

## BAB II

### TINJAUAN PUSTAKA DAN LANDASAN TEORI

#### 2.1 Tinjauan Pustaka

Penelitian sebelumnya dilakukan oleh Firdaus Sinta Maulina dan Ari Kurniawan dengan judul “Pengembangan Aplikasi Radio Streaming Dan on-Demand Berbasis Website Di Radio Suara Bangkalan Fm”. Aplikasi ini dibuat untuk menutup kekurangan dari radio konvensional dimana gelombang FM tidak dapat menembus perbukitan maupun menghadapi cuaca ekstrim. Aplikasi ini juga memuat informasi mengenai berita Bangkalan. Metode yang digunakan dalam penelitian ini adalah waterfall dengan menggunakan framework Laravel untuk membangun website ini serta database MySQL [1].

Penelitian lain yang dilakukan oleh Ibnu Bukhari, Moh. Iwan Wahyuddin, Novi Dian Nathasia dengan judul “Perancangan Sistem Informasi Radio Online Universitas Nasional Berbasis Web”. Tujuan dilakukannya penelitian ini adalah membangun dan mengembangkan radio *online* Universitas Nasional dengan *Icecast* sebagai *freeware streaming* radio dan winamp sebagai perangkat lunak yang digunakan untuk media penyiaran radio. Perancangan sistem informasi ini melalui beberapa tahapan diantaranya adalah analisis kebutuhan, perancangan sistem, pengujian dan implementasi [2].

Penelitian yang masih sejenis yang dilakukan oleh Ahmad Salafuddin, De Rosal Ignatius Moses Setiadi, Desi Purwanti Kusumaningrum, , Eko Hari Rachmawanto dan Christy Atika Sari dengan judul “Pengembangan Website Radio SSFM Semarang dengan Codeigniter Berbasis Model View Controller”. Penelitian ini bertujuan untuk perancangan ulang untuk membuat perubahan yang signifikan pada website, sehingga penyiaran juga dapat dilakukan dengan sistem berbasis web yang modern, kreatif dan inovatif. *Framework codeigniter* dan *metode Model View Controller* (MVC) yang digunakan untuk mengimplementasikan website SSFM Semarang [3].

Penelitian lain yang dilakukan oleh Arya Andrian ZR dan Asyifah Nur Istyanti dengan judul “Rancang Bangun Sistem Pendataan Request Lagu dari Program Acara di RRI Medan Berbasis Website” bertujuan untuk memudahkan admin dan staff untuk mendata acara program radio yang terdapat polling *request* lagu. Dalam pengembangan sistem

menggunakan metode waterfall yaitu pengembangan sistem yang setiap perkembangannya harus melalui susunan yang teratur. Sistem ini menggunakan UML (*Unified Modelling Language*) [4].

Penelitian sebelumnya juga dilakukan oleh Jumirah, Jannatun Aliyah dan Jizan Qifli Ilhamdi yang berjudul “Perancangan Sistem Informasi Radio Streaming Suara Sabalong Samalewa Berbasis Web Pada Dinas Komunikasi Informatika Dan Statistik Kabupaten Sumbawa”. Tujuan dari penelitian ini adalah memberikan sarana media informasi, edukasi, dan hiburan dalam bentuk website dengan konten radio streaming. Penelitian ini dilakukan dengan metode waterfall dengan melakukan observasi langsung ke lapangan [5].

Muhammad Darussalam, Mochamad Nandi Susila dan Maruloh melalui penelitian yang berjudul “Aplikasi Radio Online pada PT. Radio Swara Mersidiona 93.9 FM” membahas mengenai perancangan aplikasi radio online pada Mersi 93.9 FM, sehingga memiliki tujuan agar dapat membantu optimalisasi aktifitas kegiatan penyiaran yang dilakukan karena sudah berbasis digital aplikasi yang bisa diakses melalui perangkat yang terkoneksi internet. Metode pengembangan aplikasi menggunakan waterfall. Aplikasi ini menggunakan UML dan framework codeigniter. Tahapan pengujian aplikasi dilakukan dengan menggunakan Teknik pengujian black box testing [6].

Penelitian kali ini, bermaksud untuk merancang dan membangun sebuah sistem informasi manajemen siaran radio pada radio Thomson Cilacap berbasis website. Sistem ini dapat memudahkan masyarakat dalam mengakses radio dan melakukan *streaming* radio secara online. Pada sistem ini juga terdapat fitur pendukung dimana pendengar dapat melakukan *request* lagu dan kirim pesan sehingga nantinya akan memudahkan admin dalam mendata dan menyiarkan hasil *request* tersebut. Metode pengembangan sistem yang dilakukan pada sistem ini adalah prototipe.

## 2.2 Landasan Teori

### 2.2.1 Radio

Menurut *Kamus Besar Bahasa Indonesia (KBBI)*, Radio adalah siaran (pengiriman) suara atau bunyi melalui udara. Sedangkan pengertian radio menurut ensiklopedi Indonesia, yaitu penyampaian informasi dengan pemanfaatan gelombang elektromagnetik bebas yang memiliki frekuensi kurang dari 300 GHz (Panjang gelombang lebih dari 1 mm) [7].

### 2.2.2 Media Streaming

Media *streaming* adalah suatu system yang terbentuk dari komponen-komponen yang saling mendukung satu dengan lainnya. Media ini merupakan teknologi yang dapat memungkinkan proses pengiriman data berupa *audio*, *video*, dan *multimedia* secara berkelanjutan (*stream*). Data tersebut kemudia akan dikirim dari sebuah server aplikasi yang selanjutnya diterima serta ditampilkan secara *real-time* oleh aplikasi pada komputer client melalui internet.[8]

### 2.2.3 On-Demand

*On-demand* merupakan salah satu cara pengiriman data multimedia secara langsung, dimana *client* melakukan permintaan dan langsung dilayani oleh *server*. Pengiriman data dilakukan bagian demi bagian secara terus menerus, sedangkan penerimaan data tergantung dengan penerimaan user. Pada *on-demand* hanya terdapat tombol *play* dan *pause* saja. Saat user menekan tombol *pause*, maka penerimaan data akan terhenti, dan data akan hilang sementara hingga *user* menekan tombol *play* kembali.[9]

*On-demand* akan memungkinkan para *user* dapat mengakses acara yang diinginkan sesuai program acara atau tanggal tayang yang ditentukan. Dengan adanya *on-demand* juga dapat memuaskan para pendengar apabila ingin mengulang kembali acara yang sudah terlewat.[1]

### 2.2.4 Sistem Informasi

Sistem informasi adalah sekumpulan komponen pembentuk sistem yang mempunyai keterkaitan antara satu komponen dengan komponen lainnya yang bertujuan untuk menghasilkan suatu informasi dalam suatu bidang tertentu. Dalam sistem informasi dibutuhkan klasifikasi alur informasi. Hal itu disebabkan keanekaragaman kebutuhan akan suatu informasi oleh pengguna di lapangan. Kriteria informasi diantaranya fleksibel, efektif, dan efisien [10].

Menurut O'Brien, Sistem informasi adalah suatu kumpulan dari komponen-komponen dalam suatu organisasi yang berhubungan dengan proses penciptaan dan aliran informasi. Secara umum sistem informasi merupakan kombinasi dari orang (*people*), perangkat keras (*hardware*), perangkat lunak (*software*), jaringan komunikasi (*communications networks*) dan sumber data yang dihimpun, ditransformasi dan mengalami proses pengaliran dalam suatu organisasi [10].

### 2.2.5 Website

Abdullah berpendapat bahwa *website* adalah kumpulan-kumpulan halaman yang terdiri dari beberapa laman yang di dalamnya berisi informasi dalam bentuk data digital baik itu berupa gambar, *video*, *audio*, teks dan animasi lainnya yang disediakan melalui jalur koneksi internet [11]. Pengertian lain dari *website* adalah fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada *website* disebut dengan *web page* dan *link* dalam *website* memungkinkan pengguna bisa berpindah dari satu *page* ke *page* lainnya (*hyper text*), baik diantara *page* yang disimpan dalam *server* yang sama maupun *server* diseluruh dunia. Halaman diakses dan dibaca melalui browser seperti *Internet Explorer*, *Mozilla Firefox*, *Google Chrome* dan aplikasi browser lainnya.[1]

### 2.2.6 Hipertext Preprocessor (PHP)

*Hipertext Preprocessor* merupakan Bahasa pemrograman untuk membuat *website* yang bersifat server-side scripting. PHP bersifat dinamis. *PHP* dapat dijalankan pada berbagai macam sistem operasi seperti *Windows*, *Linux*, dan *Mac Os*. Selain *Apache*, *PHP* juga mendukung beberapa web server lain seperti *Microsoft ISS*, *Caudium*, dan *PWS*. *PHP* dapat memanfaatkan database untuk menghasilkan halaman web yang dinamis. Sistem manajemen database yang sering digunakan Bersama PHP adalah MySQL[12]

### 2.2.7 Database

Database atau basisdata merupakan perkumpulan data yang disimpan dalam komputer untuk dapat dengan mudah diakses, diupdate dan diorganisir menggunakan *Database Manangement System (DBMS)*. Database mempunyai berbagai macam model, diantaranya yang paling populer adalah relational data model. Relational data model memungkinkan data untuk disimpan dalam sebuah tabel ataupun beberapa tabel sebagai *value* yang mempunyai relasi antara satu sama lain[13].

Penggunaan database dapat memudahkan proses identifikasi data, database akan menampilkan data yang diminta beserta dengan data-data yang bersangkutan melalui *database management system*. Dengan database, penyimpanan data dengan skala besar dapat dipermudahkannya menggunakan fasilitas pada sistem manajemen database dan menghindari duplikasi data[13].

SQL Server adalah salah satu sistem manajemen database berjenis *relational database management system (RDBMS)* yang dikembangkan oleh perusahaan Microsoft. Sistem manajemen ini dibekali dengan beberapa fitur yang dapat membantu dalam pengambilan keputusan seperti *Business Intelligence* dan *Data Analysis*. SQL Server mempunyai ekstensi dari sql yang diimplementasikan oleh Microsoft sendiri Bernama Transact-SQL[13].

*MySQL* adalah sebuah sistem manajemen database yang bersifat *open source* dan relational. Artinya, data yang dikelola dalam database yang akan diletakan pada beberapa tabel yang terpisah sehingga manipulasi data akan jauh lebih cepat. *MySQL* dapat digunakan untuk mengelola database mulai dari yang kecil sampai dengan yang sangat besar[12]. *MySQL* mempunyai stabilitas dan kecepatan akses yang tinggi, dapat berjalan diberbagai sistem operasi, mudah digunakan dan tersedia diberbagai macam bahasa. Konektivitas, kecepatan dan keamanannya telah menjadikan *MySQL* sebagai server yang cocok untuk mengakses *internet*. *MySQL* mempunyai beberapa keunggulan antara lain:

1. Portabilitas  
*MySQL* dapat berjalan stabil pada berbagai sistem operasi seperti *Windows, Linux, Mac Os*, dan masih banyak lagi.
2. Perangkat Lunak *Open Source*  
*MySQL* didistribusikan sebagai perangkat lunak *open source* atau sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
3. Multi User  
*MySQL* dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. *Performance Turning*  
*MySQL* memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL persatuan waktu.
5. Ragam Tipe Data  
*MySQL* memiliki ragam tipe data yang sangat kaya seperti, *signed, unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.

Dalam penggunaan SQL terdapat beberapa perintah yang bertujuan untuk mengakses dan memanajemen data yang terdapat dalam database. Secara garis besar, SQL Server terdapat 2 jenis perintah *query* yaitu:

a. *Data Definition Language (DDL)*

*Data Definition Language (DDL)* adalah merupakan kumpulan perintah SQL yang digunakan untuk membuat, mengubah dan menghapus struktur dan definisi metadata dari objek-objek database. Berikut adalah contoh perintah DDL yang digunakan pada MySQL[14]:

- CREATE, digunakan untuk membuat termasuk diantaranya *database* dan tabel baru.
- ALTER digunakan untuk mengubah struktur tabel yang telah dibuat.
- DROP digunakan untuk menghapus *database* dan tabel.

b. *Data Manipulation Language (DML)*

*Data Manipulation Language (DML)* merupakan kumpulan perintah SQL yang digunakan untuk proses pengolahan isi data di dalam *table* seperti memasukkan, merubah dan menghapus isi data dan tidak terkait dengan perubahan struktur dan definisi tipe data dari objek database. Berikut adalah contoh perintah DML yang digunakan pada MySQL[14]:

- INSERT, digunakan untuk menyisipkan atau memasukan data dalam tabel.
- UPDATE, digunakan untuk memperbaharui data lama menjadi data terkini.
- SELECT digunakan untuk mengambil data atau menampilkan data dari satu tabel atau beberapa tabel.

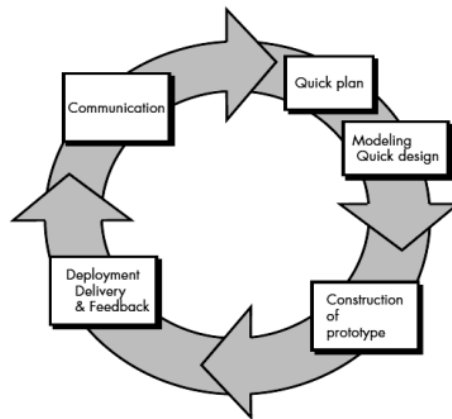
### 2.2.8 XAMPP

*XAMPP* adalah perangkat lunak bebas, yang mendukung banyak sistem operasi yang merupakan kompilasi dari beberapa program. Fungsi dari *XAMPP* yaitu sebagai server yang berdiri sendiri (*localhost*) yang terdiri dari program *Apache*, *HTTP Server*, *MySQL* database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. *XAMPP* sendiri merupakan singkatan dari X (empat sistem operasi apapun seperti *Windows*, *Linux*, *Mac Os* dan juga *Solaris*), *Apache*, *MySQL*, *PHP* dan *Perl*[15]

### 2.2.9 Prototype

Metode *prototype* merupakan salah satu metode pengembangan perangkat lunak, berupa model fisik cara kerja sistem dan berfungsi

sebagai versi awal dari sistem. Metode *prototype* biasanya dimulai dengan mengidentifikasi dan mengumpulkan kebutuhan pengguna terhadap perangkat lunak yang akan dibuat [16] Selanjutnya dilakukan perancangan cepat yang berfokus untuk mempresentasikan aspek-aspek yang ditampilkan untuk pengguna. Perancangan cepat ini menghasilkan *prototype* yang akan diuji oleh pengguna. Hasil dari pengujian akan digunakan untuk memperbaiki *prototype* dan akan dilakukan iterasi secara terus menerus hingga pengguna merasa puas dengan *prototype* yang dihasilkan[17]. Tahapan dalam metode *prototype* dapat dilihat pada **Gambar 2.1**.



**Gambar 2. 1** Tahap Metode Prototype

Keterangan :

- a. *Communication* merupakan proses berbagi informasi tentang desain, dan potensi penggunaannya dalam perancang dan pengguna. Pada tahap ini dilakukan komunikasi dengan *user*. *User* akan menceritakan fitur-fitur yang diperlukan dalam sistem ini serta ikut serta dalam memberikan evaluasi dan masukan terhadap sistem yang sedang dikembangkan[18].
- b. *Quick Plan and Quick Design*, merupakan ciri dari metode *prototyping*. Setelah mendapatkan data kebutuhan dari user, pengembang akan melakukan perencanaan dan desain untuk sistem yang sedang berjalan[18].
- c. *Construction of Prototype*, yaitu tahapan ini dilakukan pembangunan *prototype*. Pengembang akan membuat *prototype*.

Desain *prototype* ini yang akan digunakan oleh user untuk memahami serta mengevaluasi apakah kebutuhan-kebutuhan yang diharapkan sudah terpenuhi[18].

- d. *Deployment Delivery & Feedback*, tahapan terkakhir ini menyampaikan *prototype* ke pihak *user* untuk mendapatkan masukan. *Prototype* akan memberikan gambaran secara visual kepada user atas sistem informasi yang akan dibangun. Dengan *prototype* ini, selain dari sisi fungsional, user juga dapat memberikan masukan dari sisi desain interface[18].

### 2.2.10 *Black Box Testing*

*Black box testing* merupakan teknik pengujian *software* yang fokus pada spesifikasi fungsi-fungsi yang ada pada sistem yang dikembangkan. *Black box testing* cenderung dapat menemukan beberapa hal seperti fungsional yang tidak benar atau tidak ada, kesalahan struktur data, kesalahan akses basis data, kesalahan antar muka, kesalahan *performance* serta kesalahan inisialisasi dan terminasi[19]. Keuntungan menggunakan metode *black box testing* adalah:

1. Penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu
2. Pengujian dilakukan dari sudut pandang pengguna, hal ini membantu mengungkapkan ambiguitas atau inkonsistensi dalam spesifikasi persyaratan.
3. *Programmer* dan *tester* keduanya saling bergantung satu sama lainnya[19].

### 2.2.11 Object Oriented Programming (OOP)

Pemrograman berorientasi objek atau *object-oriented programming* merupakan suatu pendekatan pemrograman yang menggunakan *object* dan *class*. Pada saat ini konsep OOP sudah semakin berkembang. Hampir semua programmer maupun pengembang aplikasi menerapkan konsep OOP. OOP bukanlah sekedar cara penulisan sintaks program yang berbeda, namun lebih dari itu, OOP merupakan pandangan dalam menganalisa sistem dan permasalahan pemrograman. Dalam OOP, setiap bagian dari program adalah *object*. Sebuah *object* mewakili suatu bagian program yang akan diselesaikan[20]. Beberapa konsep OOP dasar, antara lain:



a. *Encapsulation*

*Encapsulation* adalah mekanisme pemrograman yang membungkus kode dan data yang dimanipulasi dan menjaganya supaya terhindar dari penggunaan yang tidak perlu. Salah satu caranya dengan membentuk *object*. Sedangkan manfaat *encapsulation* adalah menyembunyikan implementasi *method* untuk mengakses data membuat kode lebih terpelihara[21].

b. *Inheritance (Penurunan sifat)*

*Inheritance* disebut juga dengan turunan. Prinsipnya adalah sebuah *class* dapat diturunkan dari *class* lain. *Class* yang menurunkan ke *class* lain disebut *superclass*, *parent class*, atau *base class* atau kelas induk, sedangkan *class* yang merupakan turunan disebut *subclass*[21].

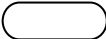
c. *Polymorphisme*

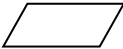
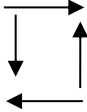
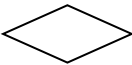





*Polymorphisme*, artinya mempunyai banyak bentuk. Dua objek atau lebih dikatakan sebagai *polymorphic*, bila *object-object* itu mempunyai antar muka yang identic namun mempunyai perilaku-perilaku yang berbeda[21].




### 2.2.12 Flowchart

Menurut Indrajani, flowchart (bagan alir) adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program. Biasanya mempengaruhi penyelesaian masalah yang khususnya perlu dipelajari dan dievaluasi lebih lanjut [22]. Flowchart adalah bagan yang menunjukkan alir di dalam program atau prosedur suatu sistem secara logika. Bagan alir digunakan untuk alat bantu komunikasi dan untuk dokumentasi. Berikut ini dapat dilihat simbol-simbol yang ada pada flowchart pada **Tabel 2.1**.

**Tabel 2. 1** Simbol Flowchart

| No | Simbol  | Nama            | Keterangan                           |
|----|---|-----------------|--------------------------------------|
| 1. |  | <i>Terminal</i> | Memulai dan mengakhiri suatu program |

| No | Simbol  | Nama                    | Keterangan   |
|----|---|-------------------------|--|
| 2. |    | <i>Input/Output</i>     | Memasukan data maupun menunjukkan hasil dari suatu process tanpa tergantung dengan jenis peralatannya  |
| 3. |    | <i>Flow</i>             | Menghubungkan antara simbol satu dengan simbol yang lain atau menyatakan jalannya arus dalam suatu proses. Simbol arus ini sering disebut juga dengan connecting line. |
| 4. |    | <i>Decision</i>         | Memilih proses berdasarkan kondisi yang ada.   |
| 5. |    | <i>Document</i>         | Merupakan simbol untuk data yang terbentuk informasi.  |
| 6. |    | <i>Manual Operation</i> | Menunjukkan pengolahan yang tidak dilakukan oleh komputer/pc.  |
| 7. |   | <i>Punched Card</i>     | Menyatakan input berasal dari kartu atau output ditulis ke kartu.  |
| 8. |  | <i>Display</i>          | Simbol yang menyatakan penggunaan peralatan output, misalnya layar, monitor, printe, dan lainnya.  |
| 9. |  | <i>Connector</i>        | Menyatakan sambungan dari suatu proses ke proses lainnya dalam halaman / lembar yang sama.   |

| No  | Simbol  | Nama                      | Keterangan  |
|-----|---|---------------------------|---|
| 10. |  | <i>Offline Connector</i>  | Menyatakan sambungan dari satu proses ke proses lainnya dalam halaman atau lembar yang berbeda. |
| 11. |  | <i>Predefined Process</i> | Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.             |
| 12. |  | <i>Manual Input</i>       | Simbol yang memasukan data secara manual dengan <i>online keyboard</i>                          |

### 2.2.13 Unified Modeling Language (UML)







Menurut Nugroho, UML (*Unified Modelling Language*) adalah Bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’ [23]. Pemodelan (*modelling*) sesungguhnya digunakan untuk menyederhanakan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. Pada UML terdapat 14 macam diagram diantaranya adalah *Activity Diagram, Use Case Diagram, Class Diagram, Sequence Diagram, Object Diagram, Component Diagram, Timing Diagram, Package Diagram, Class Diagram, Composite Diagram, Composite Structure Diagram, State Machine Diagram, Deployment Diagram, Communication Diagram*. Beberapa diagram UML yang akan digunakan dalam membuat Sistem Informasi Manajemen Siaran Pada Radio Thomson Cilacap Berbasis Website antara lain:

#### 1. Use Case Diagram

*Use case diagram* adalah pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat [24]. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Secara kasar *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Terdapat dua hal utama pada use

case yaitu mendefinisikan *actor* dan *usecase*. Berikut ini dapat dilihat simbol yang ada pada diagram usecase pada **Tabel 2.2**.

**Tabel 2. 2** Simbol Use Case



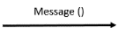



| No. | Simbol  | Nama               | Keterangan  |
|-----|---|--------------------|---|
| 1.  |    | <i>Use case</i>    | Deskripsi dari urutan aksi – aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .  |
| 2.  |    | <i>Actor</i>       | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .                             |
| 3.  |    | <i>Association</i> | Apa yang menghubungkan antara objek satu dengan objek lainnya.  |
| 4.  |    | <i>Include</i>     | Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.  |
| 5.  |   | <i>Extend</i>      | Menspesifikasikan bahwa <i>use case target</i> memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan. |
| 6.  |  | <i>System</i>      | Menspesifikasikan paket yang menampilkan sistem secara terbatas   |

## 2. Sequence Diagram

*Sequence diagram* merupakan UML yang menggambarkan interaksi antar objek di dalam dan disekitar sistem, termasuk pengguna, *display*, dan sebagainya berupa *message* yang digambarkan terhadap

waktu [24]. Berikut adalah simbol–simbol yang ada pada diagram *Sequence diagram*, dapat dilihat pada **Tabel 2.3**.

**Tabel 2. 3** Simbol Sequence Diagram

| No. | Simbol  | Nama                 | Keterangan   |
|-----|---|----------------------|--|
| 1.  |    | <i>LifeLine</i>      | Objek <i>entity</i> , antarmuka yang saling berinteraksi.  |
| 2.  |    | <i>Actor</i>         | Menggambarkan <i>user</i> atau pengguna.   |
| 3.  |    | <i>Message</i>       | Spesifikasi dari komunikasi antar <i>objek</i> yang memuat informasi – informasi tentang aktivitas yang terjadi. |
| 4.  |    | <i>Boundary</i>      | Menggambarkan sebuah <i>form</i> .   |
| 5.  |   | <i>Control Class</i> | Menghubungkan <i>boundary</i> dengan Tabel.  |
| 6.  |  | <i>Entity Class</i>  | Menggambarkan hubungan kegiatan yang akan dilakukan.   |

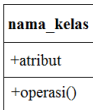



### 3. Class Diagram



*Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas. Sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu

kelas.[25] Diagram kelas dibuat agar *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.

*Class diagram* membantu dalam visualisasi struktur kelas – kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak. *Class diagram* memperlihatkan hubungan antar kelas dan penjelasan detail tiap – tiap kelas di dalam model desain (dalam logical view) dari suatu sistem. Selama proses analisis, *class diagram* memperlihatkan aturan – aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama proses analisis, *class diagram* memperlihatkan aturan – aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama tahap decian, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. Berikut dapat dilihat beberapa simbol dari *class diagram* pada **Tabel 2.4**.

**Tabel 2. 4** Simbol Class Diagram

| No. | Simbol  | Nama   | Keterangan   |
|-----|---|--|--|
| 1.  |    | Kelas  | Kelas pada struktur sistem.  |
| 2.  |    | Asosiasi/<br><i>Association</i>                  | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> . |
| 3.  |  | Asosiasi berarah/<br><i>Directed association</i> | Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain.                     |
| 4.  |  | Generalisasi                                     | Relasi antarkelas dengan makna generalisasi-   |

|    |   |                                    |  |
|----|---|------------------------------------|--|
|    |   |                                    | spesialisasi (umum khusus).  |
| 5. |  | Kebergantungan / <i>Dependency</i> | Relasi antarkelas dengan makna kebergantungan antarkelas.          |
| 6. |  | Agregasi / <i>Aggregation</i>      | Relasi antarkelas dengan makna semua-bagian ( <i>whole-part</i> ). |

### 2.2.14 Entity Relationship Diagram (ERD)

*Entity Relationship Diagram* (ERD) adalah diagram berbentuk notasi grafis yang berada dalam pembuatan database yang menghubungkan antara data satu dengan yang lain. Fungsi ERD adalah sebagai alat bantu dalam pembuatan database dan memberikan gambaran bagaimana kerja database yang akan dibuat. Di dalam ERD terdapat 3 elemen dasar, yaitu entitas, atribut, dan relasi[26].

#### 1. Entitas

Entitas merupakan objek yang akan menjadi perhatian dalam suatu database. Entitas dapat berupa manusia, tempat, benda, atau kondisi mengenai data yang dibutuhkan. Simbol dari entitas berbentuk persegi Panjang[26].

#### 2. Atribut

Atribut merupakan informasi yang terdapat dalam entitas. Sebuah entitas harus memiliki *primary key* sebagai ciri khas entitas dan atribut deskriptif. Atribut biasanya terletak dalam tabel entitas atau dapat juga terpisah dari tabel. Simbol atribut berbentuk elips[26].

#### 3. Relasi

Relasi di dalam ERD merupakan hubungan antara dua atau lebih entitas. Simbol dari relasi berbentuk belah ketupat[26]. Relasi yang dapat dimiliki oleh ERD ada beberapa macam, yaitu:

##### 1. *One to One*

Satu anggota entitas dapat berelasi dengan satu anggota entitas lain.

2. *One to Many*




Satu anggota entitas dapat berelasi dengan beberapa anggota entitas lain.

3. *Many to Many*

Beberapa anggota entitas dapat berelasi dengan beberapa anggota entitas lain.

ERD memiliki beberapa simbol yang biasa digunakan saat merancang suatu sistem. Simbol-simbol ERD dapat dilihat pada **Tabel 2.5**.

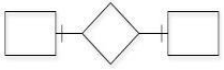
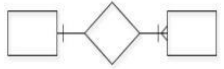
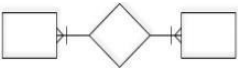
**Tabel 2.5** Simbol-simbol ERD

| No | Nama    | Simbol  | Keterangan   |
|----|---------|---|--|
| 1  | Entitas |    | Persegi panjang menyatakan himpunan entitas adalah orang, kejadian, atau berada.         |
| 2  | Atribut |    | Belah ketupat menyatakan himpunan relasi merupakan hubungan antar entitas.               |
| 3  | Relasi  |  | Atribut merupakan informasi yang diambil tentang sebuah entitas                          |
| 4  | Link    |  | Garis sebagai penghubung antara himpunan, relasi, dan himpunan entitas dengan atributnya |



Kardinalitas merupakan derajat relasi yang dimiliki oleh ERD. Kardinalitas menjelaskan tentang jumlah batasan keterhubungan *entity* satu dengan lainnya. Macam-macam kardinalitas terdapat pada **Tabel 2.6**.

**Tabel 2. 6** Macam-macam kardinalitas

| No. | Nama  | Simbol  | Keterangan  |
|-----|---|---|---|
| 1.  | Relasi Satu ke Satu ( <i>One to One</i> )         |  | Relasi yang menunjukkan bahwa setiap himpunan entitas berhubungan dengan tepat satu himpunan entitas lainnya                          |
| 2.  | Relasi Satu ke Banyak ( <i>One to Many</i> )      |  | Relasi yang menunjukkan bahwa hubungan antara entitas pertama dengan entitas kedua adalah satu banding banyak, begitu pula sebaliknya |
| 3.  | Relasi Banyak ke Banyak ( <i>Banyak to Many</i> ) |  | Relasi yang menunjukkan bahwa setiap himpunan entitas boleh berhubungan dengan banyak himpunan entitas lainnya dan sebaliknya         |

*~Halaman Ini Sengaja Dikosongkan~*