

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

Pada bahasan ini akan menjelaskan dan memberikan beberapa pengertian yang berhubungan dengan konsep dasar sistem berhubungan dengan judul penelitian yang digunakan dalam penyusunan tugas akhir ini. Uraian konsep dasar sistem yang digunakan antara lain:

1. Penjadwalan

Penjadwalan merupakan salah satu kegiatan yang penting dalam penentuan waktu dan urutan kegiatan. Dengan adanya penjadwalan maka perusahaan akan mendapatkan gambaran mengenai kegiatan yang dilaksanakan sehingga perusahaan akan dapat memperkirakan mengenai kebutuhan dan waktu penyelesaian (Arifin & Wasito, 2017). Dengan begitu perusahaan dapat menghindari sedini mungkin apabila selama proses produksi terjadi penyimpangan dan kesalahan kegiatan yang tidak sesuai rencana, sehingga dapat mengurangi resiko.

Morton (1993), juga mendefinisikan penjadwalan sebagai pengambilan keputusan tentang penyesuaian aktivitas dan sumber daya dalam rangka menyelesaikan sekumpulan pekerjaan agar tepat pada waktunya dan mempunyai kualitas seperti yang diinginkan.

Dari penjelasan di atas dapat di simpulkan bahwa Penjadwalan merupakan kegiatan pengalokasikan sumber daya yang dimiliki suatu organisasi untuk menjalankan kegiatan operasi dalam organisasi tersebut sehingga penjadwalan dapat diselesaikan tepat waktu dan sesuai dengan rencana.

2. Sistem

Sistem adalah seperangkat komponen yang saling berhubungan dan saling bekerja sama untuk mencapai beberapa tujuan. Selain itu sistem terdiri dari unsur-unsur dan masukan (*input*), pengolahan (*processing*), serta keluaran (*output*) (Agustin, 2018). Dapat di simpulkan bahwa Sistem merupakan sekumpulan prosedur yang saling berkaitan dan terhubung yang di maksudkan untuk mencapai suatu tujuan yang di inginkan oleh suatu organisasi. Menurut Wilkonson et al dalam (Mulyani, 2017), mengatakan ada beberapa ciri-ciri system, ciri-ciri sistem adalah sebagai berikut:

- a. Sistem mempunyai komponen-komponen.
- b. Komponen-komponen sistem harus terintegrasi (saling berhubungan).
- c. Sistem mempunyai batasan sistem.
- d. Sistem mempunyai tujuan yang jelas.
- e. Sistem mempunyai lingkungan.
- f. Sistem mempunyai input, proses dan output.

3. Perancangan Sistem

Menurut Henderi dalam (Sidik et al., 2018) “Perancangan Sistem adalah penggambaran, perencanaan dan pembentukan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesatuan yang utuh dan berfungsi”. Jadi Perancangan sistem menentukan bagaimana suatu sistem akan menyelesaikan apa yang mesti diselesaikan.

Dapat disimpulkan dari definisi di atas bahwa Perancangan Sistem merupakan Suatu sistem terdiri dari sekumpulan Elemen–elemen saling bekerja sama untuk mencapai tujuan Suatu elemen yang telah ada, dan merupakan bagian dari sistem lain yang lebih besar Jadi keberadaan suatu sistem, dibentuk untuk menangani, mengatur

serta mengkoordinasikan suatu kegiatan yang rutin terjadi, Perancangan sistem menentukan bagaimana suatu sistem akan menyelesaikan apa yang mesti diselesaikan.

4. *Website*

Menurut Abdulloh dalam (Sidik et al., 2018) “Website atau web adalah sekumpulan halaman yang berisi informasi dalam bentuk data digital baik berupa text, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet”. Menurut Hariyanto (2015) dalam (Destiningrum & Adrian, 2017), *Website* adalah kumpulan halaman yang menampilkan informasi data teks, data gambar, data animasi, suara, video dan gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*)”.

Berdasarkan uraian di atas dapat disimpulkan bahwa *Web* adalah Sebuah software yang berfungsi untuk menampilkan dokumen-dokumen pada suatu web yang membuat pengguna mudah dalam mengakses internet melalui *software* yang terkoneksi dengan internet.

5. *Framework Codeigniter*

Menurut Betha Sidik (2012) *Framework* adalah :“ kumpulan intruksi-intruksi yang dikumpulkan dalam class dan function-function dengan fungsi masing-masing untuk memudahkan developer dalam memanggilnya tanpa harus menuliskan syntax program yang sama berulang-ulang serta dapat menghemat waktu”. Sedangkan Menurut Betha Sidik (2012) *CodeIgniter* adalah :“ Sebuah *framework php* yang bersifat *open source* dan menggunakan metode MVC (*Model, View, Controller*) untuk memudahkan developer atau programmer dalam membangun sebuah aplikasi berbasis web tanpa harus membuatnya dari awal”.

Dalam uraian di atas dapat disimpulkan bahwa *Framework Codeigniter* adalah sebagai kumpulan dari fungsi-fungsi atau prosedur-prosedur dan class-class untuk tujuan tertentu untuk mempermudah dan mempercepat pekerjaan seorang programmer, tanpa harus membuat fungsi atau class dari awal. Kelebihan dari *framework codeigniter* jika dibandingkan dengan *framework* lain adalah sebagai berikut:

- a. Kerangka kerja *Codeigniter* memiliki lisensi dibawah *Apache/BSD open-source* sehingga bersifat bebas atau gratis.
- b. Ukuran yang kecil merupakan keunggulan tersendiri jika dibandingkan *framework* lain yang berukuran besar.
- c. Menggunakan konsep M-V-C *Codeigniter* merupakan konsep M-V-C (*ModelView-Controller*) yang memungkinkan pemisahan antara *layer application-logic* dan *presentation*.

6. **MySQL (*My Structured Query Language*)**

Menurut Adi Nugroho (2011) dalam (Destiningrum & Adrian, 2017), MySQL (*My Structured Query Language*) adalah “ Suatu sistem basis data relation atau *Relational Database managemnt System (RDBMS)* yang mampu bekerja secara cepat dan mudah digunakan. MySQL juga juga dapat di artikan sebagai program pengakses database yang bersifat jaringan, sehingga dapat digunakan untuk aplikasi multi user yaitu banyak pengguna. Namun setiap program tidak bisa di jadikan produk turunan atau komersial.

7. **PHP (*Hipertext Preprocessor*)**

Menurut Saputra (2011) dalam (Abdurahman, 2018), PHP atau yang memiliki kepanjangan *Hypertext Propocessor* merupakan suatu bahasa pemograman yang difungsikan untuk membangun suatu website dinamis. PHP berjalan pada sisi server sehingga PHP disebut juga sebagai bahasa *Server Side Scripting*. Pada prinsipnya

menurut Kadir (2008), PHP mempunyai fungsi yang sama dengan skrip-skrip seperti ASP (*Active Server Page*), *Cold Fusion*, ataupun Perl. Namun, perlu diketahui bahwa PHP sebenarnya bisa dipakai secara *command line*. Artinya, skrip PHP dapat dijalankan tanpa melibatkan *web server* maupun browser.

Dari pemaparan di atas dapat disimpulkan bahwa PHP merupakan salah satu server yang di rancang untuk Aplikasi Web yang di sisipkan diantara bahasa HTML, dalam penggunaannya bahasa pemrograman yang berguna untuk membuat web dinamis. PHP termasuk *Open Source Product*. Jadi, dapat diubah *source code* dan mendistribusikanya secara bebas.

8. Model Pengembangan Perangkat Lunak

Model pengembangan perangkat lunak yang digunakan dalam penelitian ini adalah *prototype*. *Prototype* adalah “suatu proses yang memungkinkan *developer* membuat sebuah model *software*”. *Prototype* berfungsi sebagai mekanisme untuk mengidentifikasi kebutuhan-kebutuhan *software* dan dijadikan sebagai sampel awal guna menguji sebuah proses atau konsep yang dibutuhkan.

2.2 Teori Pendukung

1. *Entity Relationship Diagram (ERD)*

Menurut Simarmata dalam (Hartini & Dermawan, 2017) “ERD adalah alat untuk pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek ke dalam entitas- entitas dan menentukan hubungan antar entitas”.

ERD merupakan suatu model untuk menjelaskan hubungan antara data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD atau *Entity Relationship Diagram* digunakan untuk mengidentifikasi data yang

akan diambil, disimpan dan dipanggil kembali untuk keperluan tertentu dalam mendukung kegiatan yang dilakukan oleh suatu organisasi.

Terdapat Beberapa komponen ERD yang sering digunakan dalam menggambarkan alur dari suatu sistem:

a. *Entty* (Entitas)

Notasi untuk mewakili suatu objek dengan karakteristik sama yang dilengkapi oleh atribut, sehingga pada suatu lingkungan nyata setiap objek akan berbeda dengan objek lainnya. Pada umumnya, objek dapat berupa benda, pekerjaan, tempat dan orang.

b. *Relation* (Relasi)

Notasi yang digunakan untuk menghubungkan beberapa entitas berdasarkan fakta pada suatu lingkungan.

c. *Attribute* (Atribut)

Notasi yang menjelaskan karakteristik suatu entitas dan juga relasinya. Atribut dapat sebagai *key* yang bersifat unik yaitu *Primary Key* atau *Foreign Key*.

2. ***Logical Relcord Structure (LRS)***

Logical Relcord Structure (LRS) Menurut Kusrini dalam (Wahid et al., 2016)

“*Logical Record Structure* merupakan representasi dari struktur record-record pada table-tabel yang terbentuk dari hasil relasi antar himpunan entitas pada diagram E-R.”

Menurut Hasugian memberikan batasan bahwa LRS adalah sebuah model system yang digambarkan dengan sebuah diagram E-R akan mengikuti pola atau aturan permodelan tertentu yaitu mengikuti aturan-aturan menurut Hasugian dan Shidiq (2012) dalam (Imam & Nugraha, 2018) adalah berikut ini:

a. Setiap entitas akan diubahs kebentuk kotak.

- b. Sebuah atribut relasi disatukan dalam sebuah kotak bersama entitas jika hubungan yang terjadi pada diagram ER 1:M (relasi bersatu dengan *cardinality* M) atau tingkat hubungan 1:1 (relasi bersatu dengan *cardinality* yang paling membutuhkan referensi).
- c. Sebuah relasi dipisah dalam sebuah kotak tersendiri (menjadi entitas baru) jika tingkat hubungannya M:M (many to many) dan memiliki *foreign key* sebagai *primary key* yang diambil dari kedua entitas yang sebelumnya saling berhubungan.

3. *Unified Modelling Language (UML)*

UML singkatan dari *Unified Modelling Language* yang berarti bahasa permodelan standar. Menurut Munawar dalam (Ashari & Firmansyah, 2017) UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

UML diaplikasikan untuk maksud tertentu, yaitu:

- a. Merancang perangkat lunak.
- b. Sarana Komunikasi antara perangkat lunak dengan proses bisnis.
- c. Menjabarkan system secara rinci untuk analisis dan mencari apa yang diperlukan sistem.
- d. Mendokumentasi system yang ada, proses-proses dan organisasinya.

Blok pembangunan UML adalah diagram. Beberapa diagram ada yang rinci (jenis timing diagram) dan lainnya ada yang bersifat umum (misalnya diagram kelas).

Para pengembang system berorientasi objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasikan system yang mereka rancang.

Menurut (Abdillah et al., 2019) ada jenis-jenis diagram dalam UML yakni:

a. *Use Case Diagram*

Use Case diagram digunakan untuk mengkomunikasikan interaksi manusia (*actor*) dengan apa yang bisa dilakukan oleh sistem. Sebuah *Use Case* dapat mewakili beberapa jalur interaksi manusia dengan sistem dan setiap jalur disebut sebagai skenario.

b. *Class Diagram*

Class Diagram adalah model statis yang mendukung tampilan data dan informasi dari keseluruhan sistem. Penggunaan *Class Diagram* dikaitkan dengan struktur basis data sistem atau dapat menggantikan ERD pada proses penggambaran diagram rekayasa perangkat lunak yang konvensional.

c. *Sequence Diagram*

Sequence diagram mengilustrasikan objek-objek yang terdapat pada *Use Case* dan menggambarkan arus pesan antara satu sama lain pada Objek *Use Case*. *Sequence Diagram* bersifat dinamis dan lebih banyak menampilkan aktifitas objek berdasarkan urutan waktu.

d. *Activity Diagram*

Activity Diagram adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dalam suatu system. Diagram ini terutama penting dalam suatu system serta permodelan fungsi-fungsi suatu system dan memberi tekanan pada aliran kendali antar objek. Diagram aktivitas dari sebuah system atau proses bisnis atau menu yang ada.