

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem Informasi

Penelitian ini tidak terlepas dari teori-teori yang mendukung kemudahan dalam mempelajari serta merancang sistem informasi pengadaan barang atau jasa yang diharapkan akan berfungsi secara maksimal. Berikut ini adalah teori pendukung yang memperkuat penulisan ini.

2.1.1 Pengertian Sistem

Sistem dapat didefinisikan dengan pendekatan prosedur dan dengan pendekatan komponen, menurut Jogiyanto H.M (2010:34) adalah sebagai berikut :

1. Dengan sebuah pendekatan prosedur, sistem mampu didefinisikan sebagai kumpulan dari prosedur - prosedur yang mempunyai sebuah tujuan tertentu.
2. Dengan dilakukannya pendekatan komponen, maka sistem dapat didefinisikan sebagai kumpulan dari komponen yang saling berhubungan satu dengan yang lainnya dan membentuk satu kesatuan untuk mencapai tujuan tertentu.

Dikutip dari sebuah buku karya Bambang Hartono (2013:10). Menurut Bonita J. Campbel (1979) dalam buku *Understanding Information System: Foundations for control* menegaskan bahwa sistem adalah Sehimpunan bagian -

bagian atau komponen yang saling berkaitan dan secara bersama-sama berfungsi atau bergerak untuk mencapai suatu tujuan.

2.1.2 Karakteristik Sistem

Pada suatu sistem memiliki ciri-ciri karakteristik yang terdapat pada sekumpulan elemen yang harus bias dipahami dalam megidentifikasi pembuatan sistem. Adapun karakteristik sistem menurut Hutahaeen (2015:3) yang dimaksud adalah sebagai berikut :

1. **Komponen**

Sistem terdiri dari sejumlah komponen yang saling berinteraksi dan bekerja sama untuk membentuk sebuah satu kesatuan. Komponen sistem dapat berupa sub sistem atau bagian-bagian dari sistem tersebut.

2. **Batasan Masalah**

Sistem terdiri dari sejumlah komponen yang saling berinteraksi dan bekerja sama untuk membentuk sebuah satu kesatuan. Komponen sistem dapat berupa sub sistem atau bagian-bagian dari system tersebut.

3. **Lingkungan luar sistem (*environment*)**

Apapun yang berada di luar batas dari sistem dan mempengaruhi sistem tersebut dinamakan dengan lingkungan luar sistem. Lingkungan luar yang bersifat menguntungkan, wajib dipelihara dan yang merugikan harus dikendalikan agar tidak mengganggu keberlangsungan system itu sendiri.

4. **Penghubung sistem (*interface*)**

Media penghubung diperlukan untuk mengalirkan sumber-sumber daya dari sub sistem ke sub sistem lainnya dinamakan dengan penghubung sistem.

5. Masukkan sistem (*input*)

Energi yang dimasukkan ke dalam sistem dinamakan dengan masukan sistem (*input*) dapat berupa perawatan dan masukan sinyal. Perawatan ini berfungsi agar sistem dapat beroperasi dan masukan sinyal adalah energi yang diproses untuk menghasilkan keluaran (*output*).

6. Keluaran sistem (*output*)

Hasil dari energi yang telah diolah dan diklasifikasikan menjadi keluaran yang berguna dinamakan dengan keluaran sistem (*output*). Informasi merupakan contoh keluaran sistem.

7. Pengolah sistem

Untuk mengolah masukan menjadi keluaran diperlukan suatu pengolah yang dinamakan dengan pengolah sistem.

8. Sasaran sistem

Sistem pasti memiliki tujuan atau sasaran yang sangat menentukan input yang dibutuhkan oleh sistem dan keluaran yang dihasilkan.

2.1.3 Klasifikasi Sistem Informasi

Menurut Hutahaean, (2015:6) sebuah sistem merupakan suatu bentuk integrasi antara satu komponen dan komponen lain karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi di dalam sistem tersebut.

Oleh karena itu sistem dapat diklasifikasikan dari beberapa sudut pandang. Adapun klasifikasi sistem yang diuraikan adalah sebagai berikut:

1. Sistem Abstrak dan Sistem Fisik

Sistem abstrak merupakan sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem telogi. Sedangkan sistem fisik diartikan sebagai sistem yang nampak secara fisik sehingga setiap makhluk dapat melihatnya, misalnya sistem komputer.

2. Sistem Alamiah dan Sistem Buatan Manusia

Sistem alamiah merupakan sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem tata surya, sistem galaksi, sistem reproduksi dan lain-lain. Sedangkan sistem buatan manusia merupakan sistem yang dirancang oleh manusia. Sistem buatan yang melibatkan interaksi manusia, misalnya sistem akuntansi, sistem informasi, dan lain-lain.

3. Sistem Deterministik dan Sistem Probabilistik

Sistem deterministik merupakan sistem yang beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan, misalnya sistem komputer, adalah contoh sistem yang tingkah lakunya dapat dipastikan berdasarkan program-program komputer yang dijalankan. Sedangkan sistem robabilistik merupakan sistem yang

kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas, misalnya sistem manusia

4. Sistem Terbuka dan Sistem Tertutup

Sistem terbuka merupakan sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Lebih spesifik dikenal juga yang disebut dengan sistem terotomasi, yang merupakan bagian dari sistem buatan manusia dan berinteraksi dengan kontrol oleh satu atau lebih komputer sebagai bagian dari sistem yang digunakan dalam masyarakat modern. Sistem ini menerima masukan dan menghasilkan keluaran untuk subsistem lainnya, misalnya sistem kebudayaan manusia. Sedangkan sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa campur tangan dari pihak luar. Secara teoritis sistem tersebut ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system* (secara relative tertutup, tidak benar-benar tertutup).

2.1.4 Pengertian Informasi

Informasi merupakan hal yang sangat penting bagi perusahaan dalam mengambil setiap pengambilan keputusan. Secara Etimologi, Informasi berasal dari bahasa Perancis kuno yaitu *informaction* (tahun 1387) yang diambil dari bahasa latin *informationem* yang berarti “garis besar, konsep, ide”

Sedangkan menurut *Romney dan Steinbart (2015:4)*, Informasi adalah data yang telah dikelola dan diproses untuk memberikan arti dan memperbaiki proses pengambilan keputusan. Sebagaimana perannya, pengguna membuat

keputusan yang lebih baik sebagai kuantitas dan kualitas dari peningkatan informasi. Dengan demikian dapat disimpulkan bahwa informasi adalah data yang diproses menjadi suatu bentuk yang lebih berguna dan berarti bagi yang menerimanya dalam aktivitas pembuatan keputusan.

2.1.5 Pengertian Sistem Informasi

Definisi sistem menurut (Anggraeni, 2017:1) “Sistem adalah kumpulan orang yang saling bekerja sama dengan ketentuan-ketentuan aturan yang sistematis dan terstruktur untuk membentuk satu kesatuan yang melaksanakan suatu fungsi untuk mencapai tujuan.”

2.2 Pengertian Rancang

Rancang bangun sangat berkaitan dengan perancangan sistem yang merupakan satu kesatuan untuk merancang dan membangun sebuah aplikasi.

Menurut (Sutabri, 2012), rancang adalah kegiatan yang memiliki tujuan untuk mendesain sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi perusahaan yang diperoleh dari pemilihan alternatif sistem yang terbaik,

2.3 Pengertian Bangun

Menurut (Pressman 2010:89), bangun adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun sebagian.

2.4 Pengertian Rancang Bangun

Menurut Bambang (2013:27), rancang bangun adalah proses pembangunan sistem untuk menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun hanya sebagian.

2.5 Konsep Dasar Internet

Menurut Ahmadi dan Hermawan (2013), Internet adalah komunikasi jaringan komunikasi global yang menghubungkan seluruh komputer di dunia meskipun berbeda sistem operasi dan mesin

2.5.1 Pengertian Website

Menurut Murad (2013), *Website* adalah sistem dengan informasi yang disajikan dalam bentuk teks, gambar, suara, dan lainnya yang tersimpan dalam sebuah server web internet yang disajikan dalam bentuk *hypertext*. Dapat disimpulkan, *website* adalah sebuah tempat di internet, yang menyajikan informasi dengan berbagai macam format data seperti *text*, *image*, bahkan video dan dapat diakses menggunakan berbagai aplikasi klien sehingga memungkinkan penyajian informasi yang lebih menarik dan dinamis dengan pengelolaan yang terorganisasi. *Website* memiliki teknologi yang dikenal sebagai *web browser*, *web hosting*, dan *web server*.

1. Web Browser

Sejarah Web browser dimulai pada tahun 1991 saat tim Berners-Lee membuat aplikasi *web browser* pertama pada computer NeXT dengan nama *World Wide Web Browser*, kemudian di tahun 1993 NCSA (*National Center Supercomputing Application*) mengembangkan *web*

browser grafis NCSA Mosaic, yang kemudian dilanjutkan pada tahun 1994 merilis Netscape Navigator dan pada tahun 1998 berubah ,menjadi Mozilla Firefox. *Web Browser* berfungsi untuk memakai atau meminta *service*. Bahasa pemrograman yang digunakan yaitu PHP dan ASP. 2.

2. *Web Server*

Menurut Anhar (2010), *Web server* adalah aplikasi yang berfungsi untuk melayani permintaan pemanggilan alamat dari pengguna melalui *web browser*, dimana *web server* mengirimkan kembali informasi yang diminta tersebut melalui HTTP (*Hypertext Transfer Protocol*) untuk ditampilkan ke layar monitor komputer kita. Agar kita dapat mengubah isi dari *website* yang dibuat, kita membutuhkan program PHP. *Script-script* PHP tersebut yang berfungsi membuat halaman *website* menjadi dinamis. Dinamis artinya pengunjung *web* dapat memberikan komentar saran atau masukan pada *website* kita. *Website* yang kita buat menjadi lebih hidup karena ada komunikasi antara pengunjung dan kita sebagai *web* masternya.

3. *Web Hosting*

Pengertian *web hosting* dapat diartikan sebagai ruangan yang terdapat dalam harddisk tempat penyimpanan berbagai data, file-file, gambar dan lain sebagainya yang akan ditampillkan di *website*. Besarnya data yang dapat dimasukan tergantung dari besarnya *web hosting* yang disewa atau dimiliki, semakin besar *web hosting* semakin besar pula data yang dapat dimasukkan dan ditampilkan dalam *website*. *Web*

hosting juga diperoleh dengan menyewa besarnya *hosting* ditentukan ruangan *harddisk* dengan ukuran MB (*Mega Byte*) atau GB (*Giga Byte*). Waktu penyewaan *web hosting* rata-rata dihitung per tahun. Penyewaan *hosting* dilakukan dari perusahaan-perusahaan penyewa *web hosting* yang banyak dijumpai baik Indonesia maupun luar negeri.

2.6 Perangkat Lunak Yang Digunakan

2.6.1 XAMPP

Menurut Nugroho (2013), “XAMPP adalah paket program web lengkap yang dapat Anda pakai untuk belajar pemrograman web, khususnya PHP dan MySQL”. Dengan menggunakan paket software instalasi ini, maka sudah dapat melakukan beberapa instalasi software pendukung *webserver*, yaitu Apache, PHP, *phpMyAdmin*, dan *database* MySQL. Xampp merupakan tool paket perangkat lunak yang menggabungkan Apache, PHP, dan MySQL dalam satu paket aplikasi. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program apache HTTP Server, MySQL *database*, dan penterjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis. Untuk mendapatkannya dapat mendownload langsung melalui *web* resmi.

2.6.2 PHP

Menurut Arief (2011) PHP 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 akan dikirimkan ke *browser* dengan format HTML.

2.6.3 HTML (*Hypertext Markup Language*)

Menurut Arief (2011) “HTML merupakan salah satu format yang digunakan dalam pembuatan dokumen atau aplikasi yang berjalan di halaman web”. Pada dokumen HTML yang termasuk sistem *hyperteks*, kita tidak harus membaca dokumen tersebut secara urut dari atas ke bawah atau sebaliknya tetapi kita dapat menuju topik tertentu secara langsung dengan menggunakan teks penghubung yang akan membawa anda ke suatu topik atau dokumen lain secara langsung. Dokumen ini umumnya berisi informasi atau *interface* aplikasi di dalam internet. Ada dua cara untuk membuat sebuah *web page* yaitu dengan HTML editor atau dengan editor teks biasa seperti *notepad*.

2.6.4 CSS (*Cascading Style Sheet*)

Menurut Sibero (2013) menyatakan bahwa, “*Casading Style Sheet* memiliki arti Gaya Menata Halaman Bertingkat, yang artinya setiap satu elemen yang telah diformat dan memiliki anak dan telah diformat, maka anak dari elemen tersebut secara otomatis mengikuti format *element* induknya”.

2.6.5 Basis Data (Data Base)

Menurut Connolly dan Begg (2010), basis data adalah sebuah kumpulan data yang secara logis terkait dan dirancang untuk memenuhi suatu kebutuhan informasi dari sebuah organisasi.. Adapun tujuan basis data diantaranya sebagai efisiensi yang meliputi *speed, space & accuracy* menangani data dalam jumlah besar, kebersamaan pemakaian, dan meniadakan duplikasi.

2.6.6 MySQL (*My Structure Query Language*)

Menurut Raharjo (2011:21), MySQL merupakan RDBMS (atau server database) yang mengelola database dengan cepat menampung dalam jumlah sangat besar dan dapat di akses oleh banyak user.

MySQL dikembangkan oleh sebuah perusahaan swedia yang saat itu bernama TcX Data Konsult AB, dan pada akhirnya berubah nama menjadi MySQL AB.

Sekitar tahun 1994-1995, TcX membuat *database* MySQL untuk mengembangkan aplikasi klien-nya. TcX merupakan perusahaan pengembang *software* dan konsultan database.

Michael Widenius atau disebut “Monty” adalah pengembang satu-satunya di TcX. MySQLAB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU/GPL (*General Public License*), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan pengguna GPL.

2.6.7 Laravel

Laravel adalah sebuah *framework* PHP 5.3 bersifat *open source* yang ditulis oleh Taylor Otwell dengan lisensi dibawah MIT License. Laravel dibuat untuk membantu para developer khususnya dalam membuat sebuah web dengan sintak

yang sederhana, elegan, ekspresif dan menyenangkan. Dengan Laravel, tugas-tugas umum developer dapat dikurangi pada sebagian besar proyek-proyek web seperti *routing*, *session* dan *caching*. Disamping itu, Laravel berusaha menggabungkan pengalaman-pengalaman *development* dalam bahasa lain, seperti Ruby on Rails, ASP.NET, MVC dan Sinatra (Nugraha, 2014).

Laravel termasuk *Best PHP Framework 2018* versi *coders-eye*. Sesuai dengan motto Laravel “*PHP doesn’t hurt, code happy & enjoy the fresh air*”, tujuan utamanya adalah untuk mempermudah *coding* dalam membuat sebuah produk web. Ini membuktikan bahwa menggunakan Laravel memang dapat mempercepat dan mempermudah *development* website (Nugraha, 2014). Laravel pertama versi beta dirilis pada tanggal 9 Juni 2011, diikuti Laravel 1 yang dirilis pada bulan yang sama. Saat ini, Laravel sudah merilis versi 5.6 pada tanggal 7 Februari 2018.

Sebelum mulai menggunakan Laravel, terdapat beberapa kebutuhan yang harus disediakan diantaranya:

1. *Text Editor*

Terdapat banyak *tools* yang dapat digunakan sebagai *text editor*, misalnya Sublime Text, Notepad++ dan lain-lain. Dapat dipilih sesuai dengan kebutuhan atau selera dan biasanya terdapat kelebihan masing-masing.

2. *Websserver dan Database*

Yang terpenting dalam instalasi Laravel yaitu bahwa versi minimal adalah PHP 5.3 ke atas dan sudah diinstal ekstensi PHP yaitu Mcrypt. XAMPP versi 3.2.2 dapat digunakan karena telah mendukung PHP 5.4 dan Mcrypt.

3. Composer

Composer adalah sebuah *dependency* ‘manager’ untuk PHP. Composer ini dapat menambah *library* yang dibutuhkan untuk website secara otomatis tanpa perlu men-*download* satu persatu. Mirip dengan *apt-get install* pada sistem operasi Linux.

Laravel 5 memiliki struktur yang berbeda dari *framework-framework* PHP lainnya. Secara garis besar, kebanyakan *framework* PHP yang menganut pola MVC (*Model-View-Controller*) menggunakan skema direktori dengan nama “Model”, “View” dan “Controller” yang seluruhnya dikumpulkan kedalam sebuah direktori utama yang bernama “src” atau “app” / “application” (seperti Code Igniter). Pada skema tersebut, direktori “Model” digunakan untuk menyimpan class PHP yang berhubungan dengan model database, kemudian direktori “Controller” digunakan untuk menyimpan class PHP yang berhubungan dengan *application logic* dan direktori “View” digunakan untuk menyimpan file-file yang berhubungan tampilan aplikasi. Konvensi struktur direktori tersebut juga digunakan oleh Laravel versi 5 namun dengan struktur yang sedikit berbeda dari

biasanya dan hal ini akan membuat bingung para pengguna *framework* yang baru berpindah dari *framework* lain seperti Yii / Code Igniter.

Berikut adalah struktur direktori dari Laravel 5:

a. app/

Folder ini digunakan untuk menyediakan tempat *default* untuk menyimpan kode yang sudah ditulis atau dikonfigurasi. Pada folder ini lah kita meletakkan semua kode proyek aplikasi yang dimulai dari konfigurasi, *logic* dan sebagainya.

b. bootstrap/

Folder ini berisi tentang *file-file* prosedur untuk *framework* Laravel. Dalam folder ini terdapat beberapa *file* yang hanya boleh diedit oleh pengguna Laravel yang sudah berpengalaman.

c. vendor/

Folder ini berisi tentang semua paket Composer yang digunakan pada aplikasi, *file framework* Laravel juga terdapat dalam folder ini.

d. public/

Folder ini berisi *assets* yang digunakan untuk menyimpan *file-file* CSS, Javascript, *Image* ataupun *file-file* yang diperlukan (biasanya kaitannya erat dengan *View* untuk membangun *front-end*).

e. .gitattributes

File ini merupakan *file* konfigurasi standar sistem versi kontrol yang sangat populer saat ini yaitu Git.

f. `.gitignore`

File ini berisi beberapa informasi folder mana saja yang akan diabaikan oleh Git.

g. `artisan`

File ini merupakan *file* yang berfungsi untuk mengeksekusi atau menjalankan perintah-perintah artisan CLI untuk Laravel .

h. `composer.json` dan `composer.lock`

Kedua *file* ini berisi informasi tentang paket-paket Composer yang akan kita gunakan untuk keperluan aplikasi.

i. `phpunit.xml`

File ini berisi konfigurasi *default* Unit Testing PHP Laravel. *File* ini juga menangani pemuatan dependensi Composer dan mengeksekusi semua *test* yang ada pada direktori `app/tests`.

j. `server.php`

File ini berisi tentang prosedur untuk menjalankan Laravel dengan web server internal yang diperkenalkan pada PHP versi 5.4.

Adapun kelebihan Laravel adalah sebagai berikut:

a. Expressif

Laravel adalah sebuah framework PHP (Hypertext Preprocessor) yang expressif, artinya ketika melihat suatu syntax Laravel, seorang programmer diharapkan akan langsung mengetahui kegunaan dari syntax tersebut meskipun belum pernah mempelajarinya apalagi menggunakannya.

b. Sempel

Salah satu hal yang membuat Laravel menjadi simpel adalah adanya Eloquent ORM. Misalnya kita akan mengambil data yang ada pada tabel users, yang kita perlukan hanya membuat sebuah class model bernama User. Kemudian kita tinggal menuliskan syntax untuk mengambil semua data dari tabel users tersebut sebagai berikut :

```
$all_user = User::all();
```

Apabila yang akan kita ambil hanya data tertentu saja maka kita tinggal mengganti syntax menjadi :

```
$all_user=User::where('kelas','=','X MM 2')->get();
```

Dengan begitu, semua data dari tabel users akan dapat diakses dengan mudah melakukan looping terhadap variabel \$all_user menggunakan foreach pada tampilan viewnya.

b. Tersedia Composer

Apabila pengguna akan menginstall suatu library, ia dapat menginstallnya melalui composer dan composer akan secara otomatis menginstall library lain yang dibutuhkan, tanpa perlu mengunduh satu

persatu. Hal ini dapat dilakukan dengan menuliskan perintah “composer update” pada command prompt.

c. *Open Source*

Laravel adalah *framework* PHP yang bersifat *open source*, sehingga dalam penggunaannya tidak perlu membayar.

d. Kompatibel dengan PHP 5.3 keatas

Laravel dikembangkan secara khusus untuk PHP 5.3 keatas, jadi Laravel bisa memanfaatkan berbagai macam kelebihan yang dimiliki PHP versi 5.3 tersebut. Sebenarnya Laravel juga bisa digunakan pada PHP dengan versi dibawah 5.3 tetapi akan terdapat beberapa fungsi yang tidak dapat bekerja secara maksimal misalnya fungsi enkripsi *password*. Dengan di desain khusus untuk PHP 5.3 keatas maka Laravel bisa fokus memanfaatkan fitur-fitur baru PHP 5.3 tanpa terkendala *backward compability*.

e. Dokumentasi yang lengkap

Laravel menyediakan dokumentasi yang lengkap mulai dari proses instalasi, interaksi *database*, dan artisan. Artisan adalah perintah dalam command prompt yang dapat menampilkan perintah yang akan membantu dalam pengembangan aplikasi. Adanya dokumentasi yang lengkap akan memudahkan bagi pemula yang akan mempelajari pengembangan web dengan Laravel.

f. Prinsip Model View Controller (MVC)

Laravel menganut konsep MVC (model, view, controller). Framework PHP yang menganut konsep MVC menjadikan pembuatan aplikasi

yang kompleks dan berskala besar menjadi lebih cepat dan mudah. Di dalam MVC, pembuatan aplikasi dipisahkan antara tampilan, logic program, dan model. File Model mengatur interaksi antara database dan aplikasi yang dibuat. File View mengatur tampilan atau user interface dalam aplikasi. File Controller mengatur hubungan antara file Model dan View.

2.7 Restorasi

Restorasi adalah suatu kegiatan perbaikan yang sudah rusak agar dapat dipergunakan lagi dalam keadaan utuh dan lengkap. Restorasi, disebut pula reparasi atau perbaikan. Restorasi pada bidang otomotif biasanya mencakup keseluruhan komponen yang dimana nantinya sebuah kendaraan akan memiliki tampilan yang lebih baik.

Sedangkan menurut (Purwani, 2013) Restorasi adalah tindakan perbaikan bahan yang mengalami rusak parah agar kembali pada kondisi semula.

2.8 RAD (*Rapid Application Development*)

Rapid Application Development (RAD) adalah salah satu metode pengembangan suatu sistem informasi dengan waktu yang relatif singkat. Untuk pengembangan suatu sistem informasi yang normal membutuhkan waktu minimal 180 hari, akan tetapi dengan menggunakan metode RAD suatu sistem dapat diselesaikan hanya dalam waktu 30-90 hari.

Pendapat lain RAD merupakan metode iteratif (berulang) untuk mengembangkan sistem dalam *working model* (model bekerja) mengkonstruksikan

sistem pada awal pengembangan bertujuan menetapkan kebutuhan (*requirement*) pengguna dan selanjutnya diabaikan (Aswati, 2016).

Penelitian ini menggunakan metode RAD (*Rapid Application Development*) memodifikasi tahapan atau *phase*. (Kendall, J.E. & Kendall, K.E. 2010) ada tiga *phase* yaitu:

1. *Requirements Planning* (Perencanaan Syarat-Syarat)
2. *Design Workshop* (Workshop Desain)
3. *Implementation* (Implementasi)

Penelitian ini fokus pada desain dari sistem informasi sehingga akan memodifikasi dua *phase* yaitu *Requirements Planning* dan *Design Workshop*.



Gambar 2.1 Siklus RAD

(Sumber : Kendall 2010)

5. *Requirements Planning*

Requirements planning adalah proses melakukan pengumpulan bahan atau data yang sesuai dengan penelitian yang akan dikembangkan. Masukan dari tahapan ini adalah Dokumentasi dari beberapa prosedur yang sudah berjalan, observasi langsung dan

wawancara terhadap pengguna sistem informasi. Metode Observasi adalah pengumpulan data dengan pengamatan langsung pada objek penelitian untuk mendapatkan informasi sebagai bahan dan metode wawancara merupakan tanya jawab langsung pada pihak-pihak yang langsung terlibat atau berkaitan dengan objek yang diteliti (Kurniawan, 2015). Sehingga dalam proses *requirements planning* akan melakukan pengolahan data dari hasil wawancara yang dilakukan.

6. *Design Workshop*

Proses User Design Phase melewati beberapa tahapan yaitu perencanaan, analisa dan desain atau perancangan. Perancangan sistem informasi yang akan dibangun bersifat *object oriented*. Data hasil dari tahapan observasi dan metode yang akan digunakan pada tahap *user design*. Hasil atau *Output* dari tahapan ini berupa informasi *actor* dan *use case* yang terlibat, informasi interaksi antar kelas, *actor* dan objek, informasi fitur apa saja yang akan diterapkan pada pengembangan sistem informasi selanjutnya serta desain atau gambaran dari aplikasi tersebut.

7. *Implementation* (Implementasi)

Pada fase implementasi ini, penganalisis bekerja dengan para pengguna secara intens selama workshop dan merancang aspek-aspek bisnis dan nonteknis perusahaan. Segera setelah aspek-aspek ini disetujui dan sistem-sistem dibangun dan disaring, sistem-sistem baru

atau bagian dari sistem diujicoba dan kemudian diperkenalkan kepada organisasi (Kendall, 2010).

2.9 Peralatan Pendukung Sistem

2.9.1 UML (Unified Modelling Language)

Menurut Nugroho (2010:6), UML (*Unified Modeling Language*) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma (berorientasi objek).” Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

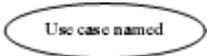
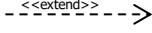
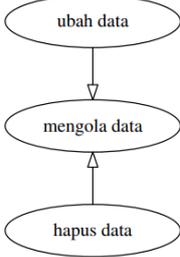
Berdasarkan pendapat yang dikemukakan Nugroho tersebut dapat ditarik kesimpulan bahwa UML adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis Objek (*Object Oriented programming*).

2.9.2 Model Diagram UML

1. Use Case Diagram

Use case diagram 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* di gunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Rosa A.S-M.Shalahuddin, 2011:137).

Tabel 2.1 Notasi *Use Case Diagram*

No	Simbol	Keterangan
1.	<p data-bbox="555 517 676 551"><i>Use Case</i></p> 	<p data-bbox="831 441 1417 689">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 use case.</p>
2.	<p data-bbox="555 837 632 871"><i>Actor</i></p> 	<p data-bbox="831 719 1417 1133">Orang, proses, atau sistem lain yang berinteraksi 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 di awal frase nama aktor.</p>
3.	<p data-bbox="512 1205 667 1238"><i>Association</i></p> 	<p data-bbox="831 1158 1417 1301">Komunikasi antara aktor dengan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.</p>
4.	<p data-bbox="555 1384 651 1417"><i>Extend</i></p> 	<p data-bbox="831 1323 1417 1525">Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu.</p>
5.	<p data-bbox="512 1704 711 1738"><i>Generalization</i></p> 	<p data-bbox="831 1547 1417 1693">Hubungan generalisasi dan spesialisasi (umum-khusus) antar dua buah use case dimana fungsi yang satu adalah fungsi</p> 

		yang lebih umum dari lainnya. Misalnya :
6.	<p style="text-align: center;"><i>Include</i></p> <p style="text-align: center;">--<<include>>--></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.

2. Skenario

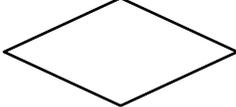
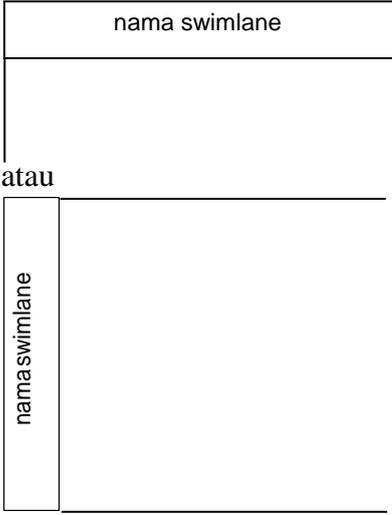
Menurut Yasin (2012), skenario adalah sebuah dokumentasi terhadap kebutuhan fungsional dari sebuah sistem. Form skenario merupakan penjelasan penulisan use case dari sudut pandang aktor.

3. Activity Diagram

Rosa dan Shalahuddin (2013:161), Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Berikut adalah simbol- simbol yang ada pada *activity diagram* dapat dilihat pada tabel berikut :

Tabel 2.2 Notasi *Activity Diagram*

No.	Simbol	Keterangan
1.		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.

2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	Swimlane 	Swimlane memisahkan organisasi bisnis yang bertanggungjawab terhadap aktivitas yang terjadi.

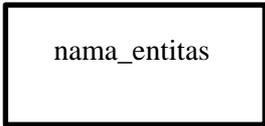
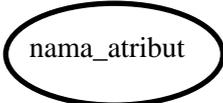
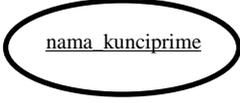
4. *Entity Relationship Diagram*

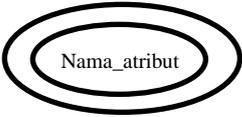
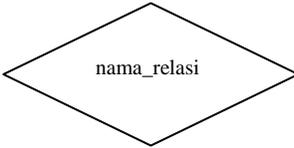
Menurut Sutanta (2011:91), Entity Relationship Diagram (ERD) merupakan suatu model data yang dikembangkan berdasarkan objek.”

Entity Relationship Diagram (ERD) digunakan untuk menjelaskan hubungan antar data dalam basis data kepada pengguna secara logis.

Entity Relationship Diagram (ERD) didasarkan pada suatu persepsi bahwa real world terdiri atas obyek-obyek dasar tersebut. Penggunaan Entity Relationship Diagram (ERD) relatif mudah dipahami, bahkan oleh para pengguna yang awam. Bagi perancang atau analis sistem, Entity Relationship Diagram (ERD) berguna untuk memodelkan sistem yang nantinya, basis data akan di kembangkan. Model ini juga membantu perancang atau analis sistem pada saat melakukan analisis dan perancangan basis data karena model ini dapat menunjukkan macam data yang dibutuhkan dan kerelasian antar data di dalamnya.

Tabel 2.3 Simbol-simbol *Entity Relationship Diagram*

No	Simbol	Keterangan
1.	Entitas/ <i>entity</i> 	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.
2.	Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas
3.	Atribut kunci primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).

4.	Atribut multinilai <i>multivalve</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
5.	Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.
6.	Asosiasi / <i>association</i> 	Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan <i>one to many</i> menghubungkan entitas A dan entitas B.

