BAB IILANDASAN TEORI

2.1. Konsep Dasar

Suatu sistem dapat terdiri dari sistem-sistem bagian (subsistem). Subsistem  
yang saling berinteraksi dan saling berhubungan membentuk suatu kesatuan sehingga  
tujuan atau sasaran sistem tersebut dapat tercapai. Interaksi dari subsistem-subsistem  
terjalin sedemikian rupa, sehingga dicapai suatu kesatuan yang terpadu atau  
terinteg.

**2.1.1. Pengertian Sistem** Menurut Atmosudirdjo dalam (Sutabri,2016) menyatakan bahwa:  
“suatu sistem terdiri atas objek-objek, atau unsur-unsur, atau komponen-komponen  
yang berkaitan atau berhubungan satu sama lainya sedemikian rupa sehingga unsur-unsur tersebut merupakan suatu kesatuan pemrosesan atau pengolahan yang  
tertentu”.  
 Menurut (Sutabri, 2016) menyatakan bahwa :  
Pendekatan yang menekankan pada prosedur mendefinisikan sistem sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Sedangkan pendekatan sistem yang lebih menekankan pada elemen atau komponen mendefinisikan system sebagai kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.

**2.1.2. Karakteristik Sistem** Menurut (Mustakini, 2017) model umum sebuah sistem terdiri dari *input*,  
proses dan *output*. Hal ini merupakan konsep sebuah sistem yang sangat sederhana  
mengingat sebuah sistem dapat mempunyai beberapa masukan dan pengeluaran.  
Adapun karakteristik yang di maksud adalah sebagai berikut:

1. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk suatu kesatuan komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat-sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih basar, yang disebut dengan Supra sistem.

2. Batasan Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara system dengan sistem lainya atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisahkan-pisahkan.

3. Lingkungan Luar Sistem (*Environment*)

Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem. Lingkungan luar sistem ini dapat menguntungkan dan dapat juga merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi bagi sistem tersebut, yang dengan demikian lingkungan luar tersebut harus selalu dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus  
dikendalikan, karena kalau tidak maka akan mengganggu kelangsungan hidup sistem tersebut.

4. Penghubung Sistem (*Interface*)

Sebagai media yang menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem atau *interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lain. Keluaran suatu subistem akan menjadi masukan untuk subsistem yang lain dengan melewati penghubung. Dengan demikian terjadi suatu integrasi yang membentuk satu kesatuan.

5. Masukan Sistem (*Input*)

Energi yang di masukan ke dalam sistem disebut masukan sistem. Yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). Sebagai contoh, di dalam suatu unit sistem komputer,”program” adalah *maintenance input* yang digunakan untuk mengoprasikan komputer sementara “data” adalah *signal input* yang akan diolah menjadi informasi.

6. Keluaran Sistem (*Output*)

Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang  
berguna, keluaran ini merupakan masukan bagi subsistem yang lain. Seperti contoh sistem informasi, keluaran yang dihasilkan adalah informasi, yang mana informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang merupakan input bagi subsistem lainya.

7. Pengolah Sistem (*Process*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran. Sebagai contoh, sistem akutansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

8. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat *deterministic*. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

**2.1.3. Klasifikasi Sistem** Menurut (Sutabri, 2016) Sistem merupakan bentuk integrasi antara satu  
komponen dengan 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 sebagai berikut:

1. Sistem abstrak dan sistem fisik

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak nampak secara fisik, misalnya sistem theologi. Yaitu suatu sistem yang berupa pemikiran tentang hubungan antara manusia dengan Tuhan, sedangkan system fisik merupakan sistem yang ada secara fisik, seperti sistem komputer, system produksi, sistem penjualan, sistem administrasi personalia, dan lain sebagainya.

2. Sistem alamiah dan sistem buatan manusia

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem perputaran bumi, terjadinya siang malam, pergantian musim. Sedangkan sistem buatan manusia merupakan sistem yang melibatkan hubungan manusia dengan mesin, yang disebut dengan *human machine system*. Sistem informasi berbasis komputer merupakan contohnya, karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

3. Sistem deterministik dan sistem probabilistic

Sistem yang beroperasi dengan tingkah laku yang dapat diprediksi disebut sistem deterministik. Sistem komputer adalah contoh dari sistem yang tingkah lakunya dapat dipastikan berdasarkan program-program komputer yang di jalankan. Sedangkan sistem yang bersifat probabilistik adalah sistem yang kondisi masa depannya tidak dapat diprediksi, karena mengandung unsur probabilitas.

4. Sistem terbuka dan sistem tertutup

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa ada campur tangan dari pihak luar. Sedangkan sistem terbuka adalah system yang berhubungan dan dipengaruhi oleh lingkungan luarnya, yang menerima masukan dan menghasilkan keluaran untuk subsistem lainnya.

**2.1.4. Pengertian Internet**

Menurut Rossa dan Shalahudin dalam (Supriyatna, 2015), *Internet* atau *internetwork* adalah sekumpulan jaringan berbeda yang saling berhubungan bersama sebagai salah satu kesatuan dengan menggunakan berbagai macam protocol, salah satunya adalah protocol TCP/IP (*Transmission Control Protocol*/*Internet Protocol*).

**2.1.5. Pengertian *Website***

Menurut raharjo dalam (Supriyanta & Nisa, 2015), "*World Wide Web* (WWW), sering disingkat dengan web, adalah suatu layanan di dalam jaringan yang berupa ruang informasi”. Sedangkan menurut Sibero dalam (Supriyanta & Nisa, 2015), “*World Wide Web* (W3) atau yang dikenal juga dengan istilah web adalah suatu sistem yang berkaitan dengan dokumen digunakan sebagai media untuk menampilkan teks, gambar, multimedia dan lainnya pada jaringan internet”.

Selanjutnya mari kita bahas pengertian dan perbedaan *web* statis dan *web*dinamis menurut (Ferdianto, 2013), yaitu sebagai berikut:

1. *Web* Statis

*Web* statis adalah *website* yang mana pengguna tidak bisa mengubah konten dari *web* tersebut secara langsung menggunakan *browser*. Interaksi yang terjadi antara pengguna dan *server* hanyalah seputar pemrosesan link saja. Halamanhalaman *web* tersebut tidak memliki *database*, data dan informasi yang ada pada *web* statis tidak berubah-ubah kecuali diubah sintaksnya. Dokumen *web* yang dikirim kepada *client* akan sama isinya dengan apa yang ada di *webserver*. Contoh dari *web* statis adalah *web* yang berisi profil perusahaan. Di sana hanya ada beberapa halaman saja dan kontennya hampir tidak pernah berubah karena konten langsung diletakan dalam file HTML saja.

2. *Web* Dinamis

Dalam *web dinamis*, interaksi yang terjadi antara pengguna dan *server* sangat kompleks. Seseorang bisa mengubah konten dari halaman tertentu dengan menggunakan *browser*. *Request* (permintaan) dari pengguna dapat diproses oleh *server* yang kemudian ditampilkan dalam isi yang berbeda-beda menurut alur programnya. Halaman-halaman *web* tersebut memiliki database. *Web* dinamis, memiliki data dan informasi yang berbeda-beda tergantung *input*apa yang disampaikan *client*. Dokumen yang sampai di *client* akan berbeda dengan  
dokumen yang ada di *webserver*. Contoh dari *web* dinamis adalah portal berita dan jejaring sosial. Lihat saja *web* tersebut, isinya sering diperbaharui (di*update*) oleh pemilik atau penggunanya. Bahkan untuk jejaring sosial sangat sering di-*update* setiap harinya.

**2.1.6. Bahasa Program *(script program*)**

Bahasa program merupakan bahasa yang digunakan untuk menterjemahkan setiap perintah pada saat *website* tersebut sedang dijalankan. Bahasa program yang penulis gunakan adalah *HTML dan PHP*.

1. HTML

Menurut Raharjo dalam (Supriyanta & Nisa, 2015), HTML adalah singkatan dari *HyperText Markup Language*, yaitu bahasa (aturan) standar yang digunakan untuk menampilkan teks, gambar, video atau audio ke dalam halaman web. Bahasa HTML merupakan bahasa yang digunakan untuk membuat halaman *website* dengan menggunakan tag-tag yang telah dideklarasikan pada halaman *notepad* dan dapat saling berhubungan dengan dokumen HTML lainnya atau yang sering disebut dengan istilah *link*.

1. PHP

Menurut (Winarno& Zaki, 2013) “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”.

1. CSS

Menurut (Hidayatullah& Kawistara, 2014) ”dengan CSS kita bisa menyimpan format dan menggunakannya kapanpun dan dimanapun kita inginkan.” Dengan CSS kita bisa membuat efek-efek tertentu untuk konten *web* yang kita buat. Contohnya kita ingin bagian *header* pada *web* selalu diberi *fontVerdana*, *size* 20 dan warna biru. Maka kita definisikan *style* tersebut untuk tag <TH>. CSS (*Cascading Style Sheet*) sudah didukung oleh hampir semua *web browser* membuat karena CSS telah distandarkan oleh *World Wide Web Consortium (W3C).* Jadi pilihan yang tepat untuk kita format halaman *web* kita agar terlihat lebih indah dan cantik dimanapun *user* membuka *web* kita.

**2.1.7. Sistem Informasi Akuntansi**

Menurut (Marshall & Paul, 2016) “Sistem Informasi Akuntansi (SIA) adalah sebuah sistem yang mengumpulkan, mencatat, menyimpan, dan mengolah data untuk menghasilkan informasi bagi para pembuat keputusan “

**2.1.8 Penjualan Tunai & Kredit**

Menurut (Sholikhah, Sairan, & Syamsiah, 2017) mengatakan bahwa: Penjualan secara umum ada 2, yaitu penjualan tunai dan penjualan kredit. Penjualan yang dilakukan dengan tidak ada sistem hutang disebut dengan penjualan tunai. Sedangankan penjualan yang dilakukan dengan system hutang disebut dengan penjualan kredit. Penjualan merupakan pendapatan yang diperoleh dari menjual barang yang mana jumlah yang dibebankan kepada pembeli untuk barang dagang yang diserahkan merupakan pendapatan perusahaan yang bersangkutan.

**2.2. Peralatan Pendukung**

**2.2.1. *Unified Modelling Language***

Menurut (Sukamto & Shalahuddin, 2016) “UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefisinikan requirement, membuat analisi dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek” UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari system perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dari teks-teks pendukung. Pada UML 2,3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori yaitu *structure diagrams* yaitu yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. Terdiri dari *class diagram*, *object diagram*, *component diagram*, *composite diagram*, *package diagram*, *deployment diagram*. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang digunakan untuk  
menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem. Terdiri dari, *use case diagram*, *activity diagram*, *state machine diagram. Interaction diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sustem lain maupun interaksi antar subsistem pada suatu sistem. Terdiri dari *sequence diagram*, *communication diagram*, *timing diagram*, *interaction overview diagram* (Sukamto & Shalahuddin, 2016). Berikut penjelasan mengenai diagram-diagram UML menurut (Sukamto &  
Shalahuddin, 2016):

1. *Use Case Diagram*

*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 mengetahi fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

1. *Activity Diagram*

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktiitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

1. *Deployment Diagram*

Diagram *deployment* atau *deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat

digunakan untuk memodelkan hal-hal berikut, yaitu sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, *node*, dan *hardware*, sistem *client/server*, sistem terdistribusi murni dan rekayasa ulang aplikasi.

1. *Component Diagram*

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.

**2.2.2. *Entity Relationship Diagram***

*Entity Relationship Diagram (ERD) digunakan dalam perancangan*

*database.* Menurut Rosa dan Shalahuddin dalam (Mulyadi & Sinnun, 2018) *Entity Relation Diagram* (ERD) adalah bentuk awal dalam melakukan perancangan basis data relasional. ERD merupakan suatu model yang menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. Pada dasarnya ada tiga simbol yang digunakan sebagai berikut:

1. Entitas

Digambarkan dengan kotak persegi panjang dan digunakan untuk menunjukkan sekumpulan orang,tempat, objek atau konsep dan sebagainya yang menunjukkan dimana data dicatat atau disimpan.

1. Hubungan atau Relasi

Digambarkan dengan kotak berbentuk *diamond* atau belah ketupat dengan garis yang menghubungkan ke entitas yang terkait. Maka *relationship* diberi nama dengan kata kerja. Hubungan atau relasi menunjukkan abstraksi dari sekumpulan hubungan yang mengaitkan antara entitas yang berbeda.

1. Atribut

Digambarkan dengan bentuk elips. Atribut menunjukkan karakteristik dari tiap entitas atau sesuatu yang menjelaskan entitas atau hubungan. Sehingga atribut dikatakan elemen dari entitas dan relasi. Dari setiap atribut entitas terdapat satu atribut yang dijadikan sebagai kunci (*key*). Beberapa jenis kunci tersebut antara lain :*Primary key, Candidate key, Composite key, Secondary key, Alternate key* dan *Foreign key*.

1. Tingkat Hubungan (*Cardinality*)

*Entity Relation Diagram* (ERD) juga menunjukkan tingkat hubungan yang

terjadi, dilihat dari segi kejadian atau banyak tidaknya hubungan antara entitas tersebut.

**2.2.3. *Logical Record Structure***

Menurut (Hasugian, 2012 ) menjelaskan bahwa :

LRS *(Logical Record Structure)* merupakan hasil proses transformasi dari ERD dimana *primary key* yang terdapat pada masing-masing relasi akan masuk pada entitas yang lebih kuat. Proses transformasi digambarkan dengan persegi empat dengan garis putus-putus. Pada LRS akan kelihatan setiap *primary key* yang akan menjadi *foreign key* pada entitas lain. LRS ini menjadi patokan untuk pembuatan *Conceptual Data Model.*

**2.2.4. Pengujian *Web***

Pengujian *web* dapat dilakukan dengan dua macam metode pengujian  
perangkat lunak yaitu *blackbox* atau *blackbox testing*. *Black-box testing* yaitu perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah (Sukamto dan Shalahuddin, 2016 ), misalkan untuk kasus proses *login* maka kasus uji coba yang dilakukan adalah:

1. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar.

*2.* Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalkan nama pemakai benar tapi kata sandi salah, atau sebaliknya atau keduanya salah.

**2.2.5. *Black-Box Testing***

Menurut (Sukamto dan Shalahuddin, 2016) mengatakan bahwa:

*“Black-Box testing* (pengujian kotak hitam) yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi,masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan”. Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus *login* maka kasus uji yang dibuat adalah :

1. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*)  
yang benar.

2. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*)  
yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau  
sebaliknya, atau keduanya salah.