

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Tinjauan Studi**

Pendidikan adalah usaha terencana untuk dapat menciptakan proses belajar mengajar dan proses pembelajaran agar peserta didik secara aktif dapat mengembangkan potensi dirinya untuk memiliki kekuatan spiritual keagamaan, pengendalian diri, kepribadian, kecerdasan, akhlak yang baik, serta terampil untuk dirinya [ Septiani, 2017]

Pendidikan Anak Usia Dini merupakan pendidikan yang akan melibatkan seluruh anak untuk mencakup kepedulian terhadap perkembangan fisik, kognitif, dan lingkungan sosial anak. Pembelajaran dikelompokkan sesuai dengan minat-minat dan gaya belajar anak.

PAUD berfungsi membina, menumbuhkan dan mengembangkan seluruh potensi anak usia dini secara optimal sehingga terbentuk perilaku dan kemampuan dasar sesuai dengan tahap perkembangannya. Disamping itu dalam perkembangan pendidikan PAUD tidak hanya mengutamakan pembangunan secara fisik namun juga mampu memberikan pelayanan optimal yang sesuai dengan Undang-undang dan peraturan yang berlaku [ Ilmiah, 2018]

Perkembangan layanan program pendidikan anak usia dini (PAUD) disambut secara positif oleh berbagai kalangan. Menjamurnya berbagai jenis layanan PAUD seiring dengan sosialisasi dan perluasan kebijakan pemerintah terhadap penyebar luasan PAUD merupakan salah satu isu internasional khususnya berkenaan dengan pendidikan dan perawatan.

##### **1.1.1 Model-Model Pembelajaran Pada Anak Usia Dini**

Pembelajaran anak usia dini dapat dikelompokkan menjadi tiga pendekatan, yaitu: pembelajaran bebas, pembelajaran terpimpin, dan pembelajaran kondusif .

##### **1. *Pembelajaran Bebas***

Pembelajaran bebas merupakan suatu strategi pembelajaran yang memberikan kesempatan kepada anak yang seluas-luasnya untuk mendapatkan pengalaman belajar yang bermakna, sehingga anak lebih senang dan enjoy dalam menerima pembelajaran ini.

Strategi ini bisa menguntungkan anak yang memiliki kekuatan untuk mandiri. Anak tidak terlalu tergantung terhadap gurunya. Bila perlu anak datang kepada guru. Kreativitas-kreativitasnya dapat berkembang lebih pesat. Tetapi sebaliknya untuk anak yang kurang mandiri, model pembelajaran ini dapat menimbulkan frustrasi, tidak tahu apa yang harus dilakukan, putus asa, cemas, bosan, bingung, dan tidak terkendalikan terhadap diri anak.

## **2. *Pembelajaran Terpimpin***

Pembelajaran terpimpin merupakan strategi yang sepenuhnya akan dikendalikan oleh guru. Guru akan lebih banyak aktif berbicara dan anak mendengarkan guru saat berbicara, guru memberi contoh anak akan mengikutinya dan perintah saat belajar yang diberikan guru, melakukan drill dan latihan sesuai rencana guru. pembelajaran ini bertujuan untuk mengetahui Anak yang tidak dapat menangkap dengan baik contoh, anak yang tidak bisa menangkap dengan baik akan dipisahkan dan dibetulkan guru atau diberi pengawasan khusus.

Pembelajaran tersebut bertujuan untuk Anak merasa berhasil kalau ia dapat menjalankan apa kehendak guru.

## **3. *Pembelajaran Kondusif (Supportive climate)***

Pembelajaran kondusif merupakan campuran dari pembelajaran bebas dengan suasana pembelajaran terpimpin. Guru dan anak berbagi/ bertukar proses pembelajaran dan pengalaman.

Guru selalu berusaha menyeimbangkan secara efektif antara kebebasan aktif anak untuk berkreasi tetapi tetap membatasi agar anak merasa aman ketika sedang belajar. Guru memberikan lingkungan pembelajaran dengan sesuai pilihan minat anak. Rutinitas teratur. dan tetap diberi leluasa untuk menuangkan kreatifitas anak.

### 1.1.2 Model Pembelajaran

Model pembelajaran adalah suatu cara bagaimana pendidik menyampaikan materi agar tujuan pembelajaran tercapai . Model pembelajaran harus berorientasi pada :

- a. Kebutuhan anak. Kegiatan pembelajaran pada anak usia dini harus senantiasa berorientasi pada kebutuhan anak untuk mendapatkan layanan pendidikan, kesehatan dan gizi yang dilaksanakan secara integratif dan holistik.
- b. Belajar melalui bermain. Bermain merupakan pendekatan dalam melaksanakan kegiatan pendidikan anak usia dini , dengan menggunakan strategi, metode, materi / bahan, dan media yang menarik agar mudah diikuti oleh anak. Melalui bermain anak diajak untuk bereksplorasi ( penjajakan ), menemukan dan memanfaatkan benda – benda disekitarnya.
- c. Kreatif dan inovatif. Proses kreatif dan inovatif dapat dilakukan melalui kegiatan – kegiatan yang menarik , membangkitkan rasa ingin tahu anak, memotivasi anak untuk berpikir kritis, dan menemukan hal – hal baru.
- d. Lingkungan yang kondusif. Lingkungan harus diciptakan sedemikian menarik dan menyenangkan dengan memperhatikan keamanan dan kenyamanan anak dalam bermain.
- e. Menggunakan pembelajaran terpadu. Model pembelajaran terpadu yang beranjak dari tema yang menarik anak ( Center Of Interest ) dimaksudkan agar anak mampu mengenal berbagai konsep secara mudah dan jelas sehingga pembelajaran menjadi bermakna bagi anak.
- f. Mengembangkan keterampilan hidup. Mengembangkan keterampilan hidup melalui pembiasaan – pembiasaan agar mampu menolong diri sendiri (mandiri), disiplin, mampu bersosialisasi, dan memperoleh bekal keterampilan dasar yang berguna untuk kelangsungan hidupnya.
- g. Menggunakan berbagai media dan sumber belajar. Media dan sumber belajar dapat berasal dari lingkungan alam sekitar atau bahan – bahan yang sengaja disiapkan.

- h. Pembelajaran yang berorientasi pada prinsip – prinsip perkembangan anak, diantaranya : anak belajar dengan sebaik – baiknya apabila kebutuha fisiknya terpenuhi serta merasakan aman dan tentram secara psikologis, siklus belajar anak selalu berulang dimulai dari membngun kesadaran, melakukan penjelajahan, memperoleh penemuan untuk selanjutnya anak dapat menggunakannya. Anak belajar melalui interaksi sosial dengan orang dewasa dan teman sebayanya, minat anak dan keingintahuannya memotivasi belajarnya, perkembangan belajar anak harus memperhatikan perbedaan individual, anak belajar dari sederhana ke yang rumit, dari konkret ke abstrak, dari gerakan ke verbal, dan dari keakuan ke rasa sosial.
- i. Stimulasi terpadu. Pada saat anak melakukan suatu kegiatan anak dapat mengembangkan beberapa aspek pengembanngan sekaligus. Contoh : Ketika anak melakukan kegiatan makan, kemampuan yang dikembangkan antara lain : bahasa ( mengenal kosakata tentang jenis sayuran, dan peralatan makan ), motorik halus ( memegang sendok, menyuap makanan ke mulut ), daya pikir ( membandingkan makanan sedikit dan banyak ), sosial emosional ( duduk rapi dan menolong diri sendiri ), dan moral ( berdo'a sebelum dan sesudah makan).

## 2.2. Tinjauan Pustaka

Sistem informasi geografis (SIG) adalah salah satu sistem yang dapat digunakan untuk keperluan mengoptimalkan sistem yang ada di atas. Berbeda dengan sistem informasi yang lain, SIG mempunyai kemampuan analisis spasial, dimana kemampuan ini dapat memberikan hasil yang lebih komprehensif dibandingkan dengan sistem informasi yang hanya berbasis statistik. Prasarana pendidikan dapat dilihat kondisinya berdasarkan keberadaan lokasi, seperti kemudahan terjangkaunya, luas wilayah layanan dan kondisi fisik prasarana tersebut [ Kridalaksana, 2016]

Sistem Informasi Geografi adalah “suatu komponen yang terdiri dari perangkat keras (hardware), perangkat lunak (software), data geografis dan sumberdaya manusia yang bekerja bersama secara efektif untuk menangkap, menyimpan, memperbaiki, memperbaharui, mengelola, memanipulasi,



mengintegrasikan, menganalisa, dan menampilkan data dalam suatu informasi berbasis geografis” [ Kridalaksana, 2016]

#### 2.2.1. Fungsi Sistem Informasi Geografis

Fungsi SIG adalah meningkatkan kemampuan menganalisis informasi spasial secara terpadu untuk perencanaan dan pengambilan keputusan. SIG dapat memberikan informasi kepada pengambil keputusan untuk analisis dan penerapan *database* keruangan. SIG mampu memberikan kemudahan-kemudahan yang diinginkan.

Dengan SIG kita akan dimudahkan dalam melihat fenomena kebumihan dengan perspektif yang lebih baik. SIG mampu mengakomodasi penyimpanan, pemrosesan, dan penayangan data spasial *digital* bahkan integrasi data yang beragam, mulai dari citra satelit, foto udara, peta bahkan data statistik. SIG juga mengakomodasi dinamika data, pemutakhiran data yang akan menjadi lebih mudah [ Swastikayana, 2018]

#### 2.2.2. SPK ( Sistem Pendukung Keputusan )

Sistem pendukung keputusan (SPK) adalah sistem yang mampu memberikan kemampuan pemecahan masalah maupun kemampuan pengkomunikasian untuk masalah dengan kondisi semi terstruktur dan tak terstruktur. Sistem ini digunakan untuk membantu pengambilan keputusan dalam situasi semi terstruktur dan situasi yang tidak terstruktur, di mana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat [ Ningsih, 2017]

Sistem Pendukung Keputusan merupakan sekumpulan prosedur berbasis model untuk pemrosesan data dan penilaian guna membantu para pengambil keputusan untuk mengambil keputusan dalam situasi yang semiterstruktural dan situasi yang tidak terstruktur secara cepat dan mudah .

Sistem Pendukung Keputusan adalah sebuah sistem untuk membantu seorang manajer dalam pengambilan keputusan dengan situasi semiterstruktur . SPK dimaksudkan untuk menjadi alat bantu bagi para pengambil keputusan untuk memperluas kapabilitas mereka, namun tidak untuk menggantikan penilaian

mereka. SPK ditujukan untuk keputusan-keputusan yang memerlukan penilaian atau keputusan-keputusan yang sama sekali tidak dapat didukung oleh algoritma [ Ernawati, 2017]

Sistem pendukung keputusan adalah sebuah sistem berbasis komputer dengan antar muka antara mesin/komputer dengan pengguna. Sistem pendukung keputusan ditujukan untuk membantu dalam menyelesaikan suatu masalah dalam berbagai level manajemen dan bukan untuk mengganti posisi manusia sebagai pembuat keputusan. Sistem pendukung keputusan mampu memberi alternatif solusi bagi masalah semi/tidak terstruktur baik bagi perseorangan atau kelompok dan dalam berbagai macam proses dan gaya pengambilan keputusan sistem pendukung keputusan menggunakan data, basis data, dan analisa model-model keputusan. Sistem pendukung keputusan bersifat adaptif, efektif, interaktif, *easy to use* dan *fleksibel* serta menyediakan akses terhadap berbagai macam format dan tipe sumber data (*data source*) [ Sianturi, 2017]

### 2.2.3. Profile matching

*Profile Matching* (PM) merupakan metode yang mengasumsikan bahwa terdapat tingkat variabel prediktor yang ideal yang harus dipenuhi oleh subyek yang diteliti, bukannya tingkat minimal yang harus dipenuhi atau dilewati. Proses *Profile Matching* secara garis besar merupakan proses membandingkan antara nilai data aktual dari suatu profil yang akan dinilai dengan nilai profil yang diharapkan sehingga dapat diketahui perbedaan kompetensinya (nilai *gap*).

*Profil Matching* merupakan sebuah metode dimana metode ini terlebih dahulu menentukan nilai kompetensi (kemampuan) yang diperlukan untuk suatu jabatan. Kompetensi kemampuan tersebut haruslah dapat dipenuhi oleh pemegang atau calon yang akan dinilai kinerjanya [ Nuraeni, 2018]

Metode Profile Matching adalah sebuah mekanisme pengambilan keputusan dengan mengasumsikan bahwa terdapat variable predicator yang ideal yang harus dimiliki, bukannya tingkat minimal yang harus dipenuhi atau dilewati dalam proses Metode Profile Matching Modelling secara garis besar merupakan proses membandingkan antara nilai data actual dari suatu profile yang akan dinilai

dengan nilai profil yang diharapkan, sehingga dapat diketahui perbedaan kompetensinya (disebut juga gap) [ Setiawan, 2017]

Menurut Kusriani menyatakan bahwa *profile matching* atau yang biasa disebut dengan Analisis GAP Kompetensi adalah sebuah mekanisme pengambilan keputusan dengan mengasumsikan bahwa terdapat tingkat variabel prediktor yang ideal yang harus dimiliki oleh pelamar, bukannya tingkat minimal yang harus dipenuhi atau dilewati.

Dalam proses *profile matching* bisa di garis besarkan merupakan proses perbandingan antara kompetensi individu dengan kompetensi jabatan sehingga dapat diketahui perbedaan kompetensinya (disebut juga gap), semakin kecil gap yang dihasilkan maka bobot nilainya semakin besar maka memiliki peluang lebih besar untuk menempati posisi tersebut [ Ichsan, 2016]

#### 2.2.4. Tahapan-Tahapan Pada Metode *Profile Matching*

Menurut [ Kurniasih, 2016] tahapan dalam metode *profile matching* adalah sebagai berikut :

- b. Menentukan Bobot Nilai Gap.
- c. Melakukan pemetaan Gap

Gap = Profil Jabatan – Profil Calon Jabatan.

- d. Melakukan pencocokan dengan table bobot Gap Hasil Gap Tabel bobot nilai bisa dilihat

**Tabel 2. 1 Bobot Nilai Gap**

No	Selisih Gap	Bobot Nilai	Keterangan
1	0	5	Kompetensi sesuai dengan yang dibutuhkan
2	1	4.5	Kompetensi Individual kelebihan 1 tingkat/level
3	-1	4	Kompetensi individual kurang 1 tingkat/level
			Kompetensi individual kelebihan 2

4	2	3.5	tingkat/level
5	-2	3	Kompetensi individual kurang tingkat/level 2
6	3	2.5	Kompetensi individual kelebihan tingkat/level 3
7	-3	2	Kompetensi individual kurang tingkat/level 3
8	4	1.5	Kompetensi individual kelebihan tingkat/level 4
9	-4	1	Kompetensi individual kurang tingkat/level 4

#### 2.2.5. Perhitungan Profile Matching

##### Perhitungan Core dan Secondary Factor

*Core factor* merupakan aspek yang sangat dibutuhkan oleh suatu posisi yang diperkirakan dapat menghasilkan kinerja yang optimal [ Indapuri, 2018]

Rumus perhitungan *Core Factor* di bawah ini :

$$NCT = \frac{\sum NC}{\sum IC} \quad (2.1)$$

NCT : Nilai rata – rata *core factor* teknikal

NC : Jumlah total nilai *Core Factor*

IC : Jumlah *Core Factor*

*Secondary factor* merupakan faktor pendukung yang ada di *core factor*(Khotijah, 2015) Sedangkan untuk perhitungan *secondary factor* dapat ditunjukkan pada rumus di bawah ini :

$$NST = \frac{\sum NS}{\sum IS} \quad (2.2)$$

NST : Nilai rata-rata *secondary factor*

NS :Jumlah total nilai *Secondary Factor*



IS : Jumlah *Secondary Factor*

### Perhitungan Nilai Total

Nilai total berdasar nilai dari *core Factor* dan *secondary factor* yang diperkirakan menjadi pengaruh terhadap kinerja tiap-tiap profil . Contoh perhitungan dapat dilihat pada rumus di bawah ini :

$$60\%NCT + 40\%NST = NT \quad (2.3)$$

#### 2.2.6. Metode Haversine

Metode *haversine* merupakan salah satu metode untuk menghitung jarak dari suatu tempat ke tempat tujuan. Proses kalkulasi jarak membutuhkan 4 masukan antara lain titik koordinat *latitude* dan *longitude* tempat asal serta titik koordinat *latitude* dan *longitude* tempat tujuan. Berikut ini rumus *haversine*

[ Vanese, 2016]

$$\Delta lat = lat2 - lat1 \quad (2.4)$$

$$\Delta long = long2 - long1 \quad (2.5)$$

$$a = \sin^2 \left( \frac{\Delta lat}{2} \right) + \cos(lat1) \cdot \cos(lat2) \cdot \sin^2 \left( \frac{\Delta long}{2} \right) \quad (2.6)$$

$$c = 2 \cdot \text{atan2} (\sqrt{a}, \sqrt{1-a}) \quad (2.7)$$

$$d = R \cdot c \quad (2.8)$$

Keterangan:

R = jari-jari bumi sebesar 6371(km)

$\Delta lat$  = besaran perubahan *latitude*

$\Delta long$  = besaran perubahan *longitude*

c = kalkulasi perpotongan sumbu

d = jarak (km)

#### 2.2.7. Geolocation

*Geolocation* adalah identifikasi lokasi geografis suatu objek pada dunia nyata. *Geolocation* mempunyai kaitan erat dengan *positioning*, perbedaannya adalah *geolocation* lebih spesifik dalam menentukan sebuah lokasi (misalnya alamat jalan) dibandingkan dengan *positioning* yang hanya mencakup sekumpulan

koordinat geografis. Suatu lokasi geografis mengandung nilai Latitude dan longitude [ Nur Fajruddin, 2017]

#### 2.2.8. RAD

*RAD* merupakan metode pengembangan sistem yang memiliki keunggulan karena tahapan yang singkat dan cepat seperti tahapan *requirements planning* untuk mengidentifikasi tujuan dari aplikasi atau sistem, tahapan *RAD design workshop* (pemodelan) untuk membangun tampilan visual desain dan alur kerja pengguna dan tahapan implementasi untuk pembangunan sistem dan pengujian. Sehingga dengan tahapan-tahapan tersebut, penerapan metode *RAD* sangat tepat dan sesuai dalam pembangunan sistem berbasis website [ Arsia, 2017]

RAD mempunyai 3 tahapan:

a. requirement planning

Pada tahap ini yang dilakukan adalah melakukan pengumpulan dan identifikasi data yang nantinya dibutuhkan untuk pembuatan sistem, data bisa diperoleh dengan komunikasi antara user dan analyst.

b. design system

Pada tahap secara rinci dengan membuat rancangan. Perancangan bisa dibuat dengan model dfd ataupun *Usecase Diagram*.

c. *implementation*

Tahap implementasi merupakan tahap pengkonversian desain sistem menjadi sebuah program utuh dan dilakukan oleh seorang programmer menggunakan bahasa pemrograman tertentu.

#### 2.2.9. UML

UML adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *Class* dan *operation* dalam konsep

dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasabahasa berorientasi objek seperti C++, Java, C# atau VB.NET.

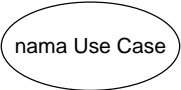
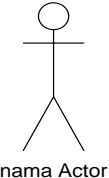
Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan syntax/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai *Diagram* piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML syntax mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT (Object Modeling Technique), dan Ivar Jacobson OOSE (Object-Oriented Software Engineering). (Darwiyanti Dan Romi 2016)


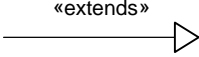

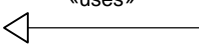
#### 2.2.10. Use Case Diagram

Menurut A.S. Rosa dan M. Shalahudin dalam bukunya “Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek” *Use Case* atau diagram *use case* merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat [roman. 2016].

*Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kuat, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Simbol-simbol yang ada pada diagram *use case* dapat dilihat pada tabel 2.1 :

**Tabel 2. 2 Simbol Use Case Diagram**

Simbol	Nama Simbol	Deskripsi
	<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 frase nama <i>use case</i>
	<i>Aktor</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 aktor adalah gambar orang, biasanya dinyatakan menggunakan kata benda di awal frase nama actor
	<i>Association</i> Asosiasi	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan actor
	<i>Extend</i> Ekstensi	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau 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
	<i>Generalization</i> Generalisasi	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya
	<i>Include / Uses</i> Menggunakan	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya / sebagai syarat dijalankan <i>use case</i> ini Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> : a. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan. b. <i>Include</i> berarti <i>use case</i> yang tambahan



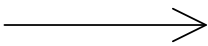

		akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan.
		Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.

### 2.2.11. Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode operasi [roman. 2016]. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Simbol-simbol yang ada pada *class diagram* dapat dilihat pada tabel 2.2 :

**Tabel 2. 3 Simbol Class Diagram**






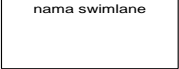
Simbol	Nama Simbol	Deskripsi
<div style="border: 1px solid black; padding: 2px; display: inline-block;">           nama kelas            -attribute            +operasi()         </div>	<i>Class</i> Kelas	Kelas pada struktur system
nama interface ○ —	<i>Interface</i> Antarmuka	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
—————	<i>Association</i> Asosiasi	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
—————>	<i>Directed Association</i> Asosiasi berarah	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity
—————▷	<i>Generalization</i> Generalisasi	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus)

	<i>Dependency</i> Kebergantungan	Relasi antarkelas dengan makna kebergantungan antarkelas.
	<i>Aggregation</i> Agregresi	Relasi antarkelas dengan makna semua-bagian ( <i>whole-part</i> )

### 2.2.12. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *work-flow* (aliran kerja) atau aktivitas dari sebuah sistem atau sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Simbol-simbol *activity diagram* dapat dilihat pada tabel 2.3 :

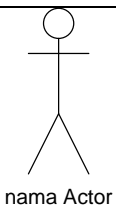

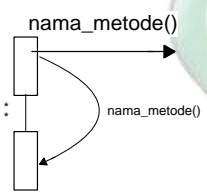
**Tabel 2. 4 Simbol Activity Diagram**

Simbol	Nama Simbol	Deskripsi
	<i>Initial State</i> Status Awal	Status awal aktifitas sistem, sebuah diagram aktifitas memiliki sebuah status awal
	<i>Action State</i> Aktifitas	Aktifitas yang dilakukan sistem, aktifitas biasanya diawali dengan kata kerja
	Decision Percabangan	Asosiasi percabangan dimana jika ada pilihan aktifitas lebih dari satu
	Join Penggabungan	Asosiasi penggabungan lebih dari satu aktifitas digabungkan menjadi satu
	Final State Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktifitas memiliki sebuah status akhir
	Swimlane	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi

### 2.2.13. Sequence Diagram

*Sequence Diagram* digunakan untuk menggambarkan perilaku pada sebuah skenario. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. Simbol-simbol *sequence diagram* dapat dilihat pada tabel 2.4 :

**Tabel 2. 5 Simbol Sequence Diagram**










Simbol	Nama Simbol	Deskripsi
	<i>Actor</i> Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor
	<i>Lifeline</i> Garis hidup	Menyatakan kehidupan suatu objek
	Message (Call) Pesan tipe Call	Menyatakan suatu objek memanggil operasi-metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarah kepada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi

#### 2.2.14. Flowchart

*Flowchart* adalah diagram alir yang menggambarkan urutan logika dari suatu prosedur pemecahan masalah. Bagan alir program merupakan bagan yang

menjelaskan secara rinci langkah-langkah dari proses program [roman. 2016]]. Simbol-simbol dari *flowchart* dapat dilihat pada tabel 2.5 :

**Tabel 2. 6 Simbol Flowchart**


Simbol	Nama Simbol	Fungsi
	<i>Terminator</i>	Permulaan / akhir program
	<i>Flow Line</i> Garis Alir	Arah aliran program
	<i>Preparation</i>	Proses inisialisasi / pemberian harga awal
	<i>Process</i>	Proses perhitungan / proses pengolahan data
	<i>Input / Output Data</i>	Proses input / output data, parameter, informasi
	<i>Predefined Process</i> Sub Program	Permulaan sub program / proses menjalankan sub program
	Decision Keputusan	Perbandingan pernyataan, penyeleksian data yang memberikan pihak untuk langkah selanjutnya
	On Page Connector	Penghubung bagian-bagian flowchart pada satu halaman
	Off Page Connector	Penghubung bagian-bagian flowchart pada halaman berbeda

#### 2.2.15. ERD (Entity Relationship Diagram)

ERD adalah diagram yang berisi komponen-komponen entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang mempresentasikan seluruh fakta yang ditinjau sehingga dapat diketahui hubungan antara *entity-entity* yang ada dengan atribut-atributnya [wijaya, 2017]. Simbol-simbol ERD dapat dilihat pada tabel 2.6 :

**Tabel 2. 7 Simbol ERD**



Simbol / Notasi	Nama Simbol	Deskripsi
	<i>Entity</i> Entitas	Entitas adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai
	<i>Relation</i> Relasi	Relasi menunjukkan adanya hubungan diantara sejumlah entitas yang berbeda
	<i>Atribut</i> Kelengkapan	Atribut berfungsi mendeskripsikan karakter entitas (atribut yang berfungsi sebagai <i>key</i> diberi garis bawah)
	<i>Line / Connector</i> Garis / Koneksi	Garis sebagai penghubung antara relasi dan entitas atau relasi dan entitas dengan atribut

#### 2.2.16. Basis Data (*Database*)

Basis data terdiri dari dua kata yaitu basis yang berarti gudang atau tempat bersarang/berkumpul, dan data yang berarti representasi fakta dalam dunia nyata yang mewakili suatu objek seperti manusia, barang, hewan, peristiwa, konsep, dll, yang diwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya [Azizah, 2016].

*Database* merupakan kumpulan informasi yang disimpan dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut [Sumadya, 2016].

Dari beberapa pernyataan di atas, maka database dapat diartikan sebagai kumpulan data yang disimpan dalam komputer untuk membuat, mengakses, mengubah, menghapus, dan memproses data yang tersimpan dalam komputer untuk mendukung aplikasi dalam system.

#### 2.2.17. WEB

*World Wide Web* atau yang biasa kita kenal dengan sebutan WEB adalah salah satu layanan yang didapat oleh pemakai computer yang terhubung ke internet. Dalam web ini menyediakan beragam informasi yang dibutuhkan oleh

pemakai computer yang terhubung ke internet. Website atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, atau gabungan dari semuanya itu baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing konten atau isi web itu sendiri dihubungkan dengan jaringan-jaringan halaman. (Afriliani, Fitriani 2017)

#### 2.2.18. MySQL

SQL (*Structured Query Language*) adalah bahasa yang berisi perintah-perintah untuk memanipulasi *database*, mulai dari melakukan perintah *select* untuk menampilkan *database*, *insert* untuk menambahkan isi ke dalam *database*, *update* untuk mengubah isi *database*, dan *delete* untuk menghapus isi dari *database* [ahmad, 2016].

Sedangkan MySQL berawal dari proyek yang dipelopori oleh Michael Widenius dan David Axmark pada tahun 1994. Proyek ini didasari karena mereka ingin membuat sistem *database* yang murah dimana pada saat itu *database* yang paling handal adalah *oracle* yang bersifat komersil dan mahal harganya namun dapat menguasai pasar.

#### 2.2.19. PHP

PHP adalah sebuah bahasa scripting yang dibundel dengan HTML, yang dijalankan di sisi server. Bahasa ini memungkinkan para pembuat aplikasi web menyajikan halaman HTML dinamis dan interaktif dengan cepat dan mudah, yang dihasilkan server. PHP juga dimaksudkan untuk mengganti teknologi lama seperti CGI (Common Gateway Interface). PHP biasa berinteraksi dengan hampir semua teknologi web yang sudah ada. Developer biasa menulis sebuah program PHP yang mengeksekusi suatu program CGI di server web lain. Fleksibilitas ini amat bermanfaat bagi pemilik situs-situs web yang besar dan sibuk, karena pemilik masih biasaa mempergunakan aplikasi-aplikasi yang sudah terlanjur dibuat di masa lalu dengan CGI, ISAP, atau dengan script seperti perl, Awk atau python selama proses migrasi ke aplikasi baru yang dibuat dengan PHP. Ini mempermudah dan memperhalus peralihan antara teknologi lama den teknologi

baru. Saat server melayani permintaan dari browser web akan suatu dokumen, server sebenarnya hanya mengambil suatu file di dalam disk dan melakukan beberapa pekerjaan untuk transmisi seperti menambahkan informasi tipe dokumen, merubah formatnya agar biasa dikirim menggunakan HTTP, yang mengirimkan semuanya ke browser. Browser web menerima file HTML dan menampilkannya ke layar monitor client. Sumbernya tetap berada di server dan di sana ia tidak berubah sama sekali.(Yusman Dan Maryati. 2017).

#### 2.2.20. XAMPP

XAMPP merupakan paket PHP dan MYSQL berbasis open source yang dapat digunakan sebagai tool pembantu pengembangan aplikasi berbasis PHP. Fungsi XAMPP adalah sebagai web server yang berdiri sendiri (localhost) yang terdiri dari program Apache, MySQL, PHP dan Pearl [wijaya, 2016].

