



BAB II

TINJAUAN PUSTAKA DAN

LANDASAN TEORI

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Pada tahun 2021 penelitian pertama dilakukan oleh Aviv Mahmudi pada STIE YPPI Rembang dengan judul “Implementasi Sistem Informasi Wisuda STIE YPPI Rembang Berbasis Website”. Kurang efektif dan efisiennya pelayanan pendaftaran wisuda di STIE YPPI Rembang disebabkan oleh pengolahan data peserta wisuda masih dilakukan secara manual. Berdasarkan masalah tersebut penulis membuat penelitian dengan tujuan untuk membuat perancangan dan melakukan implementasi sistem informasi pendaftaran wisuda. Agar meningkatkan kualitas pelayanan di STIE YPPI Rembang. Sistem tersebut dibangun dengan bahasa pemrograman PHP dan *database* MySQL. Selain itu, pendekatan sistem yang digunakan dalam perancangan dan implementasi tersebut adalah dengan menggunakan pendekatan terstruktur. Sedangkan untuk metode pengembangan sistem menggunakan metode SDLC (Software Development Life Cycle). Perbedaan penelitian ini dengan penelitian sebelumnya yaitu, aplikasi tidak dimulai dari proses validasi berkas ketika mahasiswa tersebut sudah dinyatakan lulus TA. Selain itu, aplikasi ini tidak dilengkapi fitur pemberian nomor tempat duduk secara otomatis dan fitur absensi kehadiran pada saat pelaksanaan wisuda berlangsung[4].

Penelitian lain juga dilakukan oleh Arizal dan Arrisa Nurul Puteri pada Universitas Fajar pada tahun 2020 dengan judul penelitian yaitu sistem informasi manajemen wisuda berbasis *website* dengan metode *waterfall*. Sistem tersebut dirancang menggunakan *use case* diagram dan *rich picture diagram* untuk menjelaskan interaksi antara pengguna dengan sistem. Hasil dari penelitian tersebut adalah sebuah sistem informasi manajemen wisuda berbasis *website* yang menjadikan proses manajemen wisuda lebih efektif, efisien, dan sistematis. Berbagai fitur dalam sistem informasi ini memudahkan calon wisudawan, staf prodi, dan panitia wisuda dalam melakukan manajemen data dan informasi wisuda. Perbedaan penelitian ini dengan penelitian sebelumnya yaitu, aplikasi tidak dimulai dari proses validasi berkas ketika mahasiswa tersebut sudah dinyatakan lulus TA. Selain itu, aplikasi ini tidak dilengkapi fitur pemberian nomor tempat duduk secara otomatis [5].

Pada tahun 2020 Taufiq Hidayat dan Mahmudin Muttaqin juga melakukan penelitian dengan judul “Sistem Informasi Pendaftaran dan Pembayaran Wisuda Online menggunakan Black Box Testing pada Universitas Islam Syekh Yusuf”. Pada saat ini mahasiswa Universitas Islam Syekh Yusuf yang hendak mendaftar wisuda harus mengisi kelengkapan data – data yang dipersyaratkan secara manual yaitu dengan cara datang ke kampus dan mengisi data-data calon wisudawan. Dengan adanya permasalahan tersebut maka peneliti melakukan penelitian untuk membangun sistem informasi wisuda agar calon wisudawan dan wisudawati bisa melakukan pendaftaran wisuda secara *online* dan mengisi serta melengkapi persyaratan wisuda secara *online*. Sistem tersebut dibangun dengan bahasa pemrograman PHP dan *database* MySQL. Perbedaan penelitian ini dengan penelitian sebelumnya yaitu, aplikasi tidak dimulai dari proses validasi berkas ketika mahasiswa tersebut sudah dinyatakan lulus TA. Selain itu, aplikasi ini tidak dilengkapi fitur pemberian nomor tempat duduk secara otomatis dan fitur absensi kehadiran pada saat pelaksanaan wisuda berlangsung [6].

Penelitian serupa juga dilakukan oleh Adam William Yordan Sihotang dan Elmor Benedict Wagiu pada tahun 2020 dengan judul “Perancangan Sistem Informasi Calon Wisudawan Universitas Advent” dengan menggunakan metode SDLC (Software Development Life Cycle). Sistem tersebut dirancang menggunakan *use case* diagram dan *Activity diagram* untuk menjelaskan interaksi antara pengguna dengan sistem. Sistem tersebut dibangun dengan bahasa pemrograman PHP dan *database* MySQL. Pada saat ini pendaftaran wisuda pada Universitas Advent masih bersifat manual, yaitu dengan cara mengambil formulir serta mengisi formulir tersebut. Kurang efisiennya proses pendaftaran wisuda yang berlangsung pada Universitas Advent membuat peneliti membangun sistem untuk menunjang agar calon wisudawan dan wisudawati mudah dalam melakukan pendaftaran wisuda secara *online*. Hasil dari penelitian tersebut adalah sebuah sistem informasi manajemen wisuda berbasis *website* yang menjadikan proses manajemen wisuda lebih efektif, efisien, dan sistematis[7].

Selanjutnya Eldy Walf Rumaf juga melakukan penelitian yang serupa pada tahun 2019 dengan judul penelitian yaitu “Sistem Informasi Pendaftaran Wisuda pada STIMK Tidore Mandiri”. Sistem tersebut dibangun dengan bahasa pemrograman PHP dan *database* MySQL. Proses pendaftaran calon peserta wisuda di STMIK Tidore Mandiri saat

ini masih dilakukan dengan tulis tangan yaitu mengisi formulir yang disediakan oleh petugas pendaftaran di bagian program studi. Berdasarkan permasalahan tersebut, maka peneliti perlu membuat aplikasi sistem informasi pendaftaran wisuda yang nantinya dapat digunakan sebagai media pendaftaran calon peserta wisuda dan pengontrolan pemasukan file foto dan file skripsi/tugas akhir sebagai syarat dokumen peserta wisuda di STMIK Tidore Mandiri. Perbedaan penelitian ini dengan penelitian sebelumnya yaitu, aplikasi tidak dimulai dari proses validasi berkas ketika mahasiswa tersebut sudah dinyatakan lulus TA. Selain itu, aplikasi ini tidak dilengkapi fitur pemberian nomor tempat duduk secara otomatis dan fitur absensi kehadiran pada saat pelaksanaan wisuda berlangsung[8].

Penelitian yang akan dibuat penulis tentu akan berbeda dengan yang ada sebelumnya. Perbedaan sistem ini dengan sistem-sistem sebelumnya yaitu adanya proses validasi pada tahap seleksi oleh bagian-bagian terkait pendaftaran wisuda. Selain itu, adanya fitur pemberian nomor tempat duduk, sehingga memudahkan bagian BAAK. Adanya fitur presensi, juga menjadi perbedaan, sehingga nantinya diharapkan dapat memudahkan petugas presensi mendata wisudawan yang hadir ataupun tidak.

2.2 Landasan Teori

Landasan teori berisi hal-hal atau teori-teori yang berkaitan dengan permasalahan dan ruang lingkup permasalahan sebagai landasan dalam pembuatan laporan ini.

2.2.1 Sistem Informasi

Sistem adalah kumpulan orang yang saling bekerja sama dengan ketentuan-ketentuan aturan yang sistematis dan terstruktur untuk membentuk satu kesatuan melaksanakan suatu fungsi untuk mencapai tujuan. Sistem memiliki beberapa karakteristik atau sifat yang terdiri dari komponen sistem, batasan sistem, lingkungan luar sistem, penghubung sistem, masukan sistem, keluaran sistem, pengolahan sistem dan sasaran sistem[9].

Informasi adalah sekumpulan fakta-fakta yang telah diolah menjadi bentuk sekumpulan data, sehingga dapat menjadi lebih berguna dan dapat digunakan oleh siapa saja yang membutuhkan data-data tersebut sebagai pengetahuan ataupun dapat digunakan dalam pengambilan keputusan[10].

Sistem informasi adalah suatu sistem yang menyediakan informasi untuk manajemen pengambilan keputusan / kebijakan dan menjalankan operasional dari kombinasi orang-orang, teknologi informasi dan prosedur-prosedur yang terorganisasi atau sistem informasi diartikan sebagai kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi untuk mendukung operasi dan manajemen. Sedangkan dalam arti luas, sistem informasi diartikan sebagai sistem yang sering digunakan untuk interaksi antara orang, proses, algoritmik, data dan teknologi. Sistem informasi di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan[9].

Sistem Informasi terdiri dari beberapa komponen yang saling berhubungan satu sama lain membentuk satu kesatuan untuk mencapai tujuan. Komponen-komponen sistem informasi yaitu :

1. Komponen *Input*

Input merupakan data yang masuk ke dalam sistem informasi yang dapat berupa dokumen

2. Komponen Model

Komponen model terdiri kombinasi prosedur, logika, dan model matematika yang memproses data yang tersimpan di dalam basis data dengan cara yang sudah di tentukan untuk menghasilkan keluaran yang diinginkan.

3. Komponen *Output*

Komponen *output* berupa informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta pemakai sistem.

4. Komponen Teknologi

Teknologi merupakan alat dalam sistem informasi, teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan output dan membantu pengendalian sistem.

5. Komponen Basis Data

Merupakan kumpulan data yang saling berhubungan yang tersimpan di dalam komputer dengan menggunakan *software database*.

6. Komponen Kontrol

Merupakan komponen pengendalian yang dirancang untuk menanggulangi gangguan terhadap sistem informasi.

2.2.2 Pendaftaran Wisuda

Kamus Besar Bahasa Indonesia menjelaskan pendaftaran adalah proses, cara, perbuatan mendaftar (mendaftarkan). Sedangkan wisuda adalah peresmian atau pelantikan yang dilakukan dengan upacara khidmat. Berdasarkan uraian tersebut, maka dapat disimpulkan bahwa pengertian wisuda adalah acara peresmian atau pelantikan yang dilakukan dengan dengan cara khidmat dan mendapatkan ijazah. Setiap mahasiswa yang telah menyelesaikan tugas akhir dan telah memenuhi seluruh persyaratan memiliki hak penuh sebagai mahasiswa, yaitu berhak untuk mengikuti proses wisuda[7].

2.2.3 Rekayasa Perangkat Lunak

Perangkat lunak adalah kumpulan data elektronik yang disimpan dan dikelola oleh perangkat komputer. Instruksi atau program yang selanjutnya akan menjalankan perintah tertentu termasuk dalam data elektronik. Perangkat lunak merupakan komponen sistem komputer yang tidak memiliki bentuk fisik dan ditempatkan pada komputer atau laptop agar dapat berfungsi. Setiap bagian dari perangkat lunak dirancang untuk membantu manusia melakukan aktivitas seperti menghitung dan memproses angka, membuat dokumen tertulis, laporan, dan makalah, dan memproses gambar, serta menghasilkan presentasi. Rekayasa Perangkat Lunak (*Software Engineering*) merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin[11].

2.2.4 Rekayasa Web

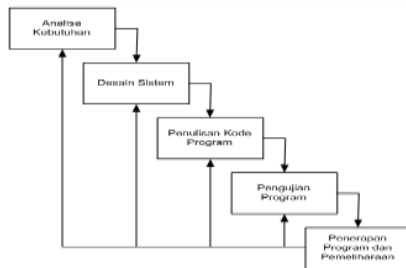
Rekayasa Web adalah proses yang digunakan untuk menciptakan aplikasi web yang berkualitas tinggi. Rekayasa web mengadaptasi rekayasa perangkat lunak dalam hal konsep dasar yang menekankan pada aktifitas teknis dan manajemen. Namun demikian adaptasi tidak secara utuh, tapi dengan perubahan dan penyesuaian. Rekayasa web gabungan antara web publishing (suatu konsep yang berasal dari printed publishing) dan aktifitas rekayasa perangkat lunak. Dikatakan demikian karena desain sebuah aplikasi web menekankan pada desain grafis, desain informasi, teori hypertext, desain sistem dan pemrograman[12].

2.2.5 Metode Pengembangan Sistem

Metode penelitian yang digunakan adalah metode *waterfall*.

Waterfall atau air terjun adalah model yang dikembangkan untuk pengembangan perangkat lunak, dan membuat perangkat lunak. Model *Waterfall* merupakan metode pengembangan sistem dimana antar satu fase ke fase yang lain dilakukan secara berurutan. Metode ini berkembang secara sistematis dari satu tahap ke tahap lain dalam mode seperti air terjun. Metode ini mengusulkan sebuah pendekatan kepada pengembangan software yang sistematis dan sekuensial yang mulai dari tingkat kemajuan sistem pada seluruh analisis, desain, kode, pengujian dan pemeliharaan[13].

Model waterfall terdiri dari 5 tahapan, yaitu Analisa Kebutuhan, Desain Sistem, Penulisan Kode Program, Pengujian Program, Penerapan Program dan Pemeliharaan. Model waterfall dapat dilihat pada Gambar dibawah ini[13].



Gambar 2. 1 Pemodelan waterfall

Penjelasan tahapan-tahapan dari metode *waterfall* [14]:

1. Analisa Kebutuhan (*Analysis*)

Langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data pada tahap ini bisa melakukan sebuah penelitian, wawancara atau *study* literatur. Seorang sistem analis akan menggali informasi sebanyak banyaknya dari *user* sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas-tugas yang diinginkan oleh *user* tersebut. Tahapan ini akan menghasilkan dokumen *user requirment* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan sistem. Dokumen inilah yang akan menjadi acuan sistem analisis untuk menterjemahkan kedalam bahasa pemrograman.

2. Desain Sistem (*Design System*)

Pada Tahap *Design System* ini akan dibentuk suatu arsitektur

sistem berdasarkan persyaratan yang telah ditetapkan. Selain itu juga, dilakukan identifikasi dan penggambaran terhadap abstraksi dasar sistem perangkat lunak beserta hubungan-hubungannya.

3. Penulisan Kode Program (*Coding*)

Coding merupakan penerjemah *design* dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh *programmer* yang akan menterjemahkan transaksi yang diminta oleh *user*. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah kita buat tadi.

4. Pengujian Program (*Testing*)

Tahapan ini bisa dikatakan final dalam pembuatan sebuah sistem. Setelah melakukan analisa, *design* dan pengkodean maka sistem yang sudah jadi akan digunakan oleh *user*.

5. Penerapan Program dan Pemeliharaan (*Maintenance*)

Perangkat lunak yang sudah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (peripheral atau dengan sistem operasi yang baru) atau karena pelanggan membutuhkan perkembangan fungsional.

2.2.6 Metode Pengujian Sistem

Metode pengujian sistem yang digunakan adalah dengan metode *black box testing*. Metode *black box testing* merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, penguji (*tester*) dapat mendefinisikan kumpulan kondisi dan melakukan pengesanan pada spesifikasi fungsional program[15].

Ciri-ciri *black box testing* :

1. *Black box testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*.
2. *Black box testing* bukan teknik alternatif daripada *white box testing*. Lebih dari itu, *black box testing* merupakan pendekatan pelengkap dalam mencakup *error* dengan kelas yang berbeda dari metode *white box testing*.
3. *Black box testing* melakukan pengujian tanpa pengetahuan detail struktur internal dari sistem atau komponen yang dites.

Kategori *error* yang akan diketahui melalui *black box testing* :

1. Fungsi yang hilang atau tak benar.

2. *Error* dari antarmuka.
3. *Error* dari struktur data atau akses *eksternal database*.
4. *Error* dari kinerja atau tingkah laku.
5. *Error* dari inialisasi dan terminasi

2.2.7 Pemograman Berorientasi Objek

Metodologi berorientasi objek adalah strategi pembangunan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis[16].

Metodologi berorientasi objek banyak dipilih karena metodologi lama banyak menimbulkan masalah seperti adanya kesulitan pada saat mentransformasi hasil dari satu tahap pengembangan ke tahap berikutnya, misalnya pada metode pendekatan tersruktur, jenis aplikasi yang dikembangkan saat ini berbeda dengan masa lalu.

Keuntungan menggunakan metodologi berorientasi objek adalah sebagai berikut:

1. Meningkatkan produktifitas
 Karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai diulang kembali untuk masalah lainnya yang melibatkan objek tersebut (*reusable*).
2. Kecepatan pengembangan
 Karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengkodean.
3. Kemudahan pemeliharaan
 Karena dengan model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola mungkin sering berubah-ubah.
4. Adanya konsistensi
 Karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.
5. Meningkatkan kualitas perangkat lunak
 Karena pendekatan pengembangan lebih dekat dengan dunia nyata dan adanya konsistensi pada saat pengembangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.

Berikut ini adalah beberapa konsep dasar yang harus dipahami

tentang metodologi berorientasi objek[16] :

1. Kelas (*class*)

Kelas adalah kumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statik dan himpunan objek yang sama yang mungkin lahir atau diciptakan dan kelas tersebut. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (operasi/metode), hubungan (*relationship*) dan arti. Suatu kelas dapat diturunkan dan kelas yang lain, dimana atribut dan kelas semula dapat diwariskan ke kelas yang baru.

2. Objek (*object*)

Objek adalah abstraksi dan sesuatu yang mewakili dunia nyata seperti benda, manusia, suatu organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan.

3. Metode (*method*)

Operasi atau metode atau *method* pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi. Metode atau operasi yang berfungsi untuk memanipulasi objek itu sendiri. Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek.

4. Atribut (*attribute*)

Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat berupa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek. Atribut dipunyai secara individual oleh sebuah objek, misalnya berat, jenis, nama dan sebagainya. Atribut sebaliknya bersifat privat untuk menjaga konsep enkapsulasi.

5. Abstraksi (*abstract*)

Prinsip untuk mempresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

6. Enkapsulasi (*encapsulation*)

Pembungkusan atribut data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya.

7. Pewarisan (*inheritance*)

Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dan dirinya.

8. Antarmuka (*interface*)



Antarmuka atau *interface* sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah *interface* dapat diimplementasikan oleh kelas lain.




Pada pemrograman berorientasi objek, UML digunakan untuk pemodelan sistem. UML adalah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan *artifacts* (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, *artifact* tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya.

2.2.8 Use Case Diagram

Use case adalah abstraksi dari interaksi antara sistem dan aktor. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah system dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. *Use case* merupakan konstruksi untuk mendeskripsikan bagaimana sistem akan terlihat di mata *user*. Sedangkan use case diagram memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan client[17].

Tabel 2. 1 Simbol dan Fungsi Use Case Diagram


No	Simbol	Fungsi
1.	 <i>Actor</i>	Segala sesuatu yang berinteraksi dengan sistem aplikasi komputer. Jadi actor ini bisa berupa orang, perangkat keras atau mungkin juga obyek lain dalam sistem yang sama.
2.	 <i>Include</i>	Menspesifikasikan bahwa perilaku <i>use case</i> merupakan bagian dari <i>use case</i> lain.




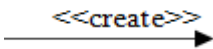
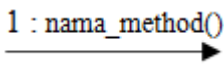
3.	 Association	Menggambarkan navigasi antar class, berupa banyak obyek lain yang berhubungan dengan satu obyek, dan apakah suatu class menjadi bagian dari class lainnya.
4.	 System Boundary	System Boundary yaitu batasan sebuah sistem.
5.	 Use Case	<i>Use case</i> menjelaskan urutan kegiatan yang dilakukan aktor dan sistem untuk mencapai suatu tujuan tertentu. Walaupun menjelaskan kegiatan, use case hanya menjelaskan apa yang dilakukan oleh aktor dan sistem, bukan bagaimana aktor dan sistem melakukan kegiatan.



2.2.9 Sequence Diagram

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima oleh objek. Oleh karena itu untuk menggambar diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram *sequence* juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Berikut simbol-simbol yang ada pada diagram *sequence*[17].

Tabel 2. 2 Simbol dan Fungsi Sequence Diagram

No	Simbol	Fungsi
1.	 Actor	<i>Actor</i> yaitu Orang, poses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.






2.	<p><i>Lifeline</i></p> 	<i>Lifeline</i> yaitu Menyatakan kehidupan suatu objek.
3.	<p><i>Object</i></p> 	<i>Object</i> yaitu Menyatakan objek yang berinteraksi pesan.
4.	<p>Waktu aktif</p> 	Waktu aktif yaitu Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.
5.	<p>Pesan tipe <i>create</i></p> 	Suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6.	<p>Pesan tipe <i>call</i></p> 	Suatu objek memanggil operasi atau metode yang ada pada objek lain atau dirinya sendiri.


7.	<p>Pesan tipe <i>send</i></p> <p>1 : masukan</p> 	<p>Suatu objek mengirimkan data atau masukan atau informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>
8.	<p>Pesan tipe <i>return</i></p> <p>1 : keluaran</p> 	<p>Suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu.</p>

2.2.10 Class Diagram

Class Diagram atau Diagram Kelas merupakan diagram yang memodelkan sekumpulan kelas, *interface*, kolaborasi, dan relasinya. Diagram kelas digambarkan dengan bentuk kotak. Simbol dan keterangan class diagram seperti pada tabel 2.4.

Tabel 2. 3 Simbol dan Fungsi Class Diagram

No	Simbol	Fungsi
1.	 <i>Generalization</i>	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus).
2.	 <i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3.	 <i>Class</i>	Kelas pada struktur sistem.
4.	 <i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
5.	 <i>Dependency</i>	Relasi antarkelas dengan makna kebergantungan antarkelas.

6.	 <i>Association</i>	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity.
----	---	---

2.2.11 Basis Data

Basis Data atau *Database* adalah himpunan kelompok data yang saling berkaitan. Data tersebut diorganisasikan sedemikian rupa agar tidak terjadi duplikasi data yang tidak perlu, sehingga dapat diolah atau dieksplorasi secara cepat dan mudah untuk menghasilkan informasi [18].

2.2.12 Flowchart

Flowchart merupakan gambar atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta instruksinya. Gambaran ini dinyatakan dengan simbol. Dengan demikian setiap simbol menggambarkan proses tertentu. Sedangkan hubungan antar proses digambarkan dengan garis penghubung. *Flowchart* ini merupakan langkah awal pembuatan program. Dengan adanya flowchart urutan poses kegiatan menjadi lebih jelas. Jika ada penambahan proses maka dapat dilakukan lebih mudah. Setelah *flowchart* selesai disusun, selanjutnya programmer menerjemahkannya ke bentuk program dengan bahasa pemrograman [19].

Simbol yang dipakai dalam *flowchart* dibagi menjadi 3 kelompok, yaitu :

1. *Flow direction symbols*

Digunakan untuk menghubungkan simbol satu dengan yang lain (*connecting line*).

2. *Processing symbols*




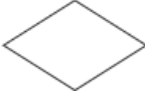


Menunjukkan jenis operasi pengolahan dalam suatu proses atau prosedur.

3. *Input / Output symbols*

Menunjukkan jenis peralatan yang digunakan sebagai media input atau output.

Flowchart disusun dengan simbol-simbol. Simbol ini dipakai sebagai alat bantu menggambarkan proses di dalam program. Simbol-simbol yang dipakai antara lain :

Tabel 2. 4 Simbol dan Fungsi Flowchart

No	Simbol	Nama dan Fungsi
1.		<i>Flow Direction Symbol</i> yaitu simbol yang digunakan untuk menghubungkan antar simbol
2.		<i>Process Symbol</i> yaitu simbol yang menunjukkan pengolahan yang dilakukan oleh komputer.
3.		<i>Manual Operation Symbol</i> yaitu simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer.
4.		<i>Decision Symbol</i> yaitu simbol yang menunjukan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban ya tau tidak.
5.		<i>Terminal Symbol</i> yaitu symbol yang menyatakan permulaan atau akhir suatu program.
6.		<i>Input atau Output Symbol</i> yaitu simbol yang menyatukan proses input atau output tanpa tergantung jenis peralatannya.