



# **BAB II**

# **LANDASAN TEORI**

## BAB II

### LANDASAN TEORI

#### 2.1 Landasan Teori

##### 2.1.1 Sistem Informasi

Sistem informasi adalah suatu sistem didalam suatu organisasi yang menyediakan proses dan informasi sehingga dapat mempertemukan kebutuhan pengelolaan transaksi harian yang mendukung operasi yang bersifat manajerial dengan strategi dari organisasi untuk menyediakan laporan-laporan untuk pihak luar dan untuk pemangku kepentingannya[1]. Menurut beberapa ahli pengertian sistem informasi adalah sebagai berikut:

1. Pengertian sistem menurut Tukino, Sistem adalah rangkaian jaringan kerja dari berbagai komponen yang saling berhubungan untuk mencapai tujuan tertentu[2].
2. Pengertian informasi menurut Martin Halomoan Lumbangaol, informasi adalah hasil pemrosesan data yang relevan dan bermanfaat bagi pengguna[3].
3. Pengertian sistem informasi menurut Wahyudi & Ridho, sistem informasi merupakan sejumlah komponen yang dimana komponen itu saling berhubungan satu sama lainnya guna untuk mencapai sebuah tujuan yang diharapkan[4].
4. Pengertian sistem informasi menurut Anjelita & Rosiska, sistem informasi adalah sebuah hubungan dari data dan metode dan menggunakan hardware serta software dalam menyampaikan sebuah informasi yang bermanfaat[5].

Dari berbagai pernyataan diatas dapat disimpulkan bahwa sistem informasi merupakan sebuah kumpulan dari beberapa komponen yang mengelola data supaya data yang diolah dapat dijadikan sebagai informasi yang bermakna dan dapat membantu mencapai tujuan organisasi. Aliran sistem informasi sangatlah penting pada suatu proses sistem. Dari proses sistem itulah didapatkan permasalahan yang sedang dihadapi, sehingga dapat diketahui nilai guna sebuah sistem informasi apakah masih bisa beroperasi dengan baik atau tidak, sistem yang manual atau sudah sistem yang lebih canggih. Jika sebuah sistem informasi sudah tidak layak dipakai lagi maka perlu dilakukan pembaruan (*upgrade system*) pada sistem supaya sistem dapat berjalan dengan lebih baik dan dalam proses pengolahan data dapat lebih akurat[6].

Sistem informasi merupakan serangkaian prosedur formal yang digunakan untuk mengumpulkan data. Berbagai sumber daya, seperti *hardware*, *software*, *netware*, *brainware*, dan data, digabungkan menjadi sistem informasi. Sistem informasi dapat digunakan untuk merencanakan, mengolah, mengendalikan, serta meracik data dalam suatu organisasi. Hal ini disebabkan oleh fakta bahwa sistem informasi memiliki *input*, model, proses, *output*,

penyimpanan, dan kontrol. Faktor-faktor ini sangat penting untuk keberhasilan bisnis. Secara umum, sistem terdiri dari komponen yang saling berhubungan yang bekerja sama untuk mencapai tujuan tertentu. Tujuan utama sistem komputer adalah mengolah data untuk menghasilkan informasi, yang membutuhkan dukungan dari komponen seperti perangkat keras (*hardware*), perangkat lunak (*software*), dan pengguna (*brainware*). Perangkat keras adalah peralatan komputer itu sendiri, dan perangkat lunak adalah program komputer. Ketiga komponen sistem komputer tersebut harus berfungsi bersama dan membentuk satu kesatuan. Tanpa perangkat lunak, perangkat keras hanyalah benda mati yang tidak berguna. Selain itu, kedua perangkat lunak dan perangkat keras tidak dapat berfungsi tanpa bantuan orang yang bertanggung jawab untuk menjalankannya. Data adalah sumber daya penting bagi suatu organisasi atau perusahaan[1].

### 2.1.2 Reservasi

Reservasi atau pemesanan berasal dari kata pesan yang artinya adalah setiap pemberitahuan, kata, atau komunikasi baik lisan maupun tertulis, yang dikirimkan dari satu orang ke orang lain. Pesan menjadi inti dari setiap proses komunikasi yang terjalin. Pemesanan yaitu menyediakan atau mempersiapkan tempat sebelumnya. Sedangkan *reservation/reservasi* yaitu pemesanan suatu tempat fasilitas. Sehingga sistem informasi reservasi merupakan suatu sistem yang mengelola data pemesanan fasilitas[7].

Reservasi dibutuhkan untuk memudahkan tamu mendapatkan kepastian untuk mendapatkan tempat dan waktu yang dipesan. Pada saat reservasi berlangsung biasanya ditandai dengan adanya proses tukar-menukar informasi antara konsumen dan produsen agar kesepahaman mengenai produk dapat terwujud. Alasan reservasi menjadi sebuah media yang sangat efektif baik bagi produsen maupun bagi konsumen adalah produsen akan dapat melakukan evaluasi terhadap produk yang akan mereka jual melalui tingkat tinggi rendahnya jumlah reservasi jauh sebelum produk tersebut dijual (barang) ataupun diselenggarakan (jasa)[8].

### 2.1.3 Website

*Website* adalah sekumpulan halaman yang berisi informasi berbentuk digital. Informasi tersebut bisa berupa teks, gambar, audio, video, animasi atau gabungan dari semuanya. *Website* pada umumnya dapat diakses oleh banyak orang di seluruh dunia selama ia memiliki koneksi internet. *Website* dibentuk dari tiga komponen yang saling melengkapi. Ketiga komponen tersebut adalah *words, pictures, dan code*[9].

*Words* merupakan kata-kata yang menyusun konten sebuah *website* sehingga dapat dipahami sebagai sebuah informasi. Contohnya pada *website* berita yang memperbaharui beritanya

hampir beberapa sekali. Perbaharuan berita tergantung pada kejadian yang sedang ramai dibicarakan pada saat itu. *Website* berita biasanya lebih menekankan kepada keterbaruan informasi. Semakin menarik informasinya maka semakin besar pula kemungkinan berita menjadi populer dan berdampak dengan meningkatnya kunjungan pada *website* berita tersebut. *Pictures* merupakan gambar yang menghiasi kata-kata ataupun informasi berbentuk gambar yang dipandu dengan kata kata agar dapat lebih mudah dipahami dan dimengerti maksudnya. Contohnya seperti pada *website* sosial media, justru yang menjadi fokus utamanya adalah gambar, jika *website* sosial media tersebut hanya berbasis teks (*words*) maka akan terasa membosankan bagi pengguna. Sedangkan *code* merupakan baris-baris perintah dalam bentuk kode-kode, seperti kode HTML (*HyperText Markup Language*), kode CSS (*Cascading Style Sheet*), dan kode-kode lainnya yang digunakan untuk mengatur bagaimana informasi yang berupa teks (*words*) dan gambar (*pictures*) dapat ditampilkan pada *software* penjelajah *website* (*browser*) baik itu pada komputer maupun *smartphone*[9].

Jika ditinjau dari fungsi, *website* secara umum di kelompokkan ke dalam empat kategori yaitu sebagai fungsi komunikasi, sebagai fungsi informasi, sebagai fungsi transaksi, dan *website* sebagai fungsi hiburan. Berikut adalah penjelasan dari fungsi *website*[9]:

#### 1. Fungsi Komunikasi

Fungsi komunikasi ini muncul pada *website* yang bersifat dinamis. Biasanya *website* ini memiliki fasilitas seperti *email*, *form contact*, *chating* dan forum diskusi. Contohnya adalah media sosial *Facebook*, *Twitter*, *Gmail*, *Kaskus*, dan lain-lain.

#### 2. Fungsi Informasi

Fungsi informasi dari sebuah *website* dapat dilihat pada kualitas konten *website* yang bersangkutan, dimana keterbaruan dan kebenaran informasi yang disampaikan merupakan hal yang penting. Fungsi informasi ini biasanya tampak dari tersedianya fasilitas *News*, *Profile Company*, *Reference*, seagai contoh adalah *website* berita, *website* perusahaan, *website* Wikipedia, dan lain-lain.

#### 3. Fungsi Transaksi

Fungsi transaksi pada *website* biasanya dari perusahaan, perdagangan ataupun dari usaha-usaha lain yang berhubungan dengan keuangan. Dalaam *website* ini terdapat fasilitas untuk pengecekan saldo, transfer, pembelian, pembayaran, investasi, administrasi. Contoh *website* dengan fungsi transaksi adalah *website* perbankan, *website* belanja *online*, *website* PayPal, dan lain-lain.

#### 4. Fungsi Hiburan

*Website* yang fungsi utamanya hiburan memiliki tampilan gambar, video, dan animasi yang dominan untuk meningkatkan presentasi visualnya. Misalnya *website* sosial media, *website* permainan, *website* film, *website* musik, dan lain-lain.

#### 2.1.4 Rental

Pengertian sewa menurut Kamus Besar Bahasa Indonesia tahun 2008 adalah pemakaian sesuatu dengan membayar uang sewa, uang yang dibayarkan karena memakai atau meminjamkan sesuatu, yang boleh pakai dengan membayar uang dengan uang[10]. Rental merupakan suatu usaha dibidang jasa yang dalam kegiatan usahanya terdapat unsur sewa menyewa yang didalamnya terdapat suatu perjanjian atau kesepakatan dimana penyewa harus membayarkan atau memberikan imbalan kepada pemilik barang yang dipinjamkan[11]. Setiap organisasi, terutama lembaga yang berhubungan bisnis, memerlukan pengelolaan data dan informasi yang efektif. Sistem penyewaan studio dan alat musik adalah contohnya[10].

#### 2.1.5 Studio Musik

Sebagian besar studio musik memiliki alat-alat musik lengkap seperti gitar, *drum*, *keyboard*, *bass*, *Amplifier*, *mixer*, *microphone*, dan masih banyak yang lainnya[12]. Bidang usaha studio musik ini telah lama muncul. Karena memiliki segmen pasar yang konsisten (pemain musik). Bisnis ini tetap eksis seiring berjalannya waktu. Penting bagi para pengusaha untuk menyesuaikan diri dengan zaman yang terus berkembang ini dengan memahami kebutuhan pelanggan dan sesuai dengan perkembangan industri hiburan yang menghasilkan artis dan musisi, yang menghasilkan berbagai jenis bisnis yang berkaitan dengan musik seperti studio musik dan *home recording*. Studio musik adalah usaha yang didirikan untuk memenuhi hasrat berlatih dan keinginan mereka untuk bermusik. Pengguna jasa studio musik adalah orang-orang yang suka bermusik tetapi tidak memiliki alat musik yang mereka butuhkan untuk menikmati hobi dan berlatih.

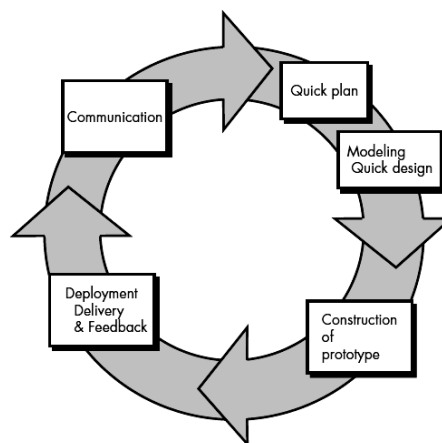
Studio musik di sini juga menyediakan *home recording*. *Home Recording* atau biasa disebut dengan metode rekaman rumahan adalah metode sederhana yang memanfaatkan kebutuhan perangkat secara sederhana namun dapat menghasilkan dengan maksimal. Metode ini tidak membutuhkan perangkat rekaman mahal seperti yang digunakan oleh studio rekaman besar label. Sebaliknya, ia dapat melakukan *home recording* dengan menggunakan perangkat yang sederhana dan dengan harga yang relatif murah. Karena aplikasinya yang fleksibel, ia dapat digunakan di rumah atau tempat lainnya[13]. Sebagian musisi memutuskan untuk beralih ke sistem perekaman secara digital yang kita kenal dengan sebutan DAW (*Digital Audio*

*Workstation*). Rekaman musik di rumah (*home recording*) dapat dicapai dengan bantuan DAW. *Home Recording* ini sendiri bisa dimulai dengan menggunakan komputer atau laptop, *soundcard*, *midi controller* atau instrumen musik, dan *speaker flat* atau monitor, yang saat ini banyak dijual dengan harga terjangkau[14].

### 2.1.6 Metode *Prototype*

Metode pengembangan sistem yang akan digunakan penulis adalah metode *prototype*. *Prototype* adalah metode pengembangan sistem yang sering digunakan untuk memungkinkan interaksi dengan pelanggan selama proses pembuatan sistem[15]. Menurut Pressman, pelanggan seringkali hanya menyebutkan tujuan umum perangkat lunak yang akan digunakan, tetapi tidak menyebutkan fitur dan fungsi yang dibutuhkan. Sebaliknya, pengembang tidak yakin dengan efisiensi Algoritma, kemampuan untuk beradaptasi dengan sistem operasi, atau bagaimana interaksi manusia dengan komputer. Ketika persyaratan tidak jelas, Model *Prototyping* memungkinkan pengembang dan pelanggan berkomunikasi tentang sistem yang akan dibuat[16].

Model *Prototype* yang digunakan oleh sistem akan mengizinkan pengguna mengetahui tahapan sistem yang dibuat sehingga sistem dapat berfungsi. Beberapa keuntungan menggunakan metode *prototype* antara lain yaitu *prototype* memungkinkan pengguna terlibat langsung dalam proses analisa dan desain, *prototype* dapat memahami kebutuhan secara nyata daripada abstrak, dan *prototype* dapat digunakan untuk memperjelas SDLC[17]. Dalam model *prototype* terdapat 5 tahap pengembangan *Prototype* menurut Pressman, antara lain ialah[15]:



**Gambar 2. 1** Metode *Prototype* Menurut Pressman

### 1. *Communication* / Komunikasi

Tim pengembang perangkat lunak bertemu dengan para stakeholder untuk mengidentifikasi kebutuhan perangkat lunak yang saat ini diketahui dan untuk menentukan wilayah yang memerlukan definisi lebih lanjut untuk iterasi.

### 2. *Quick Plan* / Perencanaan Secara Cepat

Perencanaan ini melakukan iterasi pembuatan prototipe dengan cepat, dan pemodelan dilakukan dalam bentuk "rancangan cepat".

### 3. *Modeling Quick Design* / Model Rancangan Cepat

Tahap ini yaitu memodelkan perencanaan sebelumnya dengan menggunakan sejumlah model berorientasi objek dengan menggunakan alat UML, seperti *Usecase* yang mendefinisikan fungsi sistem, *Class Diagram* yang menunjukkan *class-class* sistem, dan *Activity Diagram* yang menunjukkan alur proses bisnis.

### 4. *Construction of Prototype* / Konstruksi *Prototype*

Rancangan cepat berfokus pada menggambarkan elemen perangkat lunak yang akan digunakan oleh pengguna akhir (*end user*), seperti format tampilan atau antarmuka pengguna. Pembuatan prototipe dimulai dengan rancangan cepat.

### 5. *Deployment Delivery & Feedback* / Penyerahan Dan Memberikan Umpan Balik Terhadap Pengembangan

Rancangan cepat berfokus pada menggambarkan elemen perangkat lunak yang akan digunakan oleh pengguna akhir (*end user*), seperti format tampilan atau antarmuka pengguna. Pembuatan prototipe dimulai dengan rancangan cepat.

#### 2.1.7 *Flowchart*

*Flowchart* atau diagram alir merupakan suatu jenis diagram yang menggambarkan algoritma atau langkah-langkah instruksi yang runtut di dalam sistem. *Flowchart* digunakan untuk memberi programmer gambaran logis dari sistem yang akan dibangun, analis sistem menggunakan *flowchart* sebagai bukti dokumentasi dan membantu mereka menyelesaikan masalah. Pada dasarnya *flowchart* digambarkan menggunakan simbol-simbol, setiap simbol mewakili suatu proses, sedangkan garis berfungsi untuk menghubungkan suatu proses ke proses selanjutnya[18].

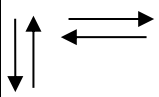


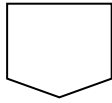
*Flowchart* digunakan oleh analis sistem sebagai bentuk dokumentasi yang berguna untuk menjelaskan gambaran yang logis dalam sebuah sistem yang akan dibuat oleh *programmer*. *Programmer* akan menafsirkan desain logis ke dalam bentuk program dengan berbagai bahasa pemrograman yang telah di sepakati setelah pembuatan *flowchart* selesai. *Flowchart* memiliki banyak fungsi lainnya, seperti merancang proyek baru, mengelola alur kerja, memodelkan proses

bisnis, mendokumentasikan setiap proses, merepresetasikan algoritma, dan mengaudit proses. Merancang *flowchart* adalah proses yang didasarkan pada pemikiran untuk mengevaluasi masalah bisnis, tidak ada persyaratan mutlak yang harus dipenuhi. Terdapat simbol-simbol yang harus di pahami untuk mulai merancang *flowchart*. Simbol-simbol *flowchart* dibagi ke dalam 3 bagian, yaitu[18]:

#### 1. Simbol Arus (*Flow Direction Symbols*)

Simbol Arus digunakan sebagai simbol penghubung. Simbol yang masuk ke dalam kategori ini, antara lain:


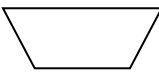
**Tabel 2. 1** *Flowchart* Simbol Arus

No	Simbol	Nama	Fungsi
1.		<i>Flow Direction Symbol/ Connecting Line</i>	Menghubungkan simbol yang satu dengan yang lainnya, menyatakan arus suatu proses.
2.		<i>Communication Link</i>	Berfungsi untuk transmisi data dari suatu lokasi ke lokasi lain.
3.		<i>On Page Connector</i>	Digunakan untuk menyatakan sambungan dari proses yang satu ke proses berikutnya di halaman yang sama.
4.		<i>Off Page Connector</i>	Digunakan untuk menyatakan sambungan dari proses yang satu ke proses berikutnya di halaman yang berbeda.

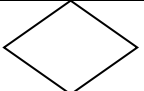
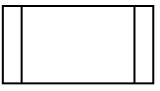
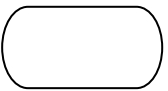
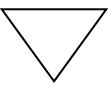

#### 2. Simbol Proses (*Processing Symbol*)

Digunakan untuk menyatakan simbol proses yang berkaitan dengan serangkaian proses yang dilakukan. Simbol yang masuk ke dalam kategori ini, antara lain:

**Tabel 2. 2** *Flowchart* Simbol Proses

No	Simbol	Nama	Fungsi
1.		<i>Processing</i>	Digunakan untuk menunjukkan pengolahan yang akan dilakukan dalam komputer.
2.		<i>Manual Operation</i>	Digunakan untuk menunjukkan pengolahan yang tidak dilakukan oleh komputer.

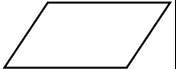

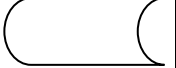


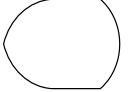


3.		Decision	Digunakan untuk memilih proses yang akan dilakukan berdasarkan kondisi tertentu.
4.		<i>Predefined Process</i>	Digunakan untuk mempersiapkan penyimpanan yang sedang/akan digunakan dengan memberikan harga awal.
5.		<i>Terminal</i>	Digunakan untuk memulai atau mengakhiri program.
6.		<i>Offline Storage</i>	Berfungsi untuk menunjukkan bahwa data akan disimpan ke media tertentu.
7.		<i>Manual Input Symbol</i>	Digunakan untuk menginputkan data secara manual dengan keyboard.

### 3. Simbol I/O (*Input-Output*)

Simbol yang termasuk dalam input-output berkaitan dengan masukan dan keluaran. Simbol yang masuk ke dalam kategori ini, antara lain:

**Tabel 2. 3** *Flowchart* Simbol I/O

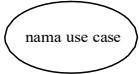

No	Simbol	Nama	Fungsi
1.		<i>Input/Output</i>	Digunakan untuk menyatakan input dan output tanpa melihat jenisnya.
2.		<i>Punched Card</i>	Digunakan untuk menyatakan masukan dan keluaran yang berasal dari card.
3.		Disk Storage	Digunakan untuk menyatakan masukan dan keluaran yang berasal dari disk.
4.		<i>Magnetic Tape</i>	Digunakan untuk menyatakan masukan dan keluaran yang berasal dari magnetis.
5.		<i>Document</i>	Digunakan untuk menyatakan masukan dan keluaran yang berasal dari dokumen.
6.		<i>Display</i>	Berfungsi untuk menyatakan keluaran melalui layar monitor.


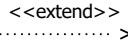

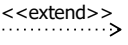

### 2.1.8 Use Case Diagram

Salah satu diagram UML (*Unified Modeling Language*) adalah *use case*, yang merupakan alat bantu teknis untuk menyelesaikan suatu masalah dengan pendekatan berorientasi objek. Termasuk perangkat lunak untuk memvisualisasikan, merancang, dan mendokumentasikan sistem, UML telah menjadi standar dalam industri rekayasa sistem. Suatu diagram yang menggambarkan fungsi-fungsi yang diharapkan dari sebuah sistem yang dikembangkan disebut *use-case diagram*. *Use case* digunakan untuk menentukan interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibangun. Sebuah contoh interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibangun disebut *use case*[19].

Secara umum, *use case* digunakan untuk menentukan fungsi sistem informasi apa yang ada dan siapa yang berhak menggunakannya. *Use-case diagram* yang digunakan dapat sangat membantu pada saat kita menyusun requirement sebuah sistem, mengomunikasikannya dengan klien, dan merancang pengujian untuk semua fitur yang terdapat dalam sistem. Dalam suatu sistem aplikasi berbasis komputer, *use-case diagram* sangat membantu menggambarkan requirement apa saja sesuai proses bisnis dan cakupan permasalahannya. *Use Case Diagram* memiliki beberapa simbol dan fungsinya masing-masing, di antaranya adalah[19]:

**Tabel 2. 4 Use Case Diagram**

No	Gambar	Nama	Deskripsi
1.		<i>Use case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase nama use case.
2.		Aktor / <i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari actor adalah gambar orang, tetapi actor belum tentu merupakan orang; biasanya dinyatakan

			menggunkan kata benda di awal frase nama aktor. Aktor merupakan peran yang dimainkan oleh pemakai ketika berinteraksi dengan sistem.
3.		Asosiasi / <i>Association</i>	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
4.		Ekstensi / <i>Extensi</i>	Relasi use case tambahan ke sebuah use case yang ditambahkan dapat berdiri sendiri walaupun tanpa use case tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; biasanya use case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan.
5.		Generalisasi / <i>Generalization</i>	Hubungan generaliasi dan spesialisasi (umum – khusus) antara dua buah use case Dimana fungsi yang satu adalah fungsi yang lebih umum daripada lainnya.
6.	 	<i>Include / Uses</i>	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini.

### 2.1.9 Entity Relationship Diagram (ERD)

Teknik *Entity-Relationship Diagram* (ERD) adalah teknik yang digunakan untuk tahap dasar dalam membuat database. ERD merupakan salah satu teknik merancang *database* yang paling banyak digunakan. ERD berdasar pada model *entity-relationship*. Data pada model *entity-relationship* yang direpresentasikan visual disebut dengan ERD. ERD ini memrepresentasikan bagaimana entitas saling terkait antara satu dengan yang lainnya dalam database. Dengan dibuat ERD dipercaya dapat membantu perancang dalam menganalisis database yang dibuat, meskipun begitu terkadang masih banyak kesalahan yang terjadi saat mendesain ERD. Banyak hal-hal yang harus diperhatikan dalam merancang ERD pada pembuatan database agar ERD yang dibuat sesuai dengan ketentuan[20].

*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[20]. ERD memiliki 3 elemen dasar yaitu[21]:

#### 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.

#### 2. Atribut

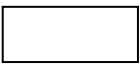
Atribut merupakan informasi yang terdapat dalam entitas. Sebuah entitas harus memiliki *primary key* sebagai ciri khas entitas dan atribut deskriptif. Atribut biasanya terletak dalam tabel entitas atau dapat juga terpisah dari tabel. Simbol dari atribut berbentuk *elips*.

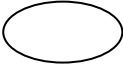

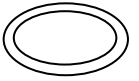
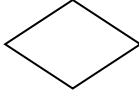
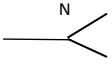
#### 3. Relasi

Relasi di dalam ERD merupakan hubungan antara dua atau lebih entitas. Simbol dari relasi berbentuk belah ketupat.

Pada ERD terdapat simbol-simbol yang harus dipahami. Berikut adalah simbol *ERD* beserta fungsinya:

**Tabel 2. 5 Entity Relationship Diagram**

No	Simbol	Nama	Fungsi
1.		Entitas/ <i>Entity</i>	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi computer;

			penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.
2.		Atribut	Field atau kolom data yang butuh disimpan dalam suatu entitas.
3.		Atribut kunci primer	Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).
4.		<i>Atribut Multinilai/ Multivalue</i>	Field atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
5.		Relasi	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.
6.		Asosiasi/ <i>Association</i>	Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki multiplicity kemungkinan jumlah pemakaian kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan one to many menghubungkan entitas A dan entitas B.

### 2.1.10 *Object Oriented Programming (OOP)*

*Object Oriented Programming (OOP)* merupakan strategi pembangunan perangkat lunak yang bertujuan untuk mengorganisasikan perangkat lunak sebagai kumpulan objek yang mengandung data dan operasi yang dilakukan padanya. Paradigma atau teknik pemrograman berorientasi objek, atau OOP, didefinisikan sebagai pemrograman berorientasi objek. Berdasarkan definisi yang ada, dapat dikatakan bahwa pemrograman berorientasi objek (OOP) adalah suatu pendekatan atau cara baru untuk membuat program atau merancang sistem dengan memperhatikan objek[22].

Metode pemrograman berorientasi objek (OOP) adalah metode yang paling populer dan banyak digunakan. Hal ini disebabkan oleh fakta bahwa metodologi lama banyak menimbulkan masalah, seperti masalah saat mengubah hasil dari satu tahap pengembangan ke tahap berikutnya, seperti dalam pendekatan terstruktur, misalnya jenis aplikasi yang dikembangkan saat ini berbeda dengan jenis aplikasi yang dikembangkan sebelumnya. Beberapa keuntungan penggunaan berorientasi objek adalah sebagai berikut[22]:

a. Meningkatkan produktivitas

Karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lainnya yang melibatkan objek tersebut (*reusable*).

b. Kecepatan pengembangan

Karena sistem yang dibangun dengan baik dan benar pada analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada pengkodean.

c. Kemudahan pemeliharaan

Karena dengan model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin sering berubah-ubah.

d. Adanya konsistensi

Karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.

e. Meningkatkan kualitas perangkat lunak

Karena pendekatan pengembangan lebih dekat dengan dunia nyata dan adanya konsistensi pada saat pengembangannya, perangkat lunak yang dihasilkan akan maupun memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.

### 2.1.11 *Framework Laravel*

Framework adalah adalah tempat untuk sebuah website yang akan dibangun. Melakukan perbaikan lebih mudah dengan menggunakan *framework* karena waktu yang dibutuhkan untuk membuat situs web menjadi lebih singkat. Banyak keuntungan yang dapat diperoleh dalam

menggunakan *framework* untuk membuat sebuah *website*. Salah satu manfaatnya adalah memberikan struktur yang baik untuk program yang dibuat karena struktur memiliki *library* atau fungsi yang dapat diakses secara instan. Selain itu, rangka kerja membuat pekerjaan tim lebih mudah karena membangun *website* memerlukan penyesuaian dengan gaya *framework* yang digunakan[23].

*Laravel* adalah *framework* berbasis *PHP* yang sifatnya *open source* yang menggunakan konsep *model-view-controller*. *Laravel* memiliki beberapa kekurangan saat digunakan, salah satunya adalah ukuran *file* yang besar. *File default* seperti *vendor* ada di *Laravel*. Hal tersebut menyebabkan *file* tidak boleh dihapus secara sembarangan, sehingga ukuran situs *web* yang dibuat cukup besar. Instalasi dan mengunduh *library Laravel* membutuhkan koneksi *internet*, dan untuk menjalankannya, *PHP* minimal versi 5.4 diperlukan. Berikut ini adalah dasar-dasar *laravel*[23].

#### a. *Artisan*

Perintah *Artisan* dijalankan melalui terminal dan menyediakan berbagai perintah perintah yang dapat digunakan sewaktu mengembangkan dan membuat aplikasi. *Php artisan serve* adalah fungsi dari *php artisan*, yang memungkinkan untuk membuka *website* yang telah dibuat tanpa menggunakan *web server* lokal.

#### b. *Routing*

Item yang diinginkan dibawa ke tujuan melalui proses yang disebut *routing*. Halaman mana yang akan muncul ketika *web browser* dibuka dapat ditentukan dengan menggunakan *routing*. *Folder routes* adalah letak di mana *file web.php* biasanya berada.

#### c. *Controller*

*Controller* adalah proses yang bertujuan untuk mengambil permintaan, menginisialisasi, dan memanggil model untuk dikirimkan ke *view*. Di *Laravel*, ada dua metode untuk membuat *controller*. Pertama, *file controller* dibuat secara manual dan dimasukkan kode *extends controller*. Cara kedua, *file controller* dapat dibuat menggunakan *command line* dengan menggunakan *php artisan make controller* nama *\_file\_controller*.

#### d. *View (blade templating)*

*Blade* merupakan *template engine* bawaan yang berasal dari *Laravel*, dan memiliki kode kode yang lebih mudah untuk digunakan saat menggunakan *Laravel*. Membuat nama *\_file.php.blade* di *folder views* adalah cara membuat *file.blade* secara manual. Dalam *blade*, dapat dibuat *template master* dan *template inheritance*. Pembuatan kedua *template* ini bertujuan untuk mencegah elemen yang sama ditulis secara berulang. Kode *extend* (nama *\_layout*) dan *section* (nama *\_content*) diberikan pada *template inheritance*.

#### e. *Middleware*

*Middleware* berfungsi sebagai penghubung antara *controller* yang dituju dan *request* yang masuk. Cara menggunakan artisan untuk membuat *middleware* adalah dengan mengetikkan *php artisan make:middleware nama\_file*. File *middleware* berada di folder *middleware*.

#### f. Session

*Session* merupakan cara untuk menyimpan pada *server* dan menyimpan pada beberapa halaman, termasuk halaman itu sendiri. Ada dua cara untuk menggunakan *session*. Pertama yaitu *request* dan yang kedua fungsi *global helper session*.

#### g. Migration

*Migration* merupakan fitur yang terdapat dalam *Laravel* dan merupakan sistem kontrol versi database. *Table* data lebih mudah dan cepat di buat menggunakan *migration*, penulis dapat membuat *table* data dengan lebih mudah dan cepat. *Migration* juga membuat atau membuat *file migration* sebagai *control system*. Saat *programmer* membuat aplikasi, mereka dapat mengirimkan *file-file* ini antara satu sama lain. Cara menggunakan *artisan* untuk membuat *migration* adalah dengan menggunakan *make:migration*, mengetikkan *php create\_namatable\_table*, dan *artisan create=namatable*.

#### h. Model

Model adalah salah satu komponen *MVC* yang memiliki hubungan langsung dengan *database*. Bisa juga dikatakan bahwa model menghubungkan setiap alur program yang berkaitan dengan data. Dengan demikian, konsep *MVC* berjalan, *Controller* dapat menggunakan atau memanggil model yang terhubung ke *database*. Cara menggunakan *artisan* untuk membuat model adalah dengan mengetikkan *php artisan make:model nama\_model*.

### 2.1.12 Basis Data (Database)

*Database* adalah sistem penyimpanan data yang dirancang untuk memenuhi kebutuhan informasi bisnis dan terdiri dari sekumpulan data yang secara logika saling terkait yang disusun dalam tabel atau relasi. Model database relasional adalah model yang paling umum karena struktur logikanya yang sederhana. Dalam model relasional, seluruh data disusun dalam relasi atau tabel, dan setiap relasi terdiri dari baris dan kolom yang diberi nama tertentu, yang disebut atribut.

Pemakai *database* dapat berupa individu atau program aplikasi. Orang biasanya menggunakan *database* dari terminal dan menggunakan *query language* untuk mengambil data dan informasi. *Query* adalah permintaan informasi dari *database*, dan *query language* adalah bahasa khusus yang mudah digunakan dan memungkinkan komputer untuk menjawab pertanyaan[24]. Dalam penggunaannya SQL dikategorikan menjadi tiga sub perintah,



yaitu DDL (*Data Definition Language*), DML (*Data Manipulation Language*) dan DCL (*Data Control Language*)[25].

**a. Data Definition Language (DDL)**

Pengertian dari DDL (*Data Definition Language*) adalah perintah SQL yang digunakan untuk mendefinisikan, menciptakan, dan menghapus sebuah database ataupun tabel. Yang termasuk dalam perintah DDL meliputi *CREATE*, *DROP*, dan *ALTER*

1. *Create*, digunakan untuk membuat data, termasuk diantaranya membuat database baru, tabel baru, tampilan baru dan kolom.
2. *Alter*, digunakan untuk mengubah struktur tabel yang telah dibuat. Tugasnya mencakup mengganti nama tabel, menambah kolom, mengubah kolom, menghapus kolom, maupun memberikan atribut pada kolom.
3. *Drop*, digunakan untuk menghapus database dan tabel.

**b. Data Manipulation Language (DML)**

*Data Manipulation Language* (DML) merupakan bahasa untuk memanipulasi/ mengubah isi tabel. Perintah yang digunakan diantaranya adalah:

1. *Insert*, digunakan untuk menyisipkan atau memasukkan data baru ke dalam tabel. Penggunaannya setelah database dan tabel dibuat.
2. *Update*, digunakan untuk mengubah atau memperbaharui data lama menjadi data baru.
3. *Delete*, digunakan untuk menghapus data dari tabel. Pada saat penghapusan data, perintah yang telah dijalankan tidak dapat digagalkan, sehingga data yang telah terhapus tidak dapat dikembalikan lagi.

**c. Data Control Language (DCL)**

*Data Control Language* (DCL) merupakan bahasa yang berhubungan dengan pengendalian akses ke database. Perintah DCL diantaranya adalah:

1. *Grant*, digunakan untuk memberikan hak/izin akses oleh administrator (pemilik utama) *server* kepada *user* (pengguna biasa). Hak akses tersebut berupa hak membuat (*CREATE*), mengambil (*SELECT*), menghapus (*DELETE*), mengubah (*UPDATE*) dan hak khusus berkenaan dengan sistem database.
2. *Revoke*, perintah *revoke* memiliki kegunaan terbalik dengan *GRANT*, yaitu untuk menghilangkan atau mencabut hak akses yang telah diberikan kepada user oleh administrator.