

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Menurt (Sri, 2016:2) mengatakan sistem adalah: “Kumpulan dari dua atau lebih komponen yang saling bekerja dan berhubungan untuk mencapai tujuan tertentu.”Sedangkan (Pratama, 2014:7) dalam bukunya menyatakan “Sistem didefinisikan sebagai kumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan suatu tugas bersama-sama”. Dalam mendefinisikan sebuah sistem, terdapat dua pendekatan yaitu pendekatan yang menekankan pada prosedur dan pendekatan yang menekankan pada elemen atau komponennya. Sebuah sistem terdiri atas bagian-bagian atau komponen-komponen yang terpadu untuk satu tujuan. Dalam landasan teori ini akan dijelaskan mengenai konsep dasar teori yang berkenaan dengan pengembangan sistem.

A. Model Pembelajaran Berbasis Web

Menurut (Firmansyah & Saidah, 2016) dalam jurnalnya “proses belajar mengajar pada dasarnya adalah proses komunikasi yang melibatkan penyampaian pesan materi dari pengantar ke penerima. Dalam proses penyampaian pesan tersebut, diperlukan suatu media agar pesan dapat diterima dengan baik. Media merupakan salah satu komponen yang sangat penting dalam suatu proses komunikasi. Tingkat efektifitas media yang digunakan sangat berpengaruh pada sejauh mana suatu peran komunikasi akan dapat diterima oleh audience dengan cepat dan tepat atau sebaliknya. Perkembangan teknologi informasi dan komunikasi (TIK) yang sangat pesat telah berpengaruh terhadap berbagai aspek kehidupan manusia, termasuk dalam interaksi

antara guru dan peserta didik. Interaksi tersebut memerlukan dukungan media instruksional yang inovatif, kreatif, tepat, dan efektif ”.

B. Sistem

1. Definisi Sistem

Dalam kehidupan sehari-hari di era teknologi dan komputrisasi saat ini, kita sering mendengar istilah sistem, misalkan sistem komputer, sistem operasi, sistem informasi, sistem geografis, sistem akademis, dan lainnya. Bahkan disadari maupun tidak, kita sering menggunakan atau memanfaatkan layanan yang diberikan oleh sistem tersebut. Kemudian muncul sebuah pertanyaan, apakah yang dimaksud dengan sistem ?

Sistem didefinisikan sebagai sekumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan suatu tugas bersama-sama. Secara garis besar, sebuah sistem informasi terdiri atas tiga komponen utama. Ketiga komponen tersebut mencakup *software*, *hardware*, dan *brainware*. Ketiga komponen ini saling berkaitan satu sama lain. *Software* mencakup semua perangkat lunak yang dibangun dengan bahasa, pemrograman tertentu, pustaka, untuk kemudian menjadi sistem operasi, aplikasi, dan *driver*. Sistem operasi, aplikasi, *driver*, saling bekerja sama agar komputer dapat berjalan dengan baik. *Hardware* mencakup semua perangkat keras (*motherboard*, *processor*, VGA, dan lainnya) yang disatukan menjadi sebuah komputer. Dalam konteks yang luas, bukan hanya sebuah komputer, namun sebuah jaringan komputer. *Brainware* mencakup kemampuan otak manusia, yang mencakup ide, pemikiran, analisis, di dalam menciptakan dan menggabungkan *hardware* dan *software*. Penggabungan *software* dan *hardware* dengan bantuan *brainware* inilah (melalui sejumlah prosedur) yang dapat menciptakan sebuah sistem yang bermanfaat bagi pengguna. (Pratama, 2014:7)

Menurut (Jeperson Hutahean, 2014:3) Supaya sistem itu dikatakan sistem yang baik memiliki karakteristik, sebagai berikut :

1. Komponen Sistem (*Component System*)

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

2. Batasan Sistem (*Boundary*)

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

3. Lingkungan Luar Sistem (*Environment*)

Lingkungan luar sistem (*Environment*) adalah diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan dapat bersifat menguntungkan yang harus tetap dijaga dan yang merugikan yang harus di jaga dan dikendalikan, kalau tidak akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung sistem (*Interface*)

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari subsistem ke subsistem lain. Keluaran (*output*) dari subsistem akan menjadi masukan (*input*) untuk subsistem lain melalui penghubung.

5. Masukan Sistem (*Input*)

Masukkan adalah energi yang dimasukkan kedalam sistem, yang dapat berupa perawatan (*maintance input*) dan masukan sinyal (*signal input*). *Maintance input* adalah energi yang dimasukkan agar sistem dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran. Contoh dalam sistem *computer* program adalah *maintance input* sedangkan data adalah *signal input* untuk diolah menjadi informasi.

6. Keluaran Sistem (*Output*)

Keluaran Sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Contoh komputer menghasilkan panas yang merupakan sisa pembuangan, sedangkan informasi adalah keluaran yang dibutuhkan.

7. Pengolah Sistem

Suatu sistem menjadi bagian pengolah yang akan merubah masukan menjadi keluaran. Sistem produksi akan mengolah bahan baku menjadi bahan jadi, system akuntansi akan mengolah data menjadi laporan-laporan keuangan.

8. Sasaran Sistem

Sasaran sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Sasaran dari sistem sangat menentukan input yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

2. Sistem Berorientasi Objek (OOP)

Pemrograman berorientasi objek atau *objek-oriented programming* (OOP) merupakan suatu pendekatan pemrograman menggunakan objek dan class. Saat ini,

konsep OOP sudah semakin berkembang, Hampir setiap perguruan tinggi di dunai mengajarkan konsep OOP pada Mahasiswa/i-nya. Pemrograman yang banyak dipakai dalam penerapan OOP adalah C++, Java, Python, Net, Ruby dan sebagainya. Tujuan dari OOP adalah untuk memudahkan programmer dalam pembuatan program dengan menggunakan konsep objek yang ada dalam kehidupan sehari – hari. Jadi, setiap bagian permasalahan adalah objek, dan objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil.(Subagia, 2018:4). Sistem berorientasi objek juga memiliki sifat-sifat tertentu, antara lain :

a) Kelas

kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh '*class of dog*' adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah *class* adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi object. Sebuah *class* secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalam sebuah *class* sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak menggunakan OOP). Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.

b) Objek

membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.

c) Abstraksi

Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.

d) Enkapsulasi

Memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak hanya metode dalam objek tersebut yang diberi izin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

e) Polimorfisme

melalui pengiriman pesan. Tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesan tersebut dikirim. Contohnya, bila sebuah burung menerima pesan "gerak cepat", dia akan menggerakkan sayapnya dan terbang. Bila seekor singa menerima pesan yang sama, dia akan menggerakkan kakinya dan berlari. Keduanya menjawab sebuah pesan yang sama, namun yang sesuai dengan kemampuan hewan tersebut. Ini disebut polimorfisme karena sebuah *variabel* tunggal dalam program dapat memegang berbagai jenis objek yang berbeda selagi program berjalan, dan

teks program yang sama dapat memanggil beberapa metode yang berbeda di saat yang berbeda dalam pemanggilan yang sama. Hal ini berlawanan dengan bahasa fungsional yang mencapai polimorfisme melalui penggunaan fungsi kelas pertama. mendapatkan data tersebut melalui objek petugas administrasi. Jadi untuk menyelesaikan suatu masalah dengan kolaborasi antar objek-objek yang ada karena setiap objek memiliki deskripsi tugasnya sendiri.

C. Website

1. Website

Menurut (Agus & Safitri, 2015) dalam jurnalnya pengertian website adalah “keseluruhan halaman-halaman *web* yang terdapat dari sebuah domain yang mengandung informasi”.

a) Internet

“Internet adalah jaringan komputer. Ibarat jalan raya, *internet* dapat dilalui berbagai sarana transportasi, seperti bus, mobil dan motor yang memiliki kegunaan masing – masing”.

b) Web Browser

“*Web browser* adalah alat yang digunakan untuk melihat halaman dengan *web browser* kita dapat memperoleh informasi yang di sediakan oleh server *web*”.

c) Web Server

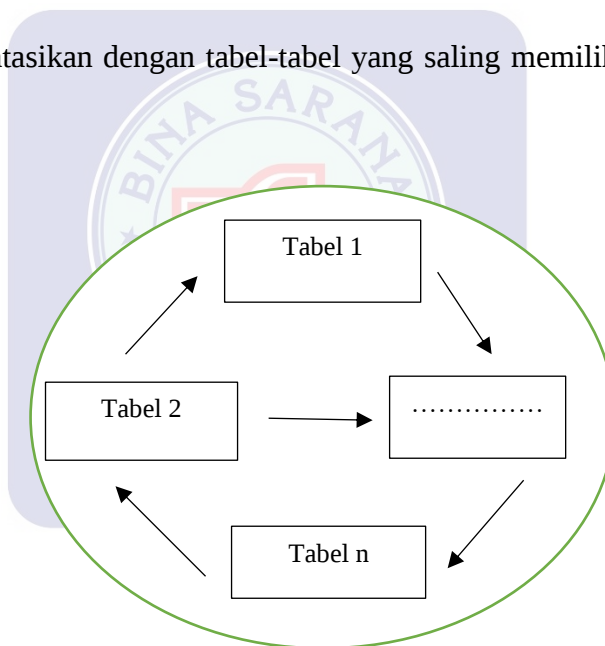
Menerangkan bahwa pengertian *web server* adalah “*Server Web* merujuk pada perangkat keras (server) dan perangkat lunak yang menyediakan layanan akses kepada pengguna melalui protokol komunikasi HTTP ataupun variannya (seperti FTP dan HTTPS) atas berkas-berkas yang terdapat pada suatu URL ke pemakai”

d) *Web Editor*

Web Editor adalah “sebuah aplikasi yang sering digunakan *Developer Web* untuk mengetikkan script atau perintah-perintah dokumen web, baik itu *Client Side Script* (CSS) atau pun *Server Side Scripting* (SSS)”.

D. Basis Data

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan memuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Pada buku ini menggunakan basis data relasional yang diimplementasikan dengan tabel-tabel yang saling memiliki relasi seperti pada gambar berikut.



Sumber : (Rosa A.S., 2018)

Gambar II.1 Ilustrasi Basis Data

Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya, entah berupa *file* teks ataupun *Database Management System* (DBMS). (Rosa A.S, 2018:43)

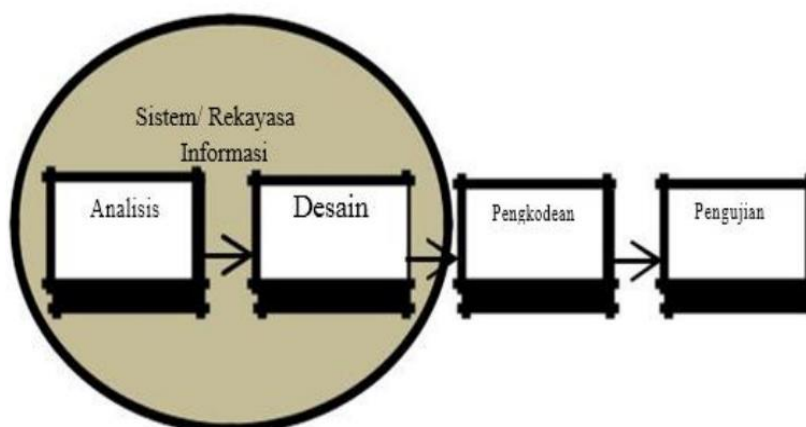
Menurut (Fridayanthie & Charter, 2016) kebutuhan basis data dalam sistem informasi meliputi.

1. Memasukan, meyimpan, dan mengambil data .
2. Membuat laporan berdasarkan data yang telah disimpan .

Tujuan dari dibuatnya table-tabel di sini adalah untuk meyimpan data ke dalam table-tabel agar mudah diakses. Oleh karena itu, untuk merancang table-tabel yang akan dibuat maka dibutuhkan pola piker peyimpanan data nantinya jika dalam bentuk baris-baris data (*record*) dimana setiap baris terdiri dari beberapa kolom.

E. Model Pengembangan Perangkat Lunak

Menurut (Suryanto, 2016) dalam jurnalnya “Model waterfall adalah model SDLC (*Software Development Life Cycle*) yang paling sederhana. Model ini hanya cocok untuk pengembangan perangkat lunak dengan spesifikasi yang tidak berubah-ubah”. Model waterfall ini sering juga disebut model sekuensial linier yang menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian dan tahap pendukung (support).



Sumber: (Suryanto, 2016).

Gambar II.2 Ilustrasi Basis Data

Adapun fungsi dari tiap tiap bagian model *waterfall* menurut (Suryanto, 2016) dalam jurnalnya adalah sebagai berikut :

1. Analisa kebutuhan perangkat lunak

Analisis sistem dilakukan untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikan.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, 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.

3. 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..

2.2. Teori Pendukung

Peralatan pendukung (*Tools system*) merupakan alat untuk menggambarkan bentuk *logical* model dan suatu sistem dengan menggunakan simbol-simbol, lambang-lambang, atau diagram yang menunjukkan secara tepat arti dan fungsinya. Fungsi dari sistem informasi dapat bekerja dengan baik dan dapat lebih dimengerti lagi peralatan apa saja yang digunakan, adapun pendukung (*Tools system*) tersebut diantara adalah sebagai berikut.

A. *Entity Relationship Diagram*

Entity Relation Diagram digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan ODBMS maka perancangan basis data tidak perlu menggunakan ERD. 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. (Rosa A.S., 2018:50). Menurut (Rosa A.S., 2018:50) komponen yang terdapat pada *Entity Relationship Diagram* yaitu :

1. **Komponen ERD**

a) Entity

Entiti merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain Simbol dari entiti ini biasanya digambarkan dengan persegi panjang.

b) Atribut

Setiap entitas pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi isi elemen satu dengan yang lain. Gambar atribut diwakili oleh simbol elips.

c) Hubungan / Relasi

Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Relasi dapat digambarkan sebagai berikut : Relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B).

2. LRS

Menurut (Kuryanti & Kom, 2016) dalam jurnalnya “Logical Record Structure (LRS) dibentuk dengan nomor dari tipe record. Beberapa tipe record digambarkan oleh kotak persegi panjang dan dengan nama yang unik. Perbedaan LRS dengan E-R diagram adalah nama tipe record berada diluar kotak field tipe record ditempatkan. LRS terdiri dari link-link diantara tipe record. Link ini menunjukkan arah dari satu tipe record field-field yang kelihatan pada kedua link tipe record. Penggambaran LRS mulai dengan menggunakan model yang dimengerti. Dua metode yang dapat digunakan, dimulai dengan hubungan kedua model yang dapat dikonversikan ke LRS, metode yang lain dimulai dengan ER- Diagram dan langsung dikonversikan ke LRS”. Adapun kardinalitas tersebut :

a) 1:1 (one to one)

Relasi yang terjadi antara suatu entity dengan entity lainnya yang memiliki hubungan 1:1.

b) 1:M (one to many)

Relasi yang terjadi antara suatu entity dengan entity lainnya yang memiliki hubungan 1:M.

c) M:N (many to many)

Relasi yang terjadi antara suatu entity dengan entity lainnya yang memiliki hubungan M:N. Pada relasi ini biasa digunakan tabel bantuan untuk memecahkan relasi tersebut menjadi 1:1 atau 1:M.

B. *Unified Modelling Language*

UML (*Unified Modeling Language*) adalah salah satu alat bantu yang sangat di dunia pengembangan sistem yang berorientasi obyek. Hal ini disebabkan karena UML, menyediakan Bahasa pemodelan visual yang memungkinkan bagi pengembang sistem

untuk membuat cetak biru atau visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.(Munawar, 2018:49)

1. Activity Diagram

Activity Diagram adalah bagian penting dari UML yang menggambarkan aspek dinamis dari sistem. Logika prosedur, proses bisnis dan aliran kerja suatu bisnis bisa dengan mudah dideskripsikan dalam *activity diagram*. *Activity diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa.(Munawar, 2018:127)

2. Use Case Diagram

Use case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara user (pengguna) sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut *scenario*. Setiap *scenario* mendeskripsikan urutan kejadian. Setiap urutan diinisiasikan oleh orang, sistem yang lain, perangkat keras atau urutan waktu. Dengan demikian secara singkat bisa dikatakan *use case* adalah serangkaian *scenario* yang digabungkan bersama-sama oleh tujuan umum pengguna. (Munawar, 2018:89)

3. Class Diagram

Class Diagram adalah diagram statis. Ini mewakili pandangan statis dari suatu aplikasi. *Class diagram* tidak hanya digunakan untuk memvisualisasikan, menggambarkan, dan mendokumentasikan berbagai aspek sistem tetapi juga untuk membangun kode eksekusi (*executable code*) dari aplikasi perangkat lunak. *Class diagram* menggambarkan atribut, operation dan juga constraint yang terjadi pada

sistem. *Class diagram* banyak digunakan dalam pemodelan sistem OO karena mereka adalah salah-satunya diagram UML, yang dapat dipetakan langsung dengan bahasa berorientasi objek.(Munawar, 2018:101)

4. *Sequence Diagram*

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh objek dan *message* (pesan) yang diletakkan diantara objek-objek ini di dalam *use case*. Komponen utama sequence diagram terdiri atas objek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditujukan dengan progress vertical.(Munawar, 2018:137)



