatan dan uji coba. Dalam tahapan komunikasi, pengembang dan user saling berinteraksi untuk menemukan solusi permasalahan, kemudian pengembang membuat rancangan secara cepat berdasarkan solusi yang telah dibahas pada tahap komunikasi, dan pada tahap akhir, pengembang dan user melakukan uji coba untuk mengetahui apakah solusi yang digunakan telah efektif dalam menyelesaikan masalah, jika masih kurang

maka kembali ketahap komunikasi dan perancangan, jika telah efektif maka solusi dapat digunakan dan dilanjutkan [4].



Gambar 2.2 Proses *Prototyping Model* [4]

Tahapan-tahapan umum dalam model *prototyping* adalah sebagai berikut [4]:

1. Mendengarkan Pelanggan

Pada proses ini pengembang dan pelanggan akan berinteraksi terlebih dahulu dan setelah itu menentukan tujuan umum, keperluan yang diketahui dan gambaran bagian-bagian yang akan diperlukan.

2. Membangun dan memperbaiki *Prototype*

Proses ini diawali dengan melakukan perancangan secara cepat, yaitu dimana perancangan ini bisa mewakili semua bagian software yang diketahui, dan rancangan ini menjadi patokan dalam membangun dan perbaikan *prototype*.

3. Pengujian *Prototype* oleh Pelanggan

Proses ini pelanggan akan melakukan pengujian *prototype* yang sudah dibangun secara cepat dan mengevaluasi *prototype* yang bertujuan memperjelas kebutuhan software.

2.2.4.2 Tujuan Prototype

Tujuan utama dalam pembuatan prototype adalah sebagai berikut:

1. Proses revisi dan pengujian terhadap produk dilakukan secara terus menerus, sehingga didapatkan produk yang sesuai dengan yang diinginkan oleh user. Proses testing dan revisi dapat dilakukan baik secara keseluruhan maupun partikal pada bagian dari produk.
2. Proses pengujian harus memiliki perbandingan baku (benchmark) sehingga menghasilkan produk yang secara empiris sehingga menghindari kegagalan produk atau terjadi perbedaan persepsi antara developer atau user.
3. Dengan proses testing dan komunikasi yang terus menerus antara user dan developer diharapkan dihasilkan produk yang user-friendly [5].

2.2.4.3 Manfaat Prototype

Manfaat prototype dalam pembuatan suatu software yaitu:

1. Mempermudah kita dalam membuat alur masalah.
2. Rancangan awal sebelum program dibuat, menjadikan gambaran aplikasi yang dibuat menjadi lebih terorganisasi.
3. Apabila terjadi suatu kesalahan, kita dapat melihat kembali prototype sebelumnya dan mengubahnya.

Dalam pembuatan prototype untuk suatu sistem memiliki beberapa keuntungan sebagai berikut:

1. Menghasilkan syarat yang lebih baik dari produksi yang dihasilkan oleh metode 'spesifikasi tulisan'.
2. User dapat mempertimbangkan sedikit perubahan selama masih bentuk prototype.
3. Memberikan hasil yang lebih akurat dari pada perkiraan sebelumnya, karena fungsi yang diinginkan dan kerumitannya sudah dapat diketahui dengan baik.
4. User merasa puas. Karena user dapat mengenal melalui komputer. Dengan melakukan prototipe (dengan analisis yang sudah ada), user belajar mengenai

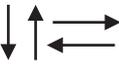
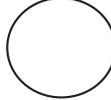
komputer dan aplikasi yang akan dibuatkan untuknya. Selain itu, user terlibat langsung dari awal dan memotivasi semangat untuk mendukung analisis selama proyek berlangsung[5].

2.2.5 Desain Sistem

2.2.5.1 Context Diagram

Context Diagram merupakan diagram yang menunjukkan batasan sistem informasi, yang merupakan pandangan dari sistem informasi dan hanya menunjukkan hubungan antara proses dengan entitas [6].

Tabel 2.1 Simbol *Context Diagram* [6]

| Simbol | Keterangan |
|---|--|
| Terminator  | Menunjukkan kesatuan luar Entity |
| Garis Alir  | Menunjukkan suatu proses untuk mengeluarkan input atau output. Menunjukkan aliran atau arus. |
| Proses  | Menunjukkan proses sebagai lambang sistem |

2.2.5.2 Decomposition

Dekomposisi merupakan bentuk grafik yang dapat dipecah menjadi beberapa bagian yang kecil sehingga mudah dipelajari [6].

Fungsi dari dekomposisi antara lain:

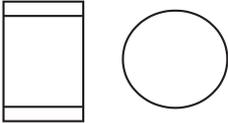
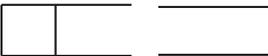
1. Memecahkan permasalahan yang besar ke dalam bagian- bagian yang lebih kecil yang bisa dipecah.
2. Untuk membantu testing program.

3. Untuk membantu penggambaran aliran.
4. Untuk membantu di dalam melacak proses terkecil sampai tinggi.

2.2.5.3 Data Flow Diagram (DFD)

Data Flow Diagram adalah alat yang digunakan untuk menggambarkan arus data dari sistem, tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir dan disimpan. DFD memudahkan pemakai bagi yang belum menguasai bidang komputer untuk mengerjakan sebuah sistem [6].

Tabel 2.2 Simbol Data Flow Diagram (DFD) [6]

| Simbol | Keterangan |
|--|--|
| Entitas  | Sumber atau tujuan dari aliran data dari atau ke sistem. |
| Proses  | Mentransformasikan data secara umum. Bagian atas berisi nomor untuk identitas proses, bagian tengah berisi fungsi dari proses, dan bagian bawah bersifat opsional. |
| Penyimpanan  | Menyimpan data atau <i>file</i> . |
| Aliran Data  | Menggambarkan aliran data dari satu proses lainnya. |

2.2.5.4 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD), penggambaran hubungan antara satu entitas dengan entitas lain yang memiliki sejumlah atribut yang sama dengan entitas yang

lain. Di dalam suatu sistem terintegrasi oleh perancang untuk memodelkan data yang nanti akan dikembangkan menjadi basis data [7].

Tabel 2.3 Simbol *Entity Relationship Diagram* [7]

| Simbol | Keterangan |
|--------|-------------------------------------|
| = | Terdiri dari |
| + | Dan |
| () | Pilihan (boleh ada atau tidak) |
| { } | Iterasi/pengulangan |
| [] | Pilih salah satu pilihan |
| | Pemisah pilihan di dalam simbol [] |
| * | Keterangan/catatan |
| @ | Penunjuk (<i>key field</i>) |

2.2.5.5 Normalisasi

Normalisasi adalah teknik yang digunakan dalam membuat desain *logic basis data relation* dengan menerapkan sejumlah aturan yang bertujuan untuk menghasilkan struktur tabel yang normal [6].

Aturan dalam masing-masing bentuk normalisasi tersebut adalah sebagai berikut:

1. Bentuk tidak normal

Bentuk ini merupakan kumpulan data yang direkam dan tidak ada keharusan mengikuti suatu format tertentu.

2. Bentuk normal Pertama

Suatu tabel dikatakan dalam bentuk normal pertama (1NF) jika dan hanya jika setiap atribut dari relasi tersebut hanya memiliki nilai tunggal (*scalar value*) dalam satu baris atau *record*.

3. Bentuk normal kedua

Suatu tabel berada dalam bentuk normal kedua (2NF) jika dan hanya jika relasi tersebut memenuhi normal pertama dan setiap atribut yang bukan kunci (*nonkey*) bergantung secara fungsional (FD) secara utuh pada kunci utama.

4. Bentuk normal ketiga

Suatu tabel berada dalam bentuk normal ketiga (3NF) jika dan hanya jika relasi tersebut memenuhi normal kedua dan setiap atribut yang bukan kunci (*nonkey*) tidak mempunyai *transitive functional dependency* kepada kunci utama (*primary key*).

2.2.5.6 Tabel Relation

Relasi menyatakan sebuah table dalam basis data, sedangkan relasi menyatakan hubungan antar relasi dalam basis data. Hubungan antar relasi ditunjukkan dengan gambar *schema* atau *subschema* dalam basis data, Kardinalitas relasi yang terjadi di antara dua himpunan entitas dapat berupa [6]:

1. Satu ke satu (*one to one*), berarti setiap entitas pada himpunan entitas A berhubungan dengan satu entitas pada himpunan entitas B, dan begitu sebaliknya.
2. Satu ke Banyak (*one to many*), berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi setiap entitas pada himpunan entitas B berhubungan paling banyak dengan satu entitas pada himpunan entitas A.
3. Banyak ke Satu (*many to one*), berarti setiap entitas pada himpunan entitas A dapat berhubungan paling banyak dengan satu entitas pada himpunan entitas B, tetapi setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B.
4. Banyak ke Banyak (*many to many*), berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, dan begitu sebaliknya.

2.2.5.7 Data Dictionary

Data *Dictionary* adalah kumpulan simbol-simbol yang digunakan untuk membantu dalam pengidentifikasian setiap *field* dan file di dalam sistem [6]. Simbol-simbol yang digunakan dalam kamus data adalah sebagai berikut:

Tabel 2.4 Simbol Kamus Data [6]

| Simbol | Keterangan |
|--------|-------------------------------------|
| = | Terdiri dari |
| + | Dan |
| () | Pilihan (boleh ada atau tidak) |
| { } | Iterasi/pengulangan |
| [] | Pilih salah satu pilihan |
| | Pemisah pilihan di dalam simbol [] |
| * | Keterangan/catatan |
| @ | Penunjuk (<i>key field</i>) |

2.2.5.8 Desain Input Output

1. Desain Input

Desain *input* merupakan bagian yang paling penting dalam pembuatan program aplikasi, karena melalui desain ini pemakai akan berinteraksi dengan komputer. Desain *input* disesuaikan dengan bentuk-bentuk *form* dokumen asli yang digunakan sebagai data inputan. Hal-hal yang perlu diperhatikan dalam desain input adalah tipe input, fleksibilitas format, dan kemudahan untuk digunakan [6].

2. Desain Output

Desain *output* merupakan bagian yang penting, karena digunakan untuk menjawab kebutuhan pemakai untuk bentuk informasi yang diinginkan. *Output* berupa laporan yang dapat digunakan untuk menentukan format dan

tata letak laporan. Desain *output* dibedakan menjadi dua, *output* internal untuk mendukung kegiatan manajemen yang ada di perusahaan, sedangkan *output* eksternal akan didistribusikan ke pihak luar yang membutuhkan [6].

2.2.6 Tinjauan Umum tentang Web

2.2.6.1 Internet

Internet adalah suatu interkoneksi sebuah jaringan komputer yang dapat memberikan layanan informasi secara lengkap. Dan, terbukti bahwa internet dilihat sebagai media maya yang dapat menjadi rekan bisnis, politik, sampai hiburan. Semuanya tersaji dengan lengkap di dalam media ini [8].

2.2.6.2 File Transfer Protocol (FTP)

File Transfer Protocol adalah protokol yang memiliki fungsi dalam pertukaran file di suatu jaringan komputer yang mendukung protokol TCP/IP. Pengertian lain menyebutkan bahwa FTP merupakan salah satu protokol internet yang mengoperasikan tugas di dalam level aplikasi dengan standar untuk proses transfer file antar komputer dalam sebuah *framework*. Sebuah protokol yang pertama kali dikembangkan dan masih digunakan oleh banyak pengguna sampai saat ini. Protokol ini membuat pengguna atau user bisa berkomunikasi dengan cara saling transfer data dari satu perangkat komputer ke perangkat komputer lainnya [8]

2.2.6.3 Website

Website dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data animasi, data gambar diam atau gerak, suara, video dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*). Bersifat statis apabila isi informasi *website* tetap, jarang berubah, dan isi informasinya searah hanya dari pemilik *website*. Bersifat dinamis apabila isi informasi *website* selalu berubah-ubah, dan isi informasinya interaktif dua arah berasal dari pemilik serta

pengguna *website*. Contoh *website* statis adalah berisi profil perusahaan, sedangkan *website* dinamis adalah seperti *Friendster*, *Multiply*, dll. Dalam sisi pengembangannya, *website* statis hanya bisa diupdate oleh pemiliknya saja, sedangkan *website* dinamis bisa diupdate oleh pengguna maupun pemilik [8].

2.2.6.4 *HyperText Markup Language (HTML)*

HyperText Markup Language (HTML) sebuah bahasa untuk menentukan struktur dokumen untuk pengambilan di Internet menggunakan program *browser WorldWideWeb* tersebut. Selain itu HTML sebuah aplikasi dari *Standard Generalized Markup Language (SGML)* yang merupakan Standar Internasional (ISO 8879) untuk teks markup. Prinsipnya adalah bahwa teks markup berkonsentrasi pada struktur daripada penampilan, membuat file lebih nyata dan meninggalkan rincian visual untuk perangkat lunak pengguna akhir (Browser) [9].

2.2.6.5 *Hyper Text Preprosesor (PHP)*

PHP adalah bahasa pemrograman untuk membuat situs web dinamis dan interaktif. PHP berjalan di server web dan melayani pengunjung dengan halaman web sesuai permintaan. PHP merupakan bahasa yang sederhana namun kuat dirancang untuk membuat konten HTML. Sejak dimulai pada tahun 1994, karena keunggulan popularitas bahasa dan kemudahan penggunaan PHP telah mampu menyaingi bahasa web. PHP pertama kali diciptakan oleh Rasmus Lerdorf. PHP biasa digunakan untuk membangun web dinamis dimana proses keseluruhan berjalan pada *web server* dan hasilnya ditampilkan pada web browser [9].

2.2.6.6 *Mysql*

MySQL merupakan database *server* yang pertama kali didukung oleh bahasa pemrograman *script* untuk internet (PHP dan Perl). MySQL lebih sering digunakan untuk membangun aplikasi berbasis web yang menggunakan *database* sebagai sumber dan pengelolaan datanya, umumnya menggunakan bahasa pemrograman *script* PHP. MySQL bersifat *open source* dan *free* pada berbagai *platform*. Keunggulan MySQL dengan menggunakan SQL sebagai bahasa dasar

untuk mengakses database sehingga mudah digunakan, kinerja *query* cepat, serta mampu mencukupi kebutuhan database perusahaan skala kecil-menengah [9].

2.2.6.7 Cascading Style Sheet (CSS)

CSS adalah bahasa pemrograman web yang mengatur komponen dalam suatu web supaya web lebih terstruktur dan lebih seragam. CSS seperti halnya *styles* pada aplikasi pengolahan kata *Microsoft Word* yang bisa mengatur beberapa style, misalnya subbab, *heading*, *bodytext*, *footer*, *images*, dan *style* lainnya untuk dapat dipakai bersama-sama dalam beberapa file. CSS biasanya digunakan untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML. CSS mampu mengatur warna body teks, ukuran gambar, ukuran *border*, warna *mouse over*, warna tabel, warna *hyperlink*, margin kiri/kanan/atas/bawah, spasi antar paragraf, spasi antar teks, dan parameter lainnya. Dengan adanya CSS dapat memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda [9].

2.2.6.8 Javascript

JavaScript adalah bahasa pemrograman berbasis *prototype* yang berjalan disisi klien. Dalam konteks web, *JavaScript* sebagai bahasa pemrograman yang berjalan khusus untuk di browser atau halaman web agar halaman web menjadi lebih hidup. *JavaScript* terdiri dari dua suku kata, yaitu *Java* dan *Script*. *Java* adalah Bahasa pemrograman berorientasi objek, sedangkan *Script* adalah serangkaian instruksi program. Secara fungsional, *JavaScript* digunakan untuk menyediakan akses *script* pada objek yang dibenamkan (*embedded*). Contoh sederhana dari penggunaan *JavaScript* yaitu seperti membuka halaman *pop up*, fungsi validasi pada *form* sebelum data dikirimkan ke *server*, merubah *image cursor* ketika melewati objek tertentu, dan lain lain [9].

2.2.7 Pengujian

2.2.7.1 Pengujian *Black Box*

Pengujian *black-box* berkaitan dengan pengujian yang dilakukan pada antarmuka perangkat lunak. Meskipun dirancang bertujuan untuk mengungkap kesalahan, pengujian *black-box* digunakan agar dapat memperlihatkan bahwa fungsi-fungsi perangkat lunak dapat beroperasi, bahwa input diterima dengan baik dan output dihasilkan dengan tepat, dan integritas informasi eksternal (seperti file data) dipelihara [10].

Pengujian *black-box* menguji beberapa aspek dasar suatu sistem dengan memperhatikan sedikit struktur logika internal perangkat lunak tersebut. Pengujian ini fokus kepada persyaratan fungsional perangkat lunak. Pengujian ini memungkinkan pelaku RPL mendapatkan serangkaian kondisi input yang memenuhi persyaratan fungsional suatu program. Pengujian ini berusaha menemukan kesalahan dengan kategori sebagai berikut:

1. Fungsi-fungsi yang salah atau hilang.
2. Kesalahan antarmuka.
3. Kesalahan struktur data atau akses basisdata eksternal.
4. Kesalahan kinerja.
5. Kesalahan inisialisasi atau terminasi.

2.2.7.2 Pengujian *White Box*

White box testing adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Secara sekilas dapat diambil kesimpulan *white box* testing merupakan petunjuk untuk mendapatkan program yang benar secara 100% [10]. Tujuan dari Pengujian *white box* adalah:

1. Untuk mengetahui cara kerja dari suatu perangkat lunak secara internal.

2. Untuk menjamin operasi-operasi internal sesuai dengan spesifikasi yang telah ditetapkan dengan menggunakan struktur kendali dari prosedur yang dirancang.