

BAB 2

TINJAUAN PUSTAKA

2.1 Pengertian Sistem

Yakub menuliskan dalam bukunya (Yakub, 2012) bahwa sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, terkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk tujuan tertentu. Prosedur itu antara lain:

- a. Prosedur adalah urutan-urutan operasi klerikal atau tulis menulis yang melibatkan beberapa orang di dalam suatu departemen untuk menjamin penanganannya yang seragam.
- b. Prosedur adalah urutan-urutan yang tepat dari tahapan-tahapan instruksi yang menerangkan apa (*what*) yang harus dikerjakan, siapa (*who*) yang mengerjakan, kapan (*when*) dikerjakan, dan bagaimana (*how*) mengerjakannya.

2.1.1 Elemen-elemen Sistem

Yakub menuliskan dalam bukunya (Yakub, 2012) bahwa tidak semua sistem memiliki kombinasi elemen-elemen yang sama, tetapi susunan dasarnya sama. Ada beberapa elemen yang membentuk suatu sistem, yaitu:

- a. Tujuan, ini menjadi motivasi yang mengarahkan pada sistem, karena tanpa tujuan yang jelas sistem menjadi tak terarah dan tak terkendali.

- b. Masukan, (*input*) sistem adalah segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan untuk diproses. Masukan dapat berupa hal-hal berwujud maupun yang tidak berwujud. Masukan berwujud adalah bahan mentah, sedangkan yang tidak berwujud adalah data.
- c. Proses, merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna dan lebih bernilai.
- d. Keluaran, (*output*) merupakan hasil dari pemrosesan sistem dan keluaran dapat menjadi masukan untuk subsistem lain.
- e. Batasan, (*boundary*) sistem adalah pemisah antara sistem dan daerah di luar sistem. Batasan sistem menentukan konfigurasi, ruang lingkup, dan kemampuan sistem.
- f. Mekanisme pengendalian dan umpan balik, mekanisme pengendalian (*control mechanism*) diwujudkan dengan menggunakan umpan balik (*feedback*), sedangkan umpan balik ini digunakan untuk mengendalikan masukan maupun proses. Tujuannya untuk mengatur agar sistem berjalan sesuai dengan tujuan.
- g. Lingkungan, adalah segala sesuatu yang berada di luar sistem.

2.1.2 Klasifikasi Sistem

Yakub menuliskan dalam bukunya (Yakub, 2012) bahwa sistem dapat diklasifikasikan dari beberapa sudut pandang. Klasifikasi sistem tersebut diantaranya: sistem abstrak (*abstract system*), sistem fisik (*physical system*), sistem tertentu (*deterministic system*), sistem tak tentu (*probabilistic system*), sistem tertutup (*close system*), dan sistem terbuka (*open system*).

- a. Sistem tak tentu (*probabilistic system*), adalah suatu sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.
- b. Sistem abstrak (*abstract system*), adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik.
- c. Sistem fisik (*physical system*), adalah sistem yang ada secara fisik.
- d. Sistem tertentu (*deterministic system*), adalah sistem yang beroperasi dengan tingkah laku yang tidak dapat diprediksi, interaksi antara bagian dapat didekati dengan pasti sehingga keluarannya dapat diramalkan.
- e. Sistem tertutup (*close system*), adalah sistem yang tidak bertukar materi, informasi, atau energy dengan lingkungan.
- f. Sistem terbuka (*open system*), adalah sistem yang berhubungan dengan lingkungan dan dipengaruhi oleh lingkungan.

2.2 Pengertian Sistem Informasi

Sistem informasi (*information system*) merupakan kombinasi teratur dari orang-orang, perangkat keras (*hardware*), perangkat lunak (*software*), jaringan komunikasi, dan sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi. Orang tergantung pada sistem informasi untuk berkomunikasi antara satu sama lain dengan menggunakan berbagai jenis alat fisik, perintah dan prosedur pemrosesan informasi, saluran telekomunikasi atau jaringan, dan data yang disimpan atau sumber daya data (Yakub, 2012).

Sistem Informasi adalah suatu kumpulan dari komponen-komponen dalam perusahaan atau organisasi yang berhubungan dengan proses penciptaan dan pengaliran informasi. Sistem informasi dapat juga didefinisikan sebagai suatu sistem yang menerima sumber data sebagai *input* dan mengolahnya menjadi produk informasi

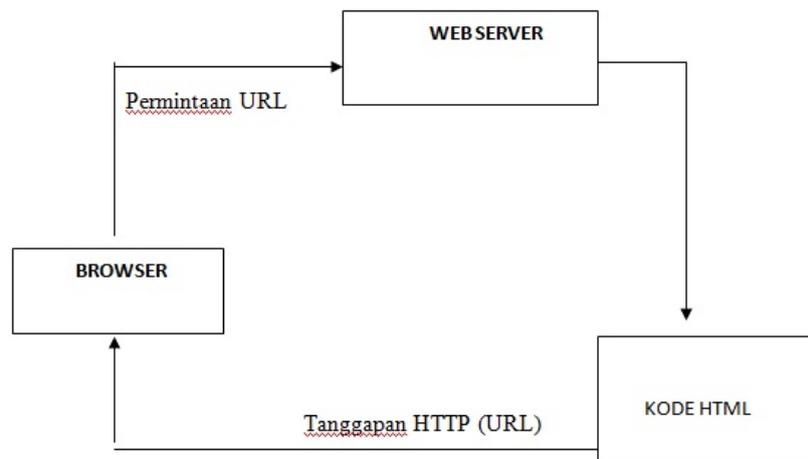
sebagai *output*. Sistem informasi merupakan suatu sistem yang terdiri dari beberapa subsistem atau komponen *hardware*, *software*, dan *brainware*, data dan prosedur untuk menjalankan *input*, proses, *output*, penyimpanan, dan pengontrolan yang mengubah sumber data menjadi informasi. Atau dapat juga didefinisikan sebagai suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Marimin, et-al. 2006).

2.3 PHP (*Hypertext preprocessor*)

Agus Saputra menuliskan dalam bukunya (Saputra, 2012) bahwa PHP (Hypertext Preprocessor) adalah suatu bahasa pemrograman yang difungsikan untuk membangun suatu web dinamis. PHP menyatu dengan kode HTML yang artinya dimana HTML digunakan sebagai pembangun atau pondasi dari kerangka *layout* web, sedangkan PHP digunakan sebagai prosesnya, sehingga dengan adanya PHP tersebut sebuah web akan sangat mudah dipelihara.

PHP berjalan pada sisi server, sehingga PHP disebut juga sebagai bahasa "*server side scripting*", artinya bahwa untuk menjalankan PHP wajib membutuhkan web *server* dalam menjalankannya.

PHP bersifat *open source*, sehingga dapat dipakai secara cuma-cuma dan mampu digunakan pada berbagai platform yakni sistem operasi Windows maupun Linux. PHP juga dibangun sebagai modul pada web server Apache dan sebagai *binary* yang dapat berjalan sebagai CGI. Cara kerja PHP ditunjukkan pada Gambar 2.1.



Gambar 2.1: Cara Kerja PHP (Saputra, 2012)

Pada Gambar 2.1 dapat dijelaskan cara kerja PHP yaitu:

- a. Server membaca permintaan dari *client/browser*.
- b. Kemudian dilanjutkan untuk mencari halaman/page pada server.
- c. Server melakukan instruksi yang diberikan oleh PHP untuk melakukan modifikasi pada halaman/*page*.
- d. Selanjutnya hasil modifikasi tersebut akan dikembalikan kepada *client/browser*.

2.4 Basis Data

Basis data (*database*) dapat diartikan sebagai kumpulan data tentang suatu benda atau kejadian yang saling berhubungan satu sama lain. Sedangkan data merupakan fakta yang mewakili suatu objek seperti manusia, hewan, peristiwa, keadaan dan lain sebagainya yang dapat dicatat dan mempunyai arti yang implisit. Data dicatat

atau direkam dalam bentuk angka, huruf, simbol, gambar, bunyi, atau kombinasinya (Triasdi, 2011).

Sistem manajemen basis data adalah sistem yang berbentuk suatu rangkaian dari metode yang memungkinkan pemberian defenisi, penciptaan, perubahan, pembacaan, pengendalian, pemeliharaan, dan perlindungan terhadap basis data (Triasdi, 2011).

2.5 Use Case Diagram

Dalam buku yang dituliskan oleh Jeffery L. Whitten, Lonnie D. Bentley, Kevin C. Dittman (Whitten, et-al. 2004) bahwa diagram yang menggambarkan interaksi antara sistem dengan eksternal sistem dan pengguna. Dengan kata lain, secara grafis menggambarkan siapa yang akan menggunakan sistem dan dengan cara apa pengguna mengharapkan untuk berinteraksi dengan sistem. Dalam *use case diagram* memiliki pemodelan sebagai berikut:

1. Use Case

Use case merupakan urutan langkah-langkah yang secara tindakan saling terkait (*scenario*), baik otomatis maupun secara manual.

2. Actor (Pelaku)

Actor merupakan segala sesuatu yang perlu berinteraksi dengan sistem untuk pertukaran informasi.

3. Relationship (Hubungan)

Pada diagram *use case*, *relationship* digambarkan sebagai sebuah garis antara dua simbol. Pemaknaan *relationship* berbeda-beda tergantung bagaimana garis tersebut digambar dan tipe simbol apa yang digunakan untuk

menghubungkan garis tersebut. Berikut ini adalah perbedaan di antara *relationship* yang ada pada sebuah diagram *use case*:

a. *Association*

Relationship antara *actor* dengan *use case* dimana terjadi interaksi di antara mereka.

b. *Extends*

Use case yang terdiri dari langkah yang terekstraksi dari *use case* yang lebih kompleks untuk menyederhanakan masalah dan karena itu memperluas fungsinya.

c. *Uses (includes)*

Hubungan *uses* menggambarkan bahwa satu *use case* seluruhnya meliputi fungsionalitas dari *use case* lainnya.

d. *Depends On*

Terkadang suatu *use case* memiliki ketergantungan pada *use case* yang lainnya yang bertujuan untuk menentukan urutan dalam pengembangan *use case*. Ketergantungan ini dimodelkan menggunakan *depends on relationship*.

e. *Inheritance*

Hubungan *inheritance* terjadi ketika dua atau lebih *actor* menggunakan *use case* yang sama.

2.6 *Data Flow Diagram*

Dalam buku yang dituliskan oleh Jeffery L. Whitten, Lonnie D. Bentley, Kevin C. Dittman (Whitten, et-al. 2004) bahwa DFD pertama dikembangkan oleh Chris Gane dan Trish Sarson pada tahun 1979. Sistem ini dikembangkan berbasis pada dekomposisi fungsional dari sebuah sistem. Edward Yourdon dan Tom DeMarco

memperkenalkan metode yang lain pada tahun 1980-an dimana mengubah persegi dengan sudut lengkung (pada DFD Chris Gane dan Trish Sarson) dengan lingkaran untuk menotasikan. Informasi yang ada dalam perangkat lunak dimodifikasi dengan beberapa transformasi yang dibutuhkan.

DFD dapat digunakan untuk merepresentasikan sebuah sistem atau perangkat lunak pada beberapa level abstraksi. DFD dapat dibagi menjadi beberapa level yang lebih detail untuk merepresentasikan aliran informasi atau fungsi yang lebih detail. DFD menyediakan mekanisme untuk pemodelan fungsional ataupun pemodelan aliran informasi. DFD lebih sesuai digunakan untuk memodelkan fungsi-fungsi perangkat lunak yang akan diimplementasikan menggunakan pemrograman terstruktur karena pemrograman terstruktur membagi-bagi bagiannya dengan fungsi-fungsi dan prosedur-prosedur.

DFD pertama kali digambarkan merupakan level teratas (*top level*) dan diagram ini disebut *context diagram*. Diagram kemudian akan digambar lebih rinci lagi yang disebut *overview diagram* dan seterusnya. Tahapan perancangan menggunakan DFD lebih rinci sebagai berikut:

1. Membuat DFD Level 0 atau sering disebut juga dengan *Context Diagram*

DFD level 0 menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. DFD level 0 digunakan untuk menggambarkan interaksi antar sistem yang akan dikembangkan dengan entitas luar. Penomoran proses yang ada di dalamnya menggunakan angka 0 dan hanya ada satu proses didalamnya. Pada diagram ini belum diperbolehkan adanya data store karena diagram ini menggambarkan sistem secara keseluruhan.

2. Membuat DFD level 1

DFD level 1 digunakan untuk menggambarkan modul-modul yang ada dalam

sistem yang dikembangkan. DFD level 1 merupakan hasil *breakdown* DFD level 0 yang sebelumnya sudah dibuat. DFD level 1 juga disebut sebagai Diagram Nol/ Zero atau lebih biasa disebut sebagai *Overview Diagram*. Diagram nol memberikan pandangan secara menyeluruh mengenai sistem yang ditangani, menunjukkan tentang fungsi-fungsi utama atau proses yang ada, aliran data, dan entitas luar. Pada level ini sudah dimungkinkan digambarkannya data store yang digunakan. Penomoran proses DFD level 1 dimulai dengan angka 1.0, 2.0, 3.0, dan seterusnya.

2.7 Entity Relationship Diagram (ERD)

Abdul Kadir menuliskan dalam bukunya (Kadir, 2003) bahwa model E-R (Diagram ER) adalah suatu model yang digunakan untuk menggambarkan data dalam bentuk entitas, atribut dan hubungan antar entitas. Huruf E menyatakan entitas dan R menyatakan hubungan (dari kata *Relationship*). Model ini dinyatakan dalam bentuk diagram sehingga sering disebut sebagai ERD.

Singkat kata ERD adalah suatu model penyajian data dengan menggunakan entitas dan hubungan. ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol. Pada dasarnya ada tiga simbol yang digunakan, yaitu:

1. Entitas

Entitas merupakan objek dari sesuatu yang nyata maupun abstrak. Simbol dari entitas ini biasanya digambarkan dengan persegi panjang.

2. Atribut

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

3. Relasi (Hubungan)

Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Biasanya digambarkan dengan simbol panah. ERD dibuat untuk merancang suatu basis data dengan memperlihatkan hubungan antara entity atau objek yang terlibat beserta atributnya. Penjelasan dari masing-masing jenis hubungan tersebut adalah sebagai berikut:

a. Hubungan *one-to-one*

Setiap entitas pada tipe entitas A paling banyak berpasangan dengan satu entitas pada tipe entitas B, begitu pula sebaliknya.

b. Hubungan *one-to-many*

Setiap entitas pada tipe entitas A bisa berpasangan dengan banyak entitas pada tipe entitas B, sedangkan setiap entitas pada entitas B hanya dapat berpasangan dengan satu entitas pada entitas A.

c. Hubungan *many-to-one*

Entitas pada tipe entitas A paling banyak berpasangan dengan satu entitas pada tipe entitas B dan setiap entitas pada tipe entitas B dapat berpasangan dengan banyak entitas pada tipe entitas A.

d. Hubungan *many-to-many*

Setiap entitas pada suatu tipe entitas A bisa berpasangan dengan banyak entitas pada tipe entitas B dan begitu pula sebaliknya.

2.8 Pengertian Kuesioner

Joko Subagyo menuliskan dalam bukunya (Subagyo, 2006) bahwa kuesioner juga merupakan alat pengumpulan data. Kuesioner diajukan kepada responden dalam bentuk tertulis disampaikan secara langsung ke alamat responden, kantor, atau tempat lain.

Kuesioner atau sistem angket mempunyai kelebihan tersendiri apabila dibandingkan alat bantu lainnya, seperti misalnya dengan cara wawancara yang mempunyai kemampuan jelajah terbatas pada keadaan pewawancara. Kuesioner dapat disebar luaskan sesuai keperluan pada setiap responden dalam waktu relative singkat dengan mengerahkan seluruh jajaran peneliti untuk membagikannya secara langsung atau dikirimkan lewat pos ke alamat responden.

Tujuan pembuatan kuesioner adalah:

- a. Lebih mengarahkan informasi yang diperoleh secara relevan sehingga terhindar data tidak terpakai.
- b. Membantu responden memberikan jawaban dalam waktu relative lebih cepat dibandingkan cara lain.
- c. Mengarah dalam pemakaian analisa kuantitatif sebagai maksud utama, ditunjang analis kualitatif atau sebaliknya.
- d. Mempercepat pengumpulan data.

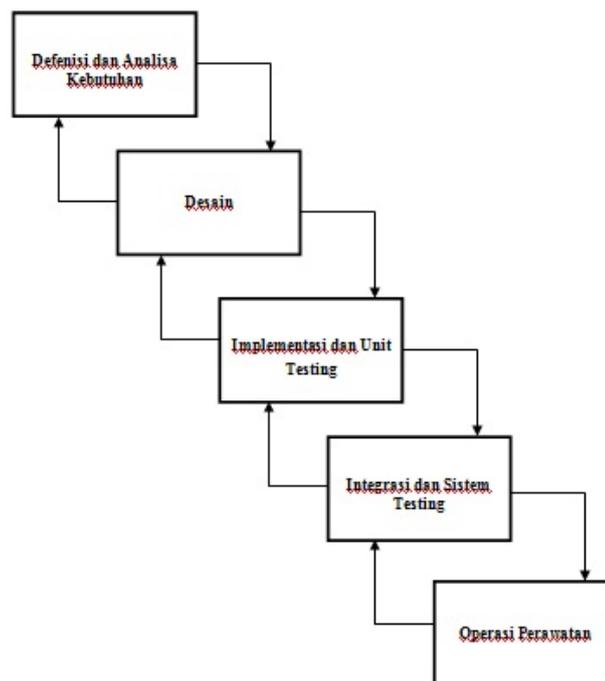
Sofian Effendi menuliskan dalam bukunya (Effendi, 2008) bahwa tujuan pokok pembuatan kuesioner adalah:

- a. Memperoleh informasi yang relevan dengan tujuan survei.
- b. Memperoleh informasi dengan reliabilitas dan validitas setinggi mungkin.

Mengingat terbatasnya masalah yang dapat ditanyakan dalam kuesioner, maka senantiasa perlu diingat agar pertanyaan-pertanyaan memang langsung berkaitan dengan hipotesis dan tujuan penelitian tersebut.

2.9 Metode Waterfall

Sommerville menyebut *Linear Sequential Model* atau juga dikenal dengan metode *Waterfall* merupakan metode yang klasik dalam pembangunan rekayasa perangkat lunak, menurut Sommerville fase-fase metode *Waterfall* memiliki lima tahap yaitu:



Gambar 2.2: Fase-fase dalam metode *Waterfall* (Sommerville, 2001)

Pada Gambar 2.2 dapat dijelaskan fase-fase dalam metode *Waterfall* yaitu:

1. Tahap Defenisi dan Analisa Kebutuhan dilakukan dengan menentukan konsep dan tujuan yang jelas, yang mana harus mendefenisikan kegunaan dari pembuatan sistem kuesioner layanan ISO ini dan mengetahui secara jelas kebutuhan yang harus dipenuhi oleh sistem kuesioner layanan ISO yang akan dibangun. Fase ini harus dikerjakan secara lengkap untuk bisa menghasilkan

desain yang lengkap.

2. Tahap Desain dilakukan setelah kebutuhan selesai dikumpulkan secara lengkap, ditahap ini yang dilakukan adalah merancang spesifikasi secara rinci mengenai struktur sistem kuesioner layanan ISO yang akan dibuat. Spesifikasi dibuat secara rinci sehingga pada tahap berikutnya, yaitu tahap pengumpulan bahan dan pembuatan tidak dibutuhkan keputusan baru, melainkan menggunakan apa yang telah ditetapkan pada tahap desain. Namun demikian, sering terjadi penambahan atau pengurangan bahan, bahkan ada perubahan pada bagian aplikasi pada awal pengerjaan sistem.
3. Tahap Implementasi dan Unit Testing merupakan tahap desain program diterjemahkan kedalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Program yang dibangun langsung diuji dengan baik.
4. Tahap Operasi dan Perawatan merupakan tahap mengoperasikan program dilingkungannya dan melakukan pemeliharaan, seperti penyesuaian atau perubahan karena adaptasi dengan situasi sebenarnya (Sommerville, 2001).

Menurut Jurnal yang dituliskan oleh Unnati dan Jain (Unnati dan Niky. 2013) bahwa tahapan dalam model Waterfall adalah:

1. Kebutuhan secara spesifikasi (identifikasi dan analisis persyaratan sistem)
2. *Software* Desain (*high level design and detail level design*)
3. Pelaksanaan (*coding, debugging and unit testing*)
4. Pengujian (*Integration Testing* dan pengujian sistem)
5. Pemeliharaan (sistem ini siap digunakan dan dikelola jika diperlukan).