

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

A. Konsep Dasar Sistem Informasi

1. Pengertian Sistem

Pengertian Sistem menurut Jogiyanto (2014:1) dalam bukunya yang berjudul *Analisis dan Desain Sistem Informasi*: “Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu”.

Menurut Mulyadi (2016:2) dalam bukunya yang berjudul *Sistem Akuntansi* menyimpulkan bahwa “Suatu sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan lainnya, yang berfungsi bersama-sama untuk mencapai tujuan tertentu”.

Berdasarkan beberapa pendapat yang dikemukakan diatas dapat disimpulkan bahwa sistem adalah kumpulan bagian atau sub sistem yang disatukan dan dirancang untuk suatu tujuan.

2. Karakteristik Sistem

Menurut Jogyanto (2014:4), dalam bukunya *Analisis dan Desain Sistem Informasi* suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu:

a. Komponen Sistem (*Components Sistem*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

b. Batas Sistem (*Boundary*)

Batas sistem (*Boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

c. Lingkungan Luar Sistem (*Environment*)

Lingkungan luar (*environment*) dari suatu sistem adalah apapun di luar batas dari sistem yang dapat mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedang lingkungan luar yang

merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup dari sistem.

d. Penghubung Sistem (*Interface*)

Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Keluaran (*output*) dari satu subsistem akan menjadi masukan (*input*) untuk subsistem yang lainnya dengan melalui penghubung. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

e. Masukan Sistem (*Input*)

Masukan (*input*) adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran.

f. Keluaran Sistem (*Output*)

Keluaran (*output*) adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supra sistem.

g. Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran. Suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan-bahan yang lain menjadi keluaran berupa barang jadi. Sistem akuntansi akan mengolah data-data transaksi menjadi laporan-laporan keuangan dan laporan-laporan lain yang dibutuhkan oleh manajemen.

h. Sasaran Sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

3. Klasifikasi Sistem

Menurut Jogiyanto (2014:6) dalam bukunya yang berjudul *Analisis dan Desain Sistem Informasi*, sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya adalah sebagai berikut:

a. Sistem Abstrak (*abstract system*) dan Sistem Fisik (*physical system*)

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Sistem merupakan sistem yang ada secara fisik.

- b. Sistem Alamiah (*natural system*) dan system Buatan Masnusia (*human made system*)

Sistem Alamiah adalah sistem yang terjadi melalui sistem alam, tidak dibuat oleh manusia. Sistem buatan manusia adalah sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi antara manusia dengan mesin disebut dengan *human-machine system* atau ada yang menyebutnya dengan *man-machine system*.

- c. Sistem Tertentu (*deterministic system*) dan Sistem Tak Tentu (*Probabilistic system*)

Sistem Tertentu yaitu sistem beroperasi dengan tingkah laku yang sudah dapat diprediksi. Sistem Tak Tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

- d. Sistem Tertutup (*closed system*) dan Sistem Terbuka (*open system*)

Sistem Tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah sistem *relatively closed system*. Sistem Terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya.

4. Pengertian Informasi

Secara Etimologi, Informasi berasal dari bahasa Perancis kuno *informacion* (tahun 1387) yang diambil dari bahasa Latin *informationem*

yang berarti “garis besar, konsep, ide”. Informasi juga dapat diartikan sebagai data yang telah diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

Telah diketahui bahwa informasi merupakan hal yang sangat penting bagi perusahaan dalam pengambilan keputusan sehari-hari. Beberapa ahli mendefinisikan informasi sebagai berikut:

Menurut Mulyanto (2009:12) dalam bukunya yang berjudul *Sistem Informasi Konsep dan Aplikasi*: “Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya, sedangkan data merupakan sumber informasi yang menggambarkan suatu kejadian yang nyata”.

Menurut Jogiyanto (2014:8) dalam bukunya yang berjudul *Analisis dan Desain Sistem Informasi* mengemukakan: “Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.”

5. Kualitas Informasi

Menurut Jogiyanto (2014:10), kualitas dari suatu informasi tergantung dari tiga hal yaitu:

a. Akurat

Informasi harus bebas dari kesalahan-kesalahan dan tidak bisa atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya.

b. Tepat pada waktunya

Informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan di dalam pengambilan keputusan.

c. Relevan

Informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda.

6. Sistem Informasi

Menurut Robert A. Leicth dan K. Roscoe Davis dalam Jogiyanto (2014:11) sistem informasi didefinisikan “Suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan”.

Dari definisi diatas dapat disimpulkan bahwa sistem informasi adalah suatu sistem yang memberikan informasi bagi pengguna untuk mengambil keputusan dalam mencapai suatu tujuan.

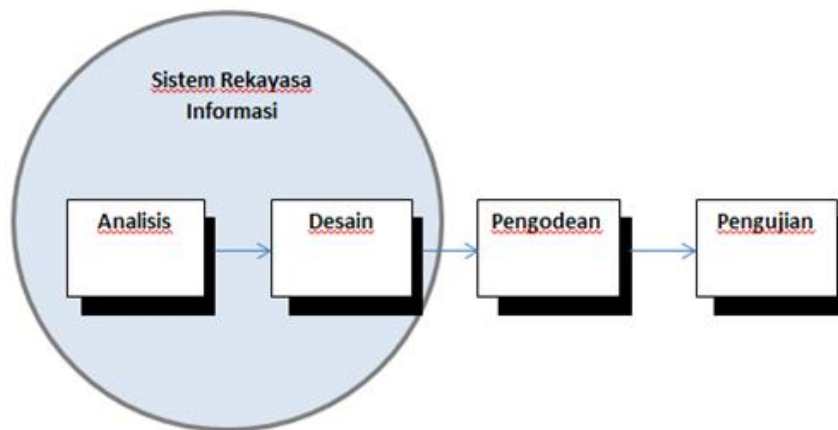
7. Sistem Informasi Manajemen

Menurut Kusrini dan Koniyo (2007:9) Sistem informasi manajemen adalah sebuah sistem yang mampu menyediakan informasi (merupakan hasil dari proses transaksi yang terjadi) dimana satu sama lain saling berinteraksi untuk mencapai tujuan yang telah ditetapkan oleh manajemen.

8. Teori Waterfall Model

Menurut Shalahuddin, M dan Rosa A. S (2015:28) mengemukakan bahwa Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linier*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian dan tahap pendukung (support).

Berikut adalah gambar model air terjun:



Sumber: Shalahuddin, M dan Rosa A. S (2015:29)

Gambar II.1. Pemodelan *Waterfall*

a. Analisis kebutuhan perangkat lunak

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 didokumentasikan.

b. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antar muka dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

c. Pembuatan kode program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

d. Pengujian

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 yang diinginkan.

B. Konsep Dasar Pemrograman

1. Pengertian Pemrograman

Menurut Kadir (2012:2) “Program adalah kumpulan instruksi yang digunakan untuk mengatur komputer agar melakukan suatu tindakan tertentu”. Tanpa program, komputer sesungguhnya tidak dapat berbuat

apa-apa. Itulah sebabnya, sering dikatakan bahwa komputer mencakup tiga aspek penting, berupa perangkat keras (*hardware*), perangkat lunak (*software*), yang dalam hal ini berupa program dan perangkat akal (*brainware*) atau orang yang berperan terhadap operasi komputer maupun pengembangan perangkat lunak. Dengan kata lain, program merupakan salah satu bagian penting pada komputer, yang mengatur komputer agar melakukan tindakan yang sesuai dengan yang dikehendaki oleh pembuatnya.

Suatu program ditulis dengan mengikuti kaidah bahasa pemrograman tertentu. Bahasa pemrograman dapat dianalogikan dengan bahasa yang digunakan manusia (bahasa manusia). Sebagaimana diketahui, ada bermacam-macam bahasa manusia, seperti bahasa inggris, bahasa indonesia, dan bahasa batak. Kumpulan instruksi dalam bahasa manusia yang berupa sejumlah kalimat dapat anda analogikan dengan suatu program. Manusia dapat mengerjakan suatu instruksi berdasarkan kalimat-kalimat dan komputer bisa menjalankan suatu instruksi menurut program.

2. Pengertian Website

Menurut Rachdian (2008:1) mengemukakan bahwa ”*website* adalah suatu pengenalan ruang informasi dimana sumber-sumber daya yang berguna diidentifikasi global yang disebut *Uniform Resource Identifier* (URI) atau lebih dikenal dengan istilah yang lebih populer yaitu *Uniform Resource Locator* (URL)”. Perkembangan *World Wide Web* (WWW)

yang sangat pesat ditandai dengan munculnya berbagai macam *website* dengan halaman *web* yang interaktif.

Berdasarkan isinya *website* terdiri dari dua jenis yaitu :

- a. *Website Statis (Static Website)* adalah *web* yang biasanya *user* tidak bisa mengubah *content* dari *web* tersebut secara langsung menggunakan *browser*. Interaksi yang terjadi hanya seputar pemrosesan *link* yang ada.
- b. *Webiste Dinamis (Dynamic Website)* adalah *web* yang biasanya *user* dapat mengubah *content* dari halaman tertentu dengan menggunakan *browser*.

3. Pengertian *HTML*

Menurut Ardhana (2012a:42) “*HTML* merupakan suatu bahasa yang dikenal oleh *web browser* untuk menampilkan informasi seperti teks, gambar, suara, animasi bahkan video”. Sekalipun banyak orang menyebutkan sebagai bahasa pemrograman. *HTML* sebenarnya sama sekali bukan bahasa pemrograman, karena dilihat dari namanya, *HTML* adalah suatu bahasa *Markup*, *HTML* digunakan untuk *Markup* (penandaan) terhadap sebuah dokumen teks, tanda tersebut digunakan untuk menentukan format atau *style* dari teks yang ditandai. Dokumen *HTML* merupakan teks murni (*ASCII*) dengan kode-kode special. Maksudnya adalah dokumen *HTML* hanya akan berisi tulisan berupa huruf, angka, tanda baca, dan karakter-karakter lainnya termasuk kode-kode *tag HTML* itu sendiri.

4. Pengertian Personal Hypertext Preprocessor (PHP)

Menurut Anhar (2010:3) “*PHP* adalah sebuah Bahasa *scripting* yang terpasang dalam *HTML*”. *PHP* dibuat pada tahun 1994 saat Rasmus Lerdorf seorang *programmer* Bahasa C membuat sejumlah *script perl* yang dapat menghitung jumlah dari pengunjung di dalam *web*-nya. Kemudian pada tahun 1995 ia membuat *tool* yang disebut Personal Home Page Tools versi 1.0 secara gratis yang isinya sekumpulan *script perl* yang dibuat agar *web* menjadi lebih dinamis, paket inilah yang menjadi cikal bakal *PHP*.

5. Pengertian MySQL

Menurut Wahana Komputer (2014:37) “*MySQL* adalah jenis *database server* yang sangat populer”. *MySQL* termasuk jenis RDBMS (*Relational Database Management System*), itulah sebabnya istilah seperti tabel, baris, dan kolom digunakan pada *MySQL*.

6. Pengertian CSS

Menurut Ardhana (2012b:108) “*Cassading Style Sheet* atau biasa disebut (*CSS*) merupakan salah satu bahasa pemrograman *web* untuk mengendalikan beberapa komponen dalam sebuah *web* sehingga akan lebih terstruktur dan seragam”. Sama hal nya *style* dalam aplikasi pengolahan kata seperti *Microsoft Word* yang dapat mengatur beberapa *style*, misalnya *heading*, *subbab*, *bodytext*, *footer*, *images*, dan *style* lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas

(*file*). Pada CSS dipakai untuk memformat tampilan dalam halaman *web* yang dibuat dengan bahasa *HTML* dan *XHTML*.

Nama CSS didapat dari fakta bahwa setiap deklarasi *style* yang berbeda dapat diletakan secara berurutan, yang kemudian membentuk hubungan ayah-anak (*parent-child*) pada setiap *style*. CSS sendiri merupakan sebuah teknologi *internet* yang direkomendasikan oleh *World Wide Web Consortium* atau W3C pada tahun 1996. Setelah CSS di standarisasikan, *Internet Explorer* dan *Netscape* melepas *browser* terbaru mereka yang telah sesuai atau paling tidak hampir mendekati dengan standar CSS.

7. Pengertian *Framework*

Menurut Budi Raharjo (2015:2), “*Web Application Framework* (WAF), atau sering disingkat *web framework* adalah suatu kumpulan kode berupa pustaka (*library*) dan alat (*tool*) yang dipadukan sedemikian rupa menjadi satu kerangka kerja (*framework*) guna memudahkan dan mempercepat proses pengembangan aplikasi web”.

Proses pengembangan *web* itu sendiri dapat dilakukan dengan beragam bahasa pemrograman; bisa *PHP*, *Python*, *Ruby*, *Java* dan sebagainya. Saat ini, banyak bermunculan *framework web* yang dirancang untuk bahasa-bahasa pemrograman tersebut. Di antara bahasa pemrograman yang lain, *framework* untuk PHP memiliki varian paling banyak. Berikut ini daftar dari beberapa *framework web* yang dapat digunakan.

Framework web untuk PHP:

- *CodeIgniter*
- *YII*
- *Zend Framework*
- *Laravel*
- *Symfony*
- dll

Framework web untuk Python:

- *Django*
- *CherryPy*
- *Flask*
- *Pyramid*
- dll

Framework web untuk Ruby:

- *Ruby on Rails*
- *Sinatra*
- *Ramaze*
- *Padrino*
- Dll

Sebagian besar dari *framework* yang tertera di atas mengimplementasikan pola desain atau aritektur *Model-View-Controller* (MVC), yang memisahkan bagian kode untuk penanganan proses bisnis dengan bagian kode untuk keperluan presentasi (tampilan).

8. Pengertian *CodeIgniter* 3.0

Menurut Budi Raharjo (2015:3), “*CodeIgniter* adalah *framework web* untuk bahasa pemrograman *PHP* yang dibuat oleh Rick Ellis pada tahun 2006, penemu dan pendiri EllisLab. EllisLab adalah suatu tim kerja yang berdiri pada tahun 2002 dan bergerak di bidang pembuatan *software* dan *tool* untuk para pengembang *web*”.

CodeIgniter memiliki banyak fitur (fasilitas) yang membantu para pengembang (*developer*) *PHP* untuk dapat membuat aplikasi *web* secara mudah dan cepat. Dibandingkan dengan *framework web PHP* lainnya, harus diakui bahwa *CodeIgniter* memiliki desain yang lebih sederhana dan bersifat fleksibel (tidak kaku). *CodeIgniter* mengizinkan para pengembang untuk menggunakan *framework* secara parsial atau secara keseluruhan.

CodeIgniter merupakan sebuah *toolkit* yang ditujukan untuk orang yang ingin membangun aplikasi *web* dalam bahasa pemrograman *PHP*. Beberapa keunggulan yang ditawarkan oleh *CodeIgniter* adalah sebagai berikut:

- *CodeIgniter* adalah *framework* yang bersifat *free* dan *open-source*.
- *CodeIgniter* memiliki ukuran yang kecil dibandingkan dengan *framework* lain. Setelah proses instalasi, *framework CodeIgniter* hanya berukuran kurang lebih 2MB (tanpa dokumentasi atau jika direktori *user_guide* dihapus). Dokumentasi *CodeIgniter* memiliki ukuran sekitar 6MB.
- Aplikasi yang dibuat menggunakan *CodeIgniter* bisa berjalan cepat.

- *CodeIgniter* menggunakan pola desain *Model-View-Controller* (MVC) sehingga satu *file* tidak terlalu berisi banyak kode. Hal ini menjadikan kode lebih mudah dibaca, dipahami, dan dipelihara di kemudian hari.
- *CodeIgniter* dapat diperluas sesuai dengan kebutuhan.
- *CodeIgniter* terdokumentasi dengan baik. Informasi tentang pustaka kelas dan fungsi yang disediakan oleh *CodeIgniter* dapat diperoleh melalui dokumentasi yang disertakan di dalam paket distribusinya.

C. Peralatan Pendukung (*Tools System*)

1. *Unified Modeling Language* (UML)

UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, *Object Modelling Technique* (OMT) dan *Object Oriented Software Engineering* (OOSE). Metode Booch terkenal dengan nama metode *Object Oriented Design*. Abstraksi konsep dasar UML terdiri dari *structural classification*, *dynamic behaviour* dan *model management*.

Menurut Widodo (2011:10), “Beberapa *literature* menyebutkan bahwa UML menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung, misanya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi”. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain:

a. *Class Diagram*

Bersifat statis, Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif.

b. *Package Diagram*

Bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas, merupakan bagian dari diagram komponen.

c. *Sequence Diagram*

Bersifat dinamis. Diagram urutan adalah iterasi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.

d. *Communication Diagram*

Bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi UML yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan.

e. *Statechart Diagram*

Bersifat dinamis. Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (state), transisi, kejadian serta aktivitas.

f. *Activity Diagram*

Bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting

dalam pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek.

g. *Use Case Diagram*

Bersifat statis. Diagram ini memperlihatkan himpunan use-case dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.

h. *Component Diagram*

Bersifat statis. Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya.

i. *Deployment Diagram*

Bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (run-time). Memuat simpul-simpul beserta komponen-komponen yang di dalamnya. Kesembilan diagram ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semuanya dibuat sesuai kebutuhan. Pada UML dimungkinkan kita menggunakan diagram-diagram lainnya misalnya data flow diagram, entity relationship diagram, dan sebagainya.

2. *Entity Relationship Diagram (ERD)*

Menurut Shalahuddin, M dan Rosa A. S (2015:50) . ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter

Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi *Crow's Foot*, dan beberapa notasi lain. Namun yang banyak digunakan adalah dari notasi Chen. Berikut adalah simbol-simbol yang digunakan pada ERD dengan notasi Chen :

a. Entitas / *Entity*

Entitas merupakan data inti yang akan disimpan untuk 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.

b. Atribut

Field atau kolom data yang butuh disimpan dalam suatu entitas.

c. Atribut Kunci Primer

Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses *record* yang diinginkan, biasanya berupa id, kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut bersifat unik (berbeda tanpa ada yang sama).

d. Relasi

Relasi yang menghubungkan antar entitas biasanya diawali dengan kata kerja.

2.2. Penelitian Terkait

Menurut S. Rintjap, Sherwin, Sompie dan Lantang (2014:1)

menyimpulkan bahwa:

“Perkembangan ilmu pengetahuan di segala bidang dalam era globalisasi saat ini begitu pesat. Terutama dalam bidang IT yang semakin maju seiring dengan kebutuhan pemakai (*user*). Absensi merupakan salah satu penunjang yang dapat mendukung atau memotivasi setiap kegiatan yang dilakukan di dalam dunia pendidikan di Indonesia pada saat ini, terutama dalam setiap proses kegiatan belajar. Penggunaan sidik jari untuk absensi siswa bisa menjadi solusi cara absensi yang lebih baik. Yang diharapkan dapat mengatasi permasalahan dan meningkatkan kinerja di Sekolah. Tugas akhir dengan judul Aplikasi Absensi Siswa Menggunakan Sidik Jari di SMA N 9 Manado ini bertujuan untuk memperbaiki system absensi yang dilakukan dengan cara manual. Pengembangan aplikasi dilakukan dengan menggunakan Bahasa pemrograman VB. Net dan menggunakan mesin *fingerprint* sebagai alat untuk merekan sidik jari. Dari hasil implementasi aplikasi ini dapat disimpulkan bahwa aplikasi ini memberikan manfaat menghindari terjadinya manipulasi data, menghemat waktu dalam melakukan absensi.”

Menurut Rinawati dan Candrawati (2013:1) menyimpulkan bahwa:

“Perkembangan Teknologi Informasi dari tahun ketahun yang semakin cepat menjadi tantangan berat bagi pengguna teknologi informasi itu sendiri dan mendorong setiap sector organisasi baik formal maupun informal atau lembaga-lembaga lainnya untuk dapat memanfaatkannya sebagai penunjang kegiatan kerja sehingga dapat menghasilkan informasi yang cepat, tepat dan akurat. PT Harja Gunatama Lestari Bandung sebagai perusahaan dagang yang aktivitas utamanya adalah pembelian dan penjualan yang tentu saja membutuhkan manajemen yang baik dalam mengelola karyawannya yang nantinya akan berdampak positif bagi perkembangan perusahaan itu sendiri, maka dibutuhkan informasi yang efisien dalam mempertahankan keberadanaannya agar tetap eksis dan dikenal oleh masyarakat luas. Salah satu upaya perusahaan untuk mendukung hal itu adalah dengan pemakaian system computer yang baik. Untuk mendapat system yang baik maka dilakukan penelitian terhadap system yang sudah ada. Tahap berikutnya adalah melakukan analisis untuk mengetahui kelemahan dan kekurangan system tersebut, kemudian dilakukan perancangan system yaitu *Flow Map*, *Data Flow Diagram*, *Data Dictionary*, keterangan proses dan tahap terakhir yaitu implementasi yang mengubah perancangan kedalam tahap Bahasa pemograman.”