

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

2.1.2 Pengertian Sistem

Suatu sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai suatu tujuan tertentu. Secara sederhana, suatu sistem diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel, yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu.

2.1.3 Karakteristik Sistem

Model umum sebuah sistem menurut Tata Sutabri (2012:20), terdiri dari input, proses, dan output. Hal ini merupakan konsep sebuah sistem yang sangat sederhana mengingat sebuah sistem dapat mempunyai beberapa masukan dan keluaran sekaligus. Selain itu sistem juga mempunyai karakteristik antara lain :

1. **Komponen Sistem (Components)**

Sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. **Batasan Sistem (Boundary)**

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem

ini memungkinkan sistem dipandang sebagai suatu kesatuan yang tidak dapat dipisahkan.

3. Lingkungan Luar Sistem (Environment)

Bentuk apapun yang ada di ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem.

4. Penghubung Sistem (Interface)

Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lain. Dengan demikian terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukan Sistem (Input)

Input merupakan energi yang dimasukkan ke dalam sistem. Input dapat berupa maintenance input dan signal input. Maintenance input adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Signal input adalah energi yang diproses untuk menghasilkan output.

6. Keluaran Sistem (Output)

Output merupakan hasil dari energi yang diolah dan diklasifikasikan menjadi output yang berguna dan sisa pembuangan. Output dapat menjadi input untuk subsistem yang lain.

7. Pengolah Sistem (Processing System)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

8. Sasaran Sistem (Objective)

Suatu sistem mempunyai tujuan (goal) atau sasaran (objective), sasaran dari sistem menentukan input yang dibutuhkan dan output yang akan dihasilkan.

2.2 Konsep Dasar *Website*

2.2.1 *Website*

Menurut Murad (2013: 49), *website* adalah sistem dengan informasi yang disajikan dalam bentuk teks, gambar, suara, dan lainnya yang tersimpan dalam sebuah *server web* internet yang disajikan dalam bentuk *hypertext*. Dapat disimpulkan bahwa *website* adalah sebuah tempat di internet yang menyajikan informasi dengan berbagai macam format data seperti teks, gambar, bahkan video yang dapat diakses menggunakan berbagai aplikasi klien sehingga memungkinkan penyajian informasi yang lebih menarik dan dinamis dengan pengelolaan yang terorganisasi. Terdapat beberapa teknologi dalam aplikasi berbasis *website* diantaranya yang cukup sering digunakan adalah *web server*, *web hosting*, dan *web browser*.

2.2.2 *Web Server*

Solichin (2016: 6) menyatakan *web server* atau *server web* merupakan suatu perangkat lunak server yang berfungsi menerima permintaan dari klien yang dikenal dengan *web browser* dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman web melalui protokol HTTP atau HTTPS. *Web server* bertugas mengelola halaman-halaman web dan dokumen-dokumen lainnya. Jadi, pada dasarnya *web server* hanyalah sebuah perangkat lunak yang bisa saja berada di komputer apapun dengan spesifikasi apapun. Komputer biasapun dapat berfungsi sebagai *web server*.

2.2.3 *Web Hosting*

Web hosting dapat diartikan sebagai ruangan yang terdapat dalam suatu media penyimpanan untuk menyimpan berbagai data, file-file, gambar, dan lain sebagainya yang nanti akan ditampilkan di *website*. Besarnya data yang dapat dimasukkan tergantung dari besarnya *web hosting* yang disewa atau dimiliki, semakin besar *web hosting* maka semakin besar pula data yang dapat dimasukkan dan ditampilkan dalam *website*. *Web Hosting* juga dapat diperoleh dengan menyewa, besarnya hosting ditentukan dengan ukuran ruangan media penyimpanan ada yang MB (*Mega Byte*) atau GB (*Giga Byte*). Waktu penyewaan *web hosting* rata-rata dihitung per tahun. Penyewaan *hosting* dilakukan dari perusahaan-perusahaan penyewa *web hosting* yang banyak dijumpai baik Indonesia maupun luar negeri.

2.2.4 *Web Browser*

Solichin (2016: 9) menyatakan *web browser* atau peramban web merupakan perangkat lunak yang berfungsi untuk menerima dan menyajikan sumber informasi di internet. Mudah-mudahan *web browser* digunakan untuk menampilkan halaman-halaman web.

2.3 Basis Data (*Database*)

Menurut Indrajani (2015: 70), basis data adalah kumpulan data yang saling berhubungan secara logis dan didesain untuk mendapatkan data yang dibutuhkan oleh suatu organisasi.

2.3.1 MySQL

Menurut Anhar (2010: 21), MySQL (*My Structure Query Language*) adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS. Dapat disimpulkan MySQL adalah salah satu jenis *database server* yang termasuk jenis RDBMS (*Relational Database Management System*).

2.3.2 PHPMyAdmin

Menurut Nugroho (2013: 71), PHPMyAdmin adalah tools yang dapat digunakan dengan mudah untuk manajemen *database* MySQL secara visual dan *server* MySQL, sehingga kita tidak perlu menulis *query* SQL setiap akan melakukan operasi *database*.

2.4 Bahasa Pemrograman dan Markup

2.4.1 PHP (*Hypertext Preprocessor*)

Menurut Sidik (2012: 4), PHP merupakan secara umum dikenal dengan sebagai bahasa pemrograman *script-script* yang membuat dokumen HTML secara *on the fly* yang dieksekusi di server web, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML, dikenal juga sebagai bahasa pemrograman *server side*.

2.4.2 CSS (*Cascading Style Sheet*)

Abdulloh (2016: 2) menyatakan CSS atau *Cascading Style Sheets* merupakan skrip yang digunakan untuk mengatur desain *website*. HTML memang mempunyai kemampuan untuk mengatur tampilan *website*, namun kemampuannya sangat terbatas. Fungsi CSS adalah memberikan pengaturan yang lebih lengkap agar struktur *website* yang dibuat dengan HTML terlihat lebih rapi dan elegan.

2.4.3 Javascript

Marisa (2017: 42) menyatakan Javascript adalah bahasa yang berbentuk kumpulan skrip yang pada fungsinya berjalan pada suatu dokumen HTML. Bahasa pemrograman ini memberikan kemampuan tambahan terhadap HTML untuk menambah fungsionalitas dan kenyamanan halaman web, dengan mengizinkan pengekseskuan perintah-perintah di sisi *web browser* bukan di sisi *web server*.

2.5 Peralatan Pendukung Sistem (*Tools System*)

2.5.1 UML (*Unified Modelling Language*)

Mulyani (2016: 35) menjelaskan bahwa *Unified Modelling Language* atau selanjutnya disebut UML adalah sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem. UML pertama kali dipopulerkan oleh Grady Booch dan James Rumbaugh pada tahun 1994 untuk mengkombinasikan dua metodologi terkenal yaitu Booch dan OOMT, kemudian Ivar Jacobson, yang menciptakan *Object Oriented Software Engineering* (OOSE) ikut bergabung. Standard UML dikelola oleh *Object Management Grup* (OMG).

1. *Use Case* Diagram

Mulyani (2016: 245) mendefinisikan *use case* diagram yaitu diagram yang menggambarkan dan merepresentasikan aktor, *use cases*, dan *dependencies* suatu proyek dimana tujuan dari diagram ini adalah untuk menjelaskan konsep hubungan antara sistem dengan dunia luar.

2. Skenario

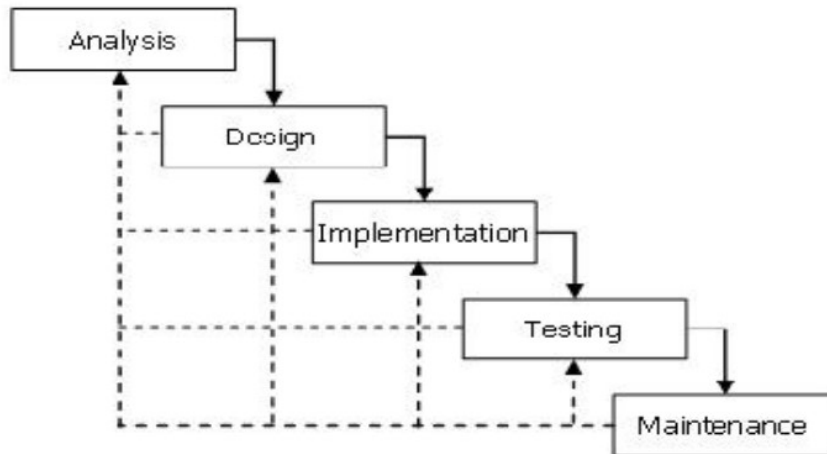
Menurut Yasin (2012:238), skenario adalah sebuah dokumentasi terhadap kebutuhan fungsional dari sebuah sistem. *Form* skenario merupakan penjelasan penulisan use case dari sudut pandang *actor*.

3. Activity Diagram

Mulyani (2016: 249) mendefinisikan *activity* diagram adalah diagram UML yang digunakan untuk menggambarkan alur aktivitas dari suatu proses. *Activity* diagram memungkinkan siapapun yang melakukan proses untuk memilih urutan dalam melakukannya, dengan kata lain diagram hanya menyebutkan aturan-aturan rangkaian dasar yang harus kita ikuti.

2.6 System Development Life Cycle (SDLC)

Untuk pengembangan sistem penelitian ini menggunakan model SDLC (Software Development Life Cycle). System Development Life Cycle (SDLC) adalah proses pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sebuah sistem. SDLC juga merupakan pola yang diambil untuk mengembangkan sistem perangkat lunak, yang terdiri dari tahap-tahap: rencana (planning), analisis (analysis), desain (design), implementasi (implementation), uji coba (testing) dan pengelolaan (maintenance). Model SDLC yang dipakai dalam penelitian ini adalah model Waterfall. Waterfall Model atau Classic Life Cycle merupakan model yang paling banyak dipakai dalam Software Engineering (SE). Menurut Bassil (2012) disebut waterfall karena tahap demi tahap yang harus dilalui menunggu selesainya tahap sebelumnya dan berjalan berurutan.



Gambar 2.1 Metode Waterfall

Tahapan-tahapan dari metode *waterfall* adalah sebagai berikut :

1. Tahap Analisis

Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi ini biasanya dapat diperoleh melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

2. Tahap Desain

Spesifikasi kebutuhan dari tahap sebelumnya akan dipelajari dalam fase ini dan desain sistem disiapkan. Desain Sistem membantu dalam menentukan perangkat keras (*hardware*) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

3. Tahap Implementasi

Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut *unit*, yang terintegrasi dalam tahap selanjutnya. Setiap *unit* dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai *unit testing*.

4. Tahap Pengujian

Seluruh *unit* yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem setelah pengujian yang dilakukan masing-masing *unit*. Setelah integrasi seluruh sistem diuji untuk mengecek setiap kegagalan maupun kesalahan.

5. Tahap Pemeliharaan

Tahap akhir dalam model *waterfall*. Perangkat lunak yang sudah jadi, dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi *unit* sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

2.7 Pengertian Sistem Pakar

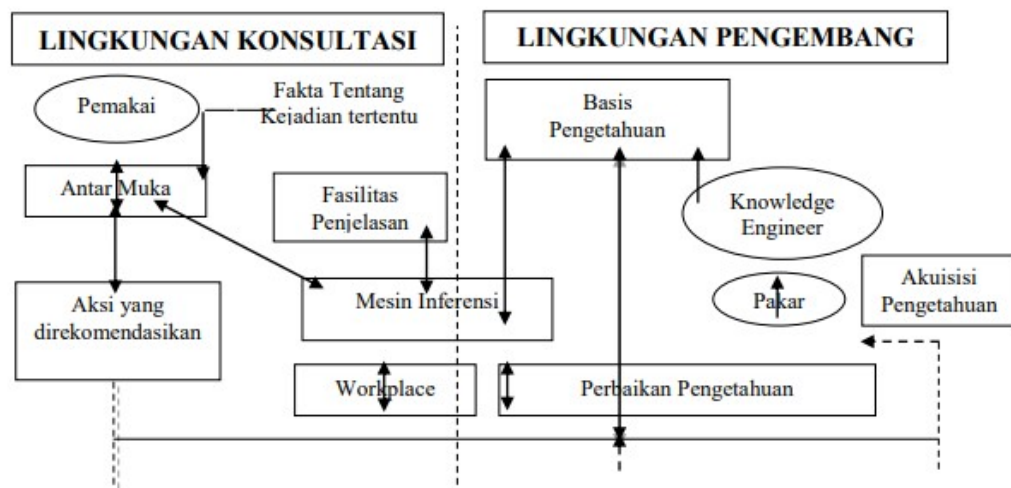
Sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta, dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tersebut. Istilah sistem pakar berasal dari knowledge-based expert system. Istilah ini muncul karena untuk memecahkan masalah, sistem pakar menggunakan pengetahuan seorang pakar yang dimasukkan ke dalam komputer. Seseorang yang bukan pakar menggunakan sistem pakar untuk meningkatkan kemampuan pemecahan masalah, sedangkan seorang pakar menggunakan sistem pakar untuk knowledge assistant.

Sistem pakar adalah suatu program komputer cerdas yang menggunakan knowledge (pengetahuan) dan prosedur inferensi untuk menyelesaikan masalah yang cukup sulit sehingga membutuhkan seorang yang ahli untuk menyelesaikannya. Pengetahuan adalah sebuah kekuatan yang dapat memecahkan suatu masalah yang

kita temui sehari-hari. Sistem pakar adalah program Artificial Intellenge yang menggabungkan pangkalan pengetahuan (knowledge base) dengan sistem inferensi. Kecerdasan buatan atau Artificial Intellenge (AI) dapat didefinisikan sebagai sub bidang pengetahuan komputer yang khusus ditujukan untuk membuat software dan hardware yang sepenuhnya biasa menirukan beberapa fungsi otak manusia. Karena itu diharapkan komputer bisa membantu manusia didalam berbagai masalah yang sangat rumit.

2.7.1 Struktur Sistem Pakar

Sistem pakar disusun oleh dua bagian utama, yaitu lingkungan pengembangan (development environment) dan lingkungan konsultasi (consultation environment). Lingkungan pengembangan sistem pakar digunakan untuk memasukan pengetahuan pakar ke dalam lingkungan sistem pakar.



Gambar 2.2 Struktur Sistem Pakar

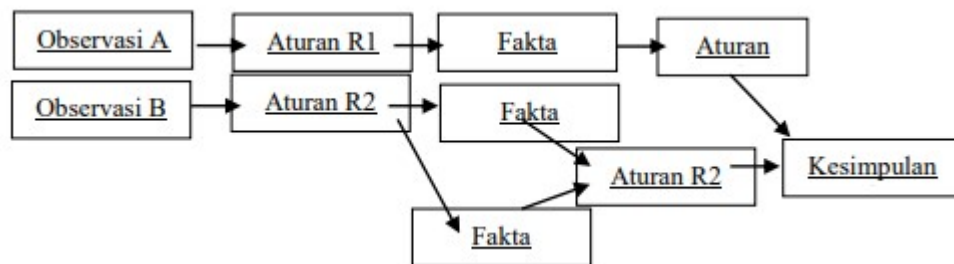
2.8 Forward Chaining

Forward Chaining adalah teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian IF dari rules IF-THEN. Bila ada fakta yang cocok dengan bagian IF, maka rule tersebut dieksekusi. Bila sebuah rule dieksekusi, maka sebuah fakta baru (bagian THEN) ditambahkan ke dalam database. Setiap kali pencocokan, dimulai dari rule teratas. Setiap rule hanya boleh dieksekusi sekali saja. Proses pencocokan berhenti bila tidak ada lagi rule yang bisa dieksekusi. Metode pencarian yang digunakan adalah Depth-First Search (DFS), Breadth-First Search (BFS) atau Best First Search. Pendekatan dalam pelacakan dimulai dari informasi masukan dan selanjutnya mencoba menggambarkan kesimpulan, pelacakan kedepan mencari fakta yang sesuai dengan bagian IF dari aturan IF-THEN. Dengan metode forward chaining dari pendekatan dan aturan yang telah dihasilkan dapat ditinjau oleh para ahli untuk diperbaiki atau dimodifikasi untuk memperoleh hasil yang lebih baik.

Metode Forward Chaining adalah metode pencarian atau teknik pelacakan kedepan yang dimulai dengan informasi yang ada dan penggabungan rule untuk menghasilkan suatu kesimpulan atau tujuan. Pelacakan maju ini sangat baik jika bekerja dengan permasalahan yang dimulai dengan rekaman informasi awal dan ingin dicapai penyelesaian akhir, karena seluruh proses akan dikerjakan secara berurutan maju. Berikut adalah diagram Forward Chaining secara umum untuk menghasilkan sebuah goal.

Forward Chaining berarti menggunakan himpunan aturan kondisi-aksi. Dalam metode ini, data digunakan untuk menentukan aturan mana yang akan dijalankan, kemudian aturan tersebut dijalankan. Mungkin proses menambahkan data ke memori kerja. Proses diulang sampai ditemukan suatu hasil. Metode inferensi runut maju

cocok digunakan untuk menangani masalah pengendalian (controlling) dan peramalan (prognosis) (Giarattano dan Riley, 1994).



Gambar 2.3 Forward Chaining

Untuk mempermudah pemahaman mengenai metode ini, akan diberikan ilustrasi kasus pembuatan sistem pakar dengan daftar aturannya sebagai berikut: R1:

Jika Premis 1 Dan Premis 2 Dan Premis 3 Maka Konklusi 1

R2: Jika Premis 1 Dan Premis 3 Dan Premis 4 Maka Konklusi 2

R3: Jika Premis 2 Dan Premis 3 Dan Premis 5 Maka Konklusi 3

R4: Jika Premis 1 Dan Premis 4 Dan Premis 5 Dan Premis 6 Maka Konklusi 4

