

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

”Sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Sebagai gambaran, jika dalam sistem terdapat elemen yang tidak memberikan manfaat dalam mencapai tujuan yang sama, maka elemen tersebut dapat dipastikan bukanlah bagian dari sistem.” (Kadir, 2014:61).

1. Model Pembelajaran Berbasis Web

a. Pembelajaran berbasis web

Adalah proses pembelajaran yang memanfaatkan jaringan internet, sehingga sering juga disebut sebagai e-learning. Internet adalah jairngan yang terdiri atas ribuan bahkan jutaan komputer, termasuk di dalamnya ada jaringan lokal yang terhubung melalui saluran (satelit, kabel, telepon) dan jangkauannya yang mencakup seluruh dunia, termasuk kegiatan pendidikan.

Fasilitas itu antara lain : *e-mail*, *telnet*, *Internet Relay Chat*, *Newsgroup*, *Mailing List* (Milis), *File Transfer Protocol* (FTP) ataupun *World Wide Web* (WWW).

b. Kelebihan model pembelajaran berbasis *web*

Memungkin setiap orang dimanapun dan kapanpun untuk mempelajari apapun dan menyediakan sumber pembelajaran tambahan yang dapat dipergunakan untuk memperkaya materi pembelajaran.

c. Fungsi model pembelajaran berbasis web

Media pembelajaran merupakan bagian integral dari keseluruhan proses pembelajaran, media pembelajaran dalam penggunaannya harus relevan dengan kompetensi yang ingin dicapai dari isi pembelajaran tersebut, median pembelajaran bukan berfungsi sebagai hiburan, media pembelajaran berfungsi untuk meningkatkan proses belajar.

2. Sistem

“Sistem adalah rangkaian dari satu atau lebih komponen-komponen yang saling berhubungan dan berinteraksi untuk mencapai tujuan. Sebagian besar sistem terdiri dari sub-sistem yang lebih kecil yang mendukung yang lebih besar.” (Reka Putri Halawa, Melanthon Rumapea & Jamaluddin, 2019).

“Pendekatan berorientasi obyek merupakan suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem (sistem perangkat lunak, sistem informasi, dsb). Pendekatan berorientasi obyek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan obyek yang berkorespondasi dengan obyek-obyek di dunia nyata.” (Sukanto dan Shalahuddin, 2014).

3. Program Website

a. Untuk rancangan *website* :

1) Internet

Internet merupakan contoh jaringan terbesar yang menghubungkan jutaan komputer yang tersebar diseluruh penjuru dunia dan tidak terikat pada satu organisasi apapun. Dengan menggunakan jaringan ini, sebuah organisasi dapat melakukan pertukaran informasi secara internal maupun eksternal dengan organisasi-organisasi lain. Jaringan tersusun atas berbagai jenis komputer dan sistem informasi.

(Kadir, 2014:306)

2) Website

“Sistem pengaksesan informasi dalam internet yang *paling terkenal* adalah *World Wide Web (WWW)*. atau yang biasa dikenal dengan web. Merupakan salah satu protokol yang disebut HTTP (*hyper text transfer protocol*) yang berjalan pada TCP/IP.” (Kadir, 2014). Sedangkan menurut Wahyu (2012) “Yang dimaksud dengan aplikasi *web* atau aplikasi berbasis *web (Web-based-application)* adalah aplikasi yang dijalankan melalui *browser*”. Aplikasi ini pertama kali dibangun hanya dengan menggunakan bahasa yang disebut HTML (*Hyper Text Markup Language*) dan protokol yang digunakan dinamakan HTTP (*Hyper Text Transfer Protocol*). Namun tentu saja hal seperti ini memiliki kelemahan. Semua perubahan harus dilakukan pada tingkat aplikasi. Pada perkembangan berikutnya, sejumlah *script* dan objek dikembangkan untuk memperluas kemampuan HTML. Pada saat ini, banyak *script* seperti itu, antara lain: PHP, ASP, ASP.NET, sedangkan contoh yang berupa objek antara lain adalah applet (java).

Dengan memperluas kemampuan HTML, yakni dengan menggunakan perangkat lunak tambahan, perubahan informasi dalam halaman-halaman *web* dapat ditangani melalui perubahan bukan melalui perubahan bahan program. Sebagai implementasinya, aplikasi *web* dapat dikoneksikan ke *database*. Dengan demikian, perubahan informasi dapat dilakukan oleh operator atau yang bertanggung jawab terhadap kemutakhiran data, dan tidak menjadi tanggung jawab pemrogram atau *webmaster*.

Konsep yang mendasari aplikasi *web* sebenarnya sederhana, operasi yang melatarbelakanginya melibatkan pertukaran informasi antara komputer yang meminta informasi, yang disebut klien, dan komputer yang memasok informasi (atau disebut *server*). Secara lebih rinci, *server* yang melayani permintaan dari klien sesungguhnya berupa suatu perangkat lunak yang dinamakan *web server*. Secara internal, *web server* inilah yang berkomunikasi dengan perangkat lunak lain yang disebut *middleware* dan *middleware* inilah yang berhubungan dengan *database*. Model seperti inilah yang mendukung *web* dinamis.

Dengan menggunakan pendekatan *web* dinamis, dimungkinkan untuk membentuk aplikasi berbasis *web* yang berinteraksi dengan *database*.

3) *Web Browser*

Merupakan program aplikasi yang digunakan untuk memudahkan melakukan navigasi berbagai data dan informasi pada WWW dan mengakses layanan WWW. *Browser* dapat memandu pengguna internet untuk berpindah antar situs web dengan mudah. *Browser* yang paling terkenal penggunaannya adalah Google Chrome. Internet Explorer dan Mozilla Firefox.

4) HTML (*Hyper Text Markup Language*)

Menurut Andi (2012a:1) yang memberikan batasan bahwa “*HTML (Hyper Text Markup Language)* merupakan sebuah bahasa program yang memungkinkan anda membuat halaman *web* yang bisa ditampilkan dalam browser (Internet Explorer, Google Chrome, Opera, Firefox dll)” .

Kode *HTML* diawali dengan `<html>` dan diakhiri dengan `</html>`, namun sebelum `<html>` diharuskan untuk diberikan deklarasi “*!DOCTYPE*”, fungsi deklarasi ini adalah sebagai *public text identifier*.

5) *Web Server*

Web server atau *server* adalah komputer yang dikhususkan untuk menaruh data *website*, komputer *server* harus 24 jam *online*, jika tidak maka data tak bisa diakses oleh pengunjung *website*. Semua komputer logikanya bisa dijadikan *server*, namun *server* yang khusus untuk *website* punya spesifikasi khusus. Hal ini untuk menjaga agar kecepatan *server* stabil dan memenuhi kriteria yang dihadapkan. Selain bergantung pada spesifikasi *hardware* dan *software*, kecepatan *webservice* juga bergantung pada akses *internet* di tempat *server* berada atau juga tergantung pada kecepatan *Internet Service Provider* (ISP) yang digunakan untuk *online*, dan terlebih lagi juga bergantung pada jalur lokasi negara yang bersangkutan.

6) PHP

Menurut Andi (2013b:171) memberikan batasan bahwa “PHP (*Hypertext Preprocessor*) adalah pemrograman yang bekerja dalam sebuah *webservice*”. *Script-script* PHP yang dibuat tersimpan dalam sebuah *server* dan diproses dalam *server* tersebut. Dengan menggunakan program PHP, sebuah *web* akan lebih interaktif dan dinamis. Pada suatu halaman *web* dinamis, PHP berfungsi sebagai bahasa pemrograman yang menjalankan suatu perintah tertentu. PHP merupakan *software open source* yang bisa digunakan oleh siapa saja

secara bebas tanpa harus membayar. PHP juga merupakan *software cross platform* sehingga bisa jalan dengan baik pada *windows* maupun *unix (linux)*.

7) *Javascript*

Menurut Kadir (2014:8) “*JavaScript* adalah bahasa skrip (bahasa yang kodenya ditulis menggunakan teks biasa) yang ditempelkan pada dokumen HTML dan diproses pada sisi klien”. Dengan adanya bahasa ini kemampuan dokumen HTML menjadi semakin luas. Beberapa contoh yang bias dilakukan melalui *JavaScript*:

- a) Menampilkan jam lokal pada halaman *web*.
- b) Mengatur warna latar belakang halaman *web*.
- c) Mengganti gambar ketika pemakai menempatkan penunjuk *mouse* ke suatu gambar.
- d) Memvalidasi keabsahan data yang dimasukan oleh pemakai.

4. **Basis Data**

“Basis data (*database*) adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi.

Basis data dimaksudkan untuk mengatasi masalah (*problem*) pada sistem yang memakai pendekatan berbasis berkas.” Kadir (2014:218).

a. **MySQL**

SQL (*Structured Query Language*) adalah database yang cepat dan tangguh, sangat cocok jika digabungkan dengan PHP, dengan database kita bisa menyimpan, mencari dan mengklarifikasikan data dengan lebih akurat dan profesional. MySQL menggunakan SQL *Language (Structure Query Language)*. SQL terdiri dari dua bahasa, yaitu *Data Definition Language (DDL)* dan *Data Manipulation Language (DML)*.

Fathur (2012).

b. XAMPP

XAMPP merupakan singkatan dari **X** (tempat sistem operasi apapun) Apache, **M**MySQL, **P**PHP, dan **P**Perl. XAMPP merupakan *tool* yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dalam paketnya sudah terdapat Apache (*web server*), MySQL (*database*), PHP (*server side scripting*), Perl, FTP server, PhpMyAdmin dan berbagai pustaka bantu lainnya. dengan menginstall XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasinya web server Apache, Php dan MySQL secara manual. XAMPP akan menginstalasi dan mengkonfigurasi secara otomatis.

Asumsi lain adalah “Paket aplikasi yang memudahkan dalam penginstalan modul PHP, Apache *Web Server*, dan database MySQL.” (Sugiri, 2014). Selain itu XAMPP dilengkapi dengan berbagai fasilitas lain yang akan memberikan kemudahan dalam mengembangkan situs *web* berbasis PHP. XAMPP merupakan aplikasi gratis dan tersedia untuk *platform* Linux, Windows, Mac OS, dan Solaris. Aplikasi ini dikembangkan oleh Kay Vogelgesang, Carsten Wiedmann, dan Kai ‘Oswald’ Seider di bawah lisensi GNU (*General Public License*).

c. PhpMyAdmin

Menurut Nugroho (2014a:13) memberikan batasan bahwa “PhpMyAdmin adalah aplikasi berbasis *web* yang dibuat dari pemrograman *PHP* dan diramu dengan *JavaScript*”. PhpMyAdmin juga dapat disebut sebagai *tools* yang berguna untuk mengakses *database* MySQL *Server* dalam bentuk tampilan *web*. *Tools* ini secara standar telah

disertakan pada saat menginstal paket PHP, seperti XAMPP ataupun Apache2Triad.

2.2. Teori Pendukung

1. *Entity Relationship Diagram (ERD)*

“Pemodelan awal basis data yang paling banyak digunakan adalah *Entity Relationship Diagram (ERD)* yang dikembangkan berdasarkan teori himpunan dalam bidang matematika, ERD digunakan untuk pemodelan basis data relasional,” Menurut Sukamto dan Shalahuddin (2014:50).

Elemen-Elemen Diagram Hubungan Entitas:

a. *Entity*

Pada *E-R* diagram, *entity* digambarkan dengan sebuah bentuk persegi panjang. *Entity* merupakan data yang akan disimpan.

b. *Relationship*

Pada *E-R* diagram, *relationship* dapat digambarkan dengan sebuah bentuk belah ketupat. *Relationship* merupakan hubungan alamiah yang terjadi antara entitas. Pada umumnya penghubung (*relationship*) diberi nama dengan kata kerja dasar, sehingga memudahkan untuk melakukan pembacaan relasinya (bisa dengan kalimat aktif atau kalimat pasif). Penggambaran hubungan yang terjadi adalah sebuah bentuk belah ketupat dihubungkan dengan dua bentuk empat persegi panjang.

c. *Relationship Degree*

Relationship degree atau derajat *relationship* merupakan jumlah entitas yang berpartisipasi dalam satu *relationship*.

Derajat *relationship* yang sering dipakai di dalam *ERD*:

1) *Unary Relationship*

Unary relationship merupakan model *relationship* yang terjadi diantara *entity* yang berasal dari *entity set* yang sama.

2) *Binary Relationship*

Binary relationship merupakan model *relationship* antara *instance-instance* dari suatu tipe entitas (dua *entity* yang berasal dari *entity* yang sama). *Relationship* ini paling umum digunakan dalam pembuatan model data.

3) *Ternary Relationship*

Ternary relationship merupakan *relationship* antara *instance-instance* dari tiga tipe entitas secara sepihak.

d. Atribut

Secara umum atribut merupakan sifat atau karakteristik dari tiap entitas maupun tiap *relationship*. Maksudnya, atribut yang menjelaskan apa sebenarnya yang dimaksud entitas maupun *relationship*, sehingga sering dikatakan atribut elemen sebagai dari setiap entitas dan *relationship*.

Atribut *Value*

Attribute value atau nilai atribut suatu *occurrence* tertentu dari sebuah *attribute* di dalam suatu *entity* atau *relationship*. Ada dua jenis atribut:

1) *Identifier (key)*

Digunakan menentukan suatu *entity* secara unik (*primary key*).

2) *Descriptor (nonkey attribute)*

Digunakan untuk menspesifikasikan karakteristik dari suatu *entity* yang tidak unik.

e. Kardinalitas (*Cardinality*)

Kardinalitas relasi menunjukkan jumlah maksimum tupel yang dapat berelasi dengan entitas pada entitas yang lain. Dari sejumlah kemungkinan banyaknya hubungan antar entitas tersebut, kardinalitas relasi merujuk kepada hubungan maksimum yang terjadi dari entitas yang satu ke entitas yang lain dan begitu juga sebaliknya. Terdapat tiga macam kardinalitas relasi, yaitu:

1) *One to One*

Tingkat hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya.

2) *One to Many* atau *Many to One*

Tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu. Tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua. Sebaliknya

pada yang kedua hanya dapat mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama.

3) *Many to Many*

Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya. Baik dilihat dari sisi entitas yang pertama, maupun dilihat dari sisi yang kedua. Yang berarti setiap tupel pada entitas A dapat berhubungan dengan banyak tupel pada entitas B, dan demikian juga sebaliknya, dimana setiap tupel pada entitas B dapat berhubungan dengan banyak tupel pada entitas A.

Langkah-langkah membuat ERD :

- a) Tentukan entitas-entitas yang diperlukan.
- b) Tentukan *relationship* antar entitas-entitas tersebut.
- c) Tentukan *cardinality* dan *participation constraint*.
- d) Tentukan atribut-atribut yang diperlukan dari setiap entitas.
- e) Tentukan *key* di antara atribut-atribut tersebut.

Hindari penamaan *entity*, *relationship* dan *attribute* yang sama.

f. *Participation Constraint*

Menunjukkan batasan keterlibatan suatu entitas dalam suatu *relationship*.

Participation Constraint terdiri dari dua jenis, yaitu :

- 1) *Total*
- 2) *Partial*

2. *Unified Modelling Language (UML)*

Unified Modeling Language (UML) adalah penerus gelombang analisis berorientasi objek dan desain (OOA & D) metode yang muncul di akhir '80-an dan awal 90-an. Hal yang paling langsung menyatukan metode Booch, Rumbaugh (OMT), dan Jacobson, namun jangkauannya lebih luas dari itu. UML telah melalui proses standarisasi dengan OMG (*Object Management Group*) dan sekarang menjadi *OMG standard*. Fowler (2014:1a).

a. *Activity Diagram*

Activity diagram adalah salah satu bagian yang paling tak terduga dari UML. Tidak seperti kebanyakan teknik lain dalam UML, diagram aktivitas tidak memiliki asal-usul yang jelas dalam karya-karya sebelumnya dari tiga amigos. Sebaliknya, diagram aktivitas menggabungkan ide dari beberapa teknik: diagram acara Jim Odell, SDL teknik pemodelan negara, pemodelan alur kerja, dan jaring Petri. Diagram ini sangat berguna dalam hubungannya dengan alur kerja dan perilaku yang memiliki banyak pemrosesan paralel menjelaskan.

Fowler (2014:129).

b. *Use Case Diagram*

Gunakan kasus adalah fenomena menarik. Untuk waktu yang lama, baik dalam pengembangan berorientasi objek dan tradisional, orang-orang menggunakan interaksi yang khas untuk membantu mereka memahami persyaratan. Namun, skenario ini diperlakukan sangat informal-selalu dilakukan tapi jarang di dokumentasikan. Ivar Jacobson dikenal baik untuk mengubah ini dengan metode *Objectory* dan buku terkait (satu pertama).

Fowler (2014:39).

c. *Sequence Diagram*

Sequence Diagram adalah diagram yang menjelaskan suatu interaksi antar objek (pengguna, *display* dsb) didalam sebuah sistem untuk setiap kegiatan (*case*) yang dilakukan oleh seorang actor berdasarkan urutan kejadian.

d. Class Diagram

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut dan operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan.

(Handini, 2016).

e. Component Diagram

Sebuah komponen merupakan modul fisik kode. Sebuah komponen sering sama sebagai sebuah paket, tetapi mungkin berbeda, karena komponen mewakili kemasan kode fisik. Dengan demikian, satu kelas dapat hadir dalam beberapa komponen, tetapi kelas yang dapat di definisikan hanya dalam satu paket. Sebagai contoh, kelas string *java* merupakan bagian dari paket *java.lang*, tapi ternyata di banyak komponen.

Fowler (2005:143).

f. Deployment Diagram

Sebuah diagram penyebaran menunjukkan hubungan fisik di antara peranti lunak dan komponen perangkat keras dalam sistem disampaikan. Diagram *deployment* adalah tempat yang baik untuk menunjukkan bagaimana komponen dan objek yang diarahkan dan bergerak di sekitar sistem terdistribusi. Setiap node pada diagram penyebaran merupakan semacam *compu-tational* unit dalam banyak kasus, sebuah perangkat keras. Perangkat keras mungkin perangkat sederhana atau sensor, atau bisa juga *mainframe*.

Fowler (2005:141).