

BAB II

DASAR TEORI

2.1 Tinjauan Pustaka

Penelitian sebelumnya mengenai sistem informasi penilaian seminar kerja proyek dan sidang tugas akhir yang dilakukan oleh Ade Yuliana dan Rabella Rigustama Az Zahra pada tahun 2021. Penelitian ini dilakukan karena terdapat beberapa permasalahan yaitu pada proses penilaian masih bersifat semi manual. Selain itu, proses pengolahan dan perhitungan nilai seminar masih menggunakan cara manual dan sebagian sudah ada yang terkomputerisasi menggunakan *Microsoft Excel*. Oleh karena itu, menyebabkan proses perhitungan nilai yang cukup memakan waktu, dokumen proposal dan tugas akhir yang sulit untuk diarsipkan. Sehingga untuk mengatasi permasalahan tersebut diperlukan suatu sistem berbasis *web* yang dibuat untuk memudahkan dan mempercepat proses penilaian seminar kerja proyek dan sidang tugas akhir. Berdasarkan hasil pengujian *black box*, semua fungsi yang ada dalam sistem ini dapat berjalan sesuai dengan yang diharapkan, dan berdasarkan pengujian *user acceptance testing* sistem ini terbukti dapat diterima dengan baik oleh calon admin dan calon pengguna dengan nilai presentase 84.6 % ^[2].

Penelitian selanjutnya pernah dilakukan oleh Achmad Syarif pada tahun 2020 mengenai sistem informasi penilaian prestasi mahasiswa program studi sekretari berbasis *web*. Penelitian ini didasarkan pada beberapa permasalahan yang terjadi seperti proses penilaian prestasi yang masih dilakukan secara manual yaitu mahasiswa menyerahkan data kegiatan prestasi beserta bukti dokumen ke Bagian Layanan Prestasi Mahasiswa yang kemudian dihitung poin pencapaiannya, sehingga sering terjadi kendala seperti perbedaan perhitungan poin yang telah diperoleh, informasi yang tidak terpusat, bukti dokumen tidak dicadangkan dalam bentuk digital yang akan mengakibatkan bukti dokumen tersebut hilang atau rusak. Oleh sebab itu, perlu adanya pembaharuan untuk mengatasi permasalahan tersebut yaitu dengan dibuatnya sistem ini yang akan memudahkan mahasiswa dan Bagian Layanan Prestasi Mahasiswa dalam perhitungan poin kegiatan prestasi mahasiswa dan data dapat tersimpan dalam bentuk digital. Kesimpulan dari sistem ini adalah telah memenuhi kebutuhan penggunaannya dengan baik ^[3].

Penelitian lainnya pernah dilakukan oleh Anita Yusmar dan Geovanne Farell pada tahun 2020 mengenai rancang bangun sistem

informasi pada penilaian kinerja guru berbasis *web* dengan menggunakan *framework codeigniter* 3. Penelitian ini didasarkan pada beberapa permasalahan yang terjadi seperti proses penilaian kinerja guru yang masih dilakukan secara konvensional yaitu menggunakan *Microsoft Excel*, sehingga menyulitkan dalam proses perhitungan nilai kinerja, perekapan nilai, dan data guru serta pembuatan laporan penilaian kinerja guru. Selain itu, terdapat masalah pada pengarsipan data penilaian kinerja yang disimpan dalam berkas yang berbeda - beda sehingga jumlah berkas yang harus disimpan setiap tahunnya banyak. Oleh karena itu, untuk mengatasi permasalahan tersebut dibuat sistem ini. Hasil dari sistem ini yaitu sudah berhasil menyelesaikan permasalahan dengan baik dan memudahkan proses pendataan, proses perhitungan, perekapan dokumen pendukung serta pembuatan laporan ^[4].

Penelitian selanjutnya juga pernah dilakukan oleh Arsito Ari Trikuncoro, Eko Haryono, dan Paulus Hartanto pada tahun 2020 mengenai sistem informasi penilaian kinerja tenaga pendidik dan tenaga kependidikan berbasis *web*. Penelitian ini dilakukan karena terdapat beberapa permasalahan yaitu proses penilaian tenaga pendidik dan kependidikan dilakukan oleh pimpinan dan koordinator cabang dengan mengisi *form* penilaian pada *Microsoft Excel*, kemudian hasilnya dikirimkan kepada bagian kepegawaian melalui *e-mail*. Bagian kepegawaian tersebut akan merekap data *Microsoft Excel* yang masuk dan hasil akhir akan dilaporkan kepada ketua STEKOM. Hal ini menyebabkan proses penilaian membutuhkan waktu relatif lama dan cenderung kurang efisien. Oleh sebab itu, perlu dibangun sistem informasi ini agar memudahkan dalam proses penilaian dengan mudah dan cepat. Kesimpulan sistem pada penelitian ini yaitu telah memenuhi kebutuhan pengguna dengan baik seperti penilaian kinerja sudah bisa dilakukan secara *online* sehingga lebih cepat dan efisien serta hasil dari penilaian kinerja dari masing - masing kampus cabang STEKOM sudah terekap secara otomatis sehingga memudahkan bagian kepegawaian untuk mendapatkan data hasil penilaian kinerja ^[5].

Penelitian lain juga pernah dilakukan oleh Silvia Ratna pada tahun 2020 mengenai sistem informasi penilaian proposal skripsi berbasis *web*. Penelitian ini dilakukan karena terdapat beberapa permasalahan yaitu pelaksanaan evaluasi sidang skripsi masih manual dimana dalam proses penilaian menggunakan media kertas yang kemudian direkap ke dalam *spreadsheet* lalu hasilnya diumumkan kepada mahasiswa. Hal ini kurang efektif apabila banyak mahasiswa yang mengambil mata kuliah skripsi.

Oleh karena itu, perlu adanya pembaharuan untuk memudahkan penilaian, menjaga konsistensi nilai yang diberikan, dan mempercepat pembuatan laporan. Kesimpulan dari penelitian ini adalah sistem ini dapat berjalan sesuai rancangan dan telah memenuhi kebutuhan pengguna dengan baik ^[6].

Penelitian yang dibuat oleh penulis memiliki perbedaan dengan penelitian yang pernah dilakukan sebelumnya, antara lain proses penilaian akan diurutkan berdasarkan nilai tertinggi, menyediakan fasilitas *download* sertifikat bagi pemenang duta kampus, membuat laporan penilaian dari juri dan panitia untuk masing-masing peserta duta kampus, dan menyediakan histori data peserta duta kampus sebelumnya. Sistem yang akan dibangun diharapkan dapat digunakan untuk mempermudah dalam serangkaian proses duta kampus dari awal pendaftaran peserta sampai dengan pengumuman pemenang duta kampus secara *online*. Sistem ini yang akan dibangun adalah berbasis *website* dengan menggunakan bahasa pemrograman PHP dan untuk penyimpanan *database* nya menggunakan *MySQL*.

2.2 Landasan Teori

2.2.1 Sistem

Suatu kesatuan yang terdiri dari dua atau lebih komponen atau sub - sistem yang saling berinteraksi untuk mencapai sebuah tujuan tertentu disebut dengan sistem ^[7].

Sistem merupakan suatu bentuk keterpaduan antara satu komponen dengan komponen lainnya karena sistem tersebut mempunyai tujuan yang berbeda - beda untuk setiap kasus yang terjadi pada sistem tersebut. Ada dua kelompok pendekatan dalam definisi sistem, yaitu kelompok yang menekankan prosedur / metode dan kelompok yang menekankan unsur / komponen. Pendekatan yang menekankan prosedur yaitu sistem sebagai jaringan prosedur yang saling berhubungan, berkumpul bersama - sama untuk melakukan kegiatan atau mencapai tujuan tertentu. Sedangkan pendekatan yang menekankan unsur / komponen yaitu sistem sebagai kumpulan dari elemen - elemen yang bekerja sama untuk mencapai suatu tujuan tertentu ^[8].

2.2.2 Informasi

Data yang diolah menjadi bentuk jamak dari bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya disebut dengan informasi ^[9]. Selain pengertian tersebut, informasi juga diartikan sebagai

data yang diubah menjadi format yang lebih berguna dan bermakna bagi pihak yang menerimanya. Informasi dikatakan berguna apabila dapat membantu dalam membuat keputusan yang berkualitas dan memberikan pedoman untuk mencapai tujuan organisasi ^[10]. Informasi dapat dikatakan berkualitas apabila informasi tersebut tergantung dari 3 hal yaitu ^[11] :

1. Akurat (*accurate*)
Artinya informasinya akurat dan tidak boleh menyesatkan. Akurat juga berarti bahwa informasi harus secara jelas mencerminkan maksudnya. Informasi harus akurat, karena biasanya dapat terjadi gangguan (*noise*) dari sumber ke penerima informasi dan dapat mengubah atau merusak informasi.
2. Tepat waktu (*timeline*)
Artinya informasi yang sampai pada penerima tidak boleh ditunda. Informasi adalah dasar pengambilan keputusan, jadi informasi yang sudah ketinggalan zaman tidak ada nilainya. Apabila pengambilan keputusan terlambat dapat berakibat fatal bagi organisasi. Saat ini, informasi menjadi mahal karena perlu dikirim atau diambil dengan cepat, sehingga penerimaan, pemrosesan, dan pengiriman informasi memerlukan teknologi terkini.
3. Relevan (*relevance*)
Artinya informasi ini memiliki manfaat dalam penggunaannya. Jika kebutuhan informasi ini untuk organisasi maka informasi tersebut harus sesuai dengan kebutuhan informasi berbagai tingkatan atau bagian yang ada dalam organisasi tersebut.

2.2.3 Sistem Informasi

Suatu sistem yang dibuat oleh manusia yang terdiri dari komponen - komponen dalam organisasi yang digunakan untuk mencapai satu tujuan yaitu menyajikan informasi merupakan definisi dari sistem informasi ^[12].

Suatu sistem yang menyediakan informasi untuk manajemen dalam mengambil keputusan dan juga untuk menjalankan operasional perusahaan, dimana sistem tersebut merupakan kombinasi dari orang - orang teknologi informasi dari prosedur - prosedur yang terorganisasi disebut dengan sistem informasi ^[2]. Adapaun komponen fisik yang menyusun sistem informasi terdiri dari ^[8] :

1. Perangkat keras (*hardware*), yaitu termasuk perangkat fisik untuk komputer dan printer.
2. Perangkat lunak (*software*), yaitu serangkaian instruksi yang memungkinkan perangkat keras untuk dapat memproses data.

3. Prosedur (*procedure*), yaitu seperangkat aturan yang digunakan untuk melakukan pemrosesan data dan produksi keluaran yang diinginkan.
4. Manusia (*brainware*), yaitu semua pihak yang bertanggung jawab atas pengembangan, pemrosesan, dan penggunaan keluaran sistem informasi.
5. Basis Data (*database*), yaitu kumpulan tabel, hubungan, dan hal-hal yang terkait dengan penyimpanan data.
6. Jaringan komputer dan komunikasi data, yaitu sistem konektivitas yang memungkinkan banyak pengguna untuk berbagi atau mengakses sumber daya.

2.2.4 Duta Kampus

Duta kampus merupakan perwakilan mahasiswa di perguruan tinggi yang dipilih sesuai dengan kriteria yang ditentukan, berperan sebagai *icon* perguruan tinggi dengan kecerdasan intelektual, kepemimpinan yang unggul, dan bertanggung jawab yang kredibel, terutama di lingkungan internal dan eksternal perguruan tinggi. Serta mendorong mahasiswa lainnya agar menjadi mahasiswa yang unggul, berprestasi dan memiliki keterampilan memasuki dunia kerja dengan keadaan yang profesional^[13]. Secara umum, terdapat beberapa komponen dalam penilaian duta kampus yaitu kecerdasan, *public speaking*, performa / keterampilan, kemampuan menjawab pertanyaan, dan kemampuan berbahasa.

2.2.5 Rekayasa Web

Rekayasa *web* adalah aplikasi yang menggunakan pendekatan sistematis, disiplin, dan terukur untuk pengembangan, pengoperasian, dan pemeliharaan aplikasi berbasis *web* (*web - based application*). Rekayasa *web* adalah sub - disiplin rekayasa perangkat lunak yang membantu menyediakan cara untuk merancang, mengembangkan, memelihara, dan mengintegrasikan aplikasi *web*^[14].

1. PHP (*Hypertext Preprocessor*)

Adalah suatu bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin dan dapat dipahami oleh komputer *server - side* yang dapat ditambahkan ke HTML. Sifat *server - side* ini berarti kode program dijalankan di *server* dan hasilnya dikirim ke *browser*^[15]. Kelebihan dari PHP adalah bahasa pemrograman yang bersifat *open source* yang artinya gratis untuk

digunakan, mudah digunakan, sederhana, mudah untuk dipahami, dan mudah dipelajari.

2. HTML (*Hypertext Markup Language*)

Adalah bahasa yang mengontrol bagaimana konten situs *web* ditampilkan. HTML memiliki tag yang memberikan informasi tentang sifat dan struktur konten, serta referensi ke gambar dan media lainnya ^[16]. Oleh karena itu, HTML berperan dalam mendesain struktur halaman *website* yang mengatur setiap elemen *website* sesuai dengan *layout* yang diinginkan.

3. *Web Server*

Adalah sebuah aplikasi *server* yang berfungsi mendapat permintaan dari klien yang dikenal dengan *web browser* dan kemudian dan kemudian mengirimkan kembali hasilnya dalam bentuk halaman - halaman *web* melalui protokol HTTP. Apache, NGINX, IIS, dan *LiteSpeed Web Server* merupakan *web server* yang paling banyak digunakan untuk melakukan pengembangan aplikasi *web* ^[17].

4. *Web Browser*

Suatu program yang digunakan untuk menjelajahi internet atau untuk mencari sebuah informasi dari suatu *web* yang tersimpan di dalam komputer dan memiliki tujuan utama yaitu untuk membawa sumber informasi kepada pengguna merupakan definisi dari *web browser*. Ada berbagai macam perangkat *web browser* seperti *Mozilla Firefox*, *Google Chrome*, *Opera*, *Safari*, dan *Internet Explorer* ^[17].

2.2.6 Pengujian *Black Box*

Metode pengujian *Black Box* adalah metode pengujian sistem yang akan digunakan, dimana metode ini digunakan untuk menguji sebuah fungsionalitas dari sistem tanpa harus memperhatikan atau melihat detail struktur pengkodeannya. Metode ini hanya memeriksa nilai keluaran berdasarkan nilai masukkan masing-masing fungsionalitas dari sistem.

2.2.7 Pemrograman Berorientasi Objek

Pemrograman berorientasi objek (*Object Oriented Programming*) merupakan paradigma pemrograman yang berorientasikan kepada objek semua data dan fungsi dalam paradigma ini dibungkus dalam kelas - kelas atau objek - objek ^[18]. Berbeda dengan pemrograman terstruktur atau pemrograman berorientasi prosedur (POP) dimana setiap objek menerima pesan atau data, kemudian memprosesnya dan mengirimkannya ke objek

lain. Kelebihan dari OOP dibandingkan dengan prosedur adalah sebagai berikut :

1. Lebih terstruktur dan mudah dibaca.
2. Kelas-kelas dapat digunakan kembali pada proyek yang lain (*reuse*).
3. Pemetaan masalah jadi lebih mudah sehingga mudah untuk membuat solusinya.
4. Pembatasan akses terhadap suatu fungsi dapat meningkatkan keamanan *code*.
5. Interaksi antara *code* lebih terasa.

Pemrograman berorientasi objek (OOP) lebih berfokus kepada manipulasi objek. Adapun konsep dari OOP yaitu ^[19] :

1. Kelas (*class*)
Adalah kumpulan objek dengan karakteristik yang sama. Kelas memiliki sifat (atribut), kelakuan (operasi / metode), hubungan (*relationship*), dan makna. Secara teknis, kelas adalah struktur khusus dalam pengembangan perangkat lunak. Kelas merupakan bentuk struktural dari kode program yang menggunakan metodologi berorientasi objek.
2. Objek (*object*)
Adalah entitas yang memiliki operasi (kelakuan) yang dapat menyimpan informasi (status) dan dapat menerapkan atau mempengaruhi status objek. Objek adalah komponen perangkat lunak yang berisi sekumpulan *variable* dan *method* yang berhubungan.
3. Metode (*method*)
Metode atau operasi yang digunakan untuk memanipulasi objek itu sendiri. Operasi atau metode adalah fungsi atau transformasi yang dapat dilakukan pada objek atau oleh objek.
4. Atribut (*attribute*)
Atribut dari sebuah kelas adalah *variable* global yang dimiliki oleh kelas tersebut. Atribut dapat berupa nilai atau elemen - elemen data yang dimiliki oleh objek dari kelas objek.
5. Abstraksi (*abstraction*)
Prinsipnya adalah merepresentasikan dunia nyata yang kompleks dalam format model yang sederhana dengan mengabaikan aspek lain yang tidak sesuai dengan permasalahan.

6. Enkapsulasi (*encapsulation*)
Pembungkusan atribut data dan layanan (operasi) milik objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya.
7. Pewarisan (*inheritance*)
Mekanisme yang memungkinkan suatu objek untuk mewarisi beberapa atau semua definisinya dan objek lain sebagai bagian dari dirinya sendiri.
8. Antarmuka (*interface*)
Antarmuka atau *interface* sangat mirip dengan kelas, tetapi tanpa atribut kelas dan metode yang dideklarasikan tanpa isi. Antarmuka biasanya digunakan oleh kelas yang lain tidak mengakses langsung ke suatu kelas, mengakses antarmukanya.
9. *Reusability*
Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut.
10. Generalisasi dan Spesialisasi
Menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek khusus.
11. Komunikasi Antar Objek
Komunikasi antar objek dilakukan melalui pesan yang dikirim dan satu objek ke objek lainnya.
12. Polimorfisme (*polymorphism*)
Kemampuan suatu objek untuk digunakan di banyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.
13. *Package*
Sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompokkan kelas - kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam *package* yang berbeda disebut dengan *package*.

2.2.8 *Flowchart*

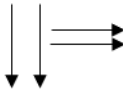

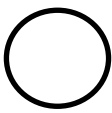
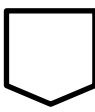
Flowchart adalah representasi secara simbolik dari algoritma atau prosedur yang digunakan untuk memecahkan masalah. *Flowchart* memudahkan pengguna untuk melihat apa yang terlupakan dalam analisis masalah. *Flowchart* juga berfungsi sebagai alat komunikasi antara *programmer* yang bekerja pada tim proyek. *Flowchart* adalah salah satu

cara untuk merepresentasikan suatu algoritma ^[20]. Simbol - simbol *flowchart* dibagi menjadi 3 (tiga) kelompok, yakni sebagai berikut ^[12] :

1. *Flow Direction Symbols*

Simbol ini disebut juga *connecting line*, yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Berikut Tabel 2.1 adalah simbol *flow direction* yang ada pada *flowchart*.



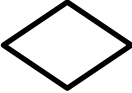



Tabel 2. 1 Simbol *Flow Direction*

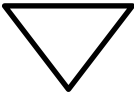

No.	Simbol	Nama	Keterangan
1.		<i>Arus / flow</i>	Untuk menyatakan jalannya arus suatu proses.
2.		<i>Communication Link</i>	Untuk menyatakan bahwa adanya transisi suatu data atau informasi dari satu lokasi ke lokasi lainnya.
3.		<i>Connector</i>	Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman atau lembar yang sama.
4.		<i>Offline Connector</i>	Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman atau lembar yang berbeda.

2. *Processing Symbols*

Simbol yang menunjukkan jenis operasi pengolahan dalam suatu proses / prosedur. Berikut Tabel 2.2 adalah simbol *process* yang ada pada *flowchart*.

Tabel 2. 2 Simbol Proses




No.	Simbol	Nama	Keterangan
1.		<i>Process</i>	Untuk menyatakan suatu proses perilaku (proses) yang dilakukan oleh komputer.
2.		Manual	Untuk menyatakan suatu perilaku (proses) yang tidak dilakukan oleh komputer.
3.		<i>Decision / logika</i>	Untuk menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban yaitu ya atau tidak.
4.		<i>Predefined Proses</i>	Untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
5.		<i>Terminal</i>	Untuk menyatakan permulaan atau akhir suatu program.
6.		<i>Keying Operation</i>	Untuk menyatakan segala jenis operasi yang diproses dengan menggunakan suatu




			mesin yang mempunyai <i>keyboard</i> .
7.		<i>Offline Storage</i>	Untuk menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu.
8.		<i>Manual Input</i>	Untuk memasukkan data secara manual dengan menggunakan <i>online keyboard</i> .

3. *Input - Output Symbols*

Simbol yang menunjukkan jenis peralatan yang digunakan sebagai media *input* atau *output*. Berikut Tabel 2.3 adalah simbol *input - output* yang ada pada *flowchart*.

Tabel 2. 3 Simbol *Input - Output*

No.	Simbol	Nama	Keterangan
1.		<i>Input / Output</i>	Untuk menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.
2.		<i>Punched Card</i>	Untuk menyatakan <i>input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu.
3.		<i>Magnetic-tape Unit</i>	Untuk menyatakan <i>input</i> berasal dari pita <i>magnetic</i> atau <i>output</i> disimpan ke pita <i>magnetic</i> .

4.		<i>Disk Storage</i>	Untuk menyatakan <i>input</i> berasal dari disk atau <i>output</i> disimpan ke <i>disk</i> .
5.		<i>Document</i>	Untuk mencetak laporan ke printer.
6.		<i>Display</i>	Untuk menyatakan peralatan output yang digunakan berupa layar (video, komputer).

2.2.9 Unified Modelling Language (UML)


Salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, melakukan analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek adalah *unified modelling language* ^[19].




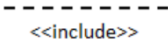
Unified Modelling Language adalah seperangkat aturan yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak berdasarkan obyek ^[10]. Alat bantu yang dipakai dalam perancangan berorientasi objek berbasis UML yakni sebagai berikut :

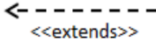
1. *Use Case Diagram*

Use case atau diagram *use case* merupakan pemodelan dari perilaku (*behavior*) sistem informasi yang akan dibuat. Secara umum, *use case* menggambarkan interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat dan digunakan untuk mengetahui fungsi yang ada di dalam sebuah sistem informasi serta siapa saja yang berhak menggunakan fungsi - fungsi tersebut. Berikut Tabel 2.4 adalah simbol - simbol yang ada pada *use case diagram* ^[19].

Tabel 2. 4 Simbol *Use Case Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>Use Case</i>	Menggambarkan fungsionalitas yang disediakan oleh sistem sebagai unit - unit yang saling bertukar pesan

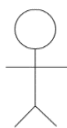

			antar unit atau aktor, yang dinyatakan dengan menggunakan kata kerja di awal <i>frase</i> nama <i>use case</i> .
2.		<i>Actor</i>	Merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
3.		<i>Association</i>	Merupakan komunikasi antara aktor dan <i>use case</i> yang berpartisipasi atau berinteraksi dengan <i>use case</i> .
4.		<i>Generalization</i>	Merupakan hubungan generalisasi dan spesialisasi (umum - khusus) antara dua <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
5.		<i>Include</i>	Merupakan relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai

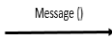


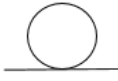
			syarat dijalankan <i>use case</i> ini.
6.		<i>Extends</i>	Merupakan relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.

2. *Sequence Diagram*

Sequence diagram merupakan gambaran dari perilaku objek dalam *use case* dengan menjelaskan waktu hidup objek dan *message* yang dikirim dan diterima antar objek. Oleh karena itu, untuk menggambar *sequence diagram* maka harus diketahui metode - metode kelas yang dipakai dalam objek itu. Berikut Tabel 2.5 adalah simbol - simbol yang ada pada *sequence diagram* ^[19].

Tabel 2. 5 Simbol *Sequence Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
2.		<i>Lifeline</i>	Menyatakan kehidupan suatu objek.

3.		<i>Message</i>	Digunakan untuk berkomunikasi antar objek dengan menjelaskan aksinya.
4.		<i>Boundary</i>	Menggambarkan sebuah <i>form</i> .
5.		<i>Control</i>	Simbol yang mengatur aliran dari informasi pada sebuah skenario.
6.		<i>Entitas</i>	Simbol yang bertanggung jawab untuk menyimpan data atau informasi.

3. *Class Diagram*

Class diagram atau diagram kelas menggambarkan struktur sistem dalam hal mendefinisikan kelas yang akan dibuat untuk membangun sistem. Kelas memiliki atribut dan metode atau operasi.

- Atribut merupakan variabel - variabel yang dimiliki oleh suatu kelas.
- Metode atau operasi adalah fungsi - fungsi yang dimiliki oleh suatu kelas.

Class diagram dibuat agar *programmer* membuat kelas sesuai dengan desain diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Berikut Tabel 2.6 adalah simbol - simbol yang ada pada *class diagram* ^[19].

Tabel 2. 6 Simbol *Class Diagram*

No.	Simbol	Nama	Keterangan			
1.	<table border="1" style="margin-left: 20px;"> <tr> <td>Nama_kelas</td> </tr> <tr> <td>+atribut</td> </tr> <tr> <td>+operasi()</td> </tr> </table>	Nama_kelas	+atribut	+operasi()	<i>Class</i>	Kelas pada struktur sistem.
Nama_kelas						
+atribut						
+operasi()						

2.		Antarmuka / <i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.		Asosiasi / <i>Association</i>	Relasi antar kelas dengan makna umum, yang biasanya disertai dengan <i>multiplicity</i> .
4.		Asosiasi Berarah / <i>Directed Association</i>	Relasi antar kelas dengan makna kelas satu digunakan oleh kelas yang lain, yang biasanya disertai dengan <i>multiplicity</i> .
5.		Generalisasi	Relasi antar kelas dengan makna <i>generalisasi - spesialisasi</i> (umum - khusus).
6.		Kebergantungan / <i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.
7.		Agregasi / <i>Aggregation</i>	Relasi antar kelas dengan makna semua bagian (<i>whole part</i>).

2.2.10 Basis Data (*Database*)

Basis data (*database*) merupakan kumpulan *record* data operasional lengkap untuk suatu organisasi atau perusahaan yang diatur menggunakan metode tertentu dan disimpan secara konsisten untuk

memenuhi kebutuhan informasi pengguna ^[21]. Pada hakekatnya basis data merupakan media penyimpanan data, dapat diakses dengan mudah dan cepat ^[19].

Database Management System (DBMS) merupakan kumpulan terintegrasi dari basis data dan program komputer (utilitas) yang digunakan untuk mengakses dan memelihara basis data. Program - program ini menyediakan berbagai fungsi untuk memasukkan, melacak, memodifikasi data dalam *database*, mendefinisikan data baru, dan memproses data menjadi informasi yang dibutuhkan ^[12].

SQL (*Structured Query Language*) adalah sebuah bahasa yang digunakan untuk mengakses data dalam basis data relasional. Terdapat 3 (tiga) jenis *query database* pada SQL yaitu ^[22] :

1. DDL (*Data Definition Language*)

Adalah kumpulan perintah SQL yang menjelaskan desain seluruh *database* dan juga digunakan untuk membuat, mengubah, maupun menghapus struktur atau definisi tipe data untuk objek yang ada di *database*. Adapun *query* DDL yaitu :

- a. *Create* : Digunakan untuk pembuatan *table* dan *database*
- b. *Drop* : Digunakan untuk penghapusan *table* dan *database*
- c. *Alter* : Digunakan untuk perubahan struktur *table* yang dibuat, baik menambah *field* (*add*), mengganti nama *field* (*change*), ataupun menamakannya kembali (*rename*) serta menghapus (*drop*).

2. DML (*Data Manipulation Language*)

Adalah metode *query* yang digunakan ketika DDL telah dibuat dan digunakan untuk memanipulasi *database*. Adapun *query* yang termasuk di dalamnya adalah :

- a. *Insert* : Digunakan untuk penginputan data pada *table database*
- b. *Update* : Digunakan untuk perubahan data pada *table database*
- c. *Delete* : Digunakan untuk penghapusan data pada *table database*.

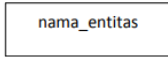
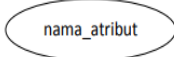
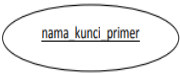
3. DCL (*Data Control Language*)

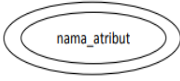

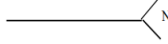
Adalah perintah SQL yang memberikan hak untuk melakukan pekerjaan yang terkait dengan *database* dan hak untuk memanipulasi objek dalam *database*. Contoh *query* DCL yaitu :

- a. *Grant* : Digunakan untuk memberikan hak akses khusus kepada user untuk melakukan aksi tertentu pada *object* tertentu di *database*.
- b. *Revoke* : Digunakan untuk menarik kembali atau membatalkan hak akses khusus yang sebelumnya telah diberikan melalui perintah *grant*.

Entity Relationship Diagram (ERD) adalah model informasi untuk menjelaskan hubungan antara data dan basis data, yang dijelaskan oleh grafik dan notasi model data konseptual. ERD merupakan teknik yang digunakan untuk menggambarkan data yang dibutuhkan oleh suatu organisasi. Pemodelan seperti analisis sistem, yaitu tahapan analisa persyaratan yang dibutuhkan dari proyek pengembangan sistem ^[21]. Berikut adalah simbol - simbol yang digunakan pada ERD dengan notasi Chen ^[19] :

Tabel 2. 7 Simbol ERD

No	Notasi	Nama	Keterangan
1.		Entitas	Merupakan data inti yang akan disimpan. Penamaan entitas biasanya menggunakan kata benda dan belum merupakan nama tabel.
2.		Atribut	Merupakan <i>field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
3.		Atribut kunci primer	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan.

4.		Atribut multi nilai / <i>multivalued</i>	<i>Field</i> 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 dimana dikedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian.