



# **BAB II**

# **TINJAUAN PUSTAKA DAN**

# **LANDASAN TEORI**

## **BAB II**

### **DASAR TEORI**

#### **2.1. Tinjauan Pustaka**

Penelitian sebelumnya yang dilakukan oleh Jimmi Hendrik P. Sitorus pada tahun 2022, dengan judul “ Sistem Informasi Buku Induk Siswa SMK Negeri 1 Siantar Berbasis *Web*”. Sistem ini digunakan untuk mempermudah pengelolaan dan penyimpanan data induk siswa. Data induk siswa yang dimaksud meliputi biodata siswa, data guru, dan cetak data siswa [1].

Penelitian serupa yang dilakukan oleh Yoga Handoko Agustin pada tahun 2022, dengan judul “Perancangan Sistem Informasi Akademik Berbasis Web di SMK YABP 1 Garut”. Sistem ini bertujuan untuk membantu memberikan informasi terkait data guru, data siswa, pendataan kelas, jadwal kelas dan nilai untuk pengajar, siswa dan orang tua/wali siswa. Data yang dikelola pada sistem akademik ini meliputi data guru, data siswa, data kelas, jadwal kelas, dan nilai [2].

Penelitian serupa yang dilakukan oleh Andrey Gustaph Setiawan pada tahun 2022, dengan judul “Perancangan Sistem Informasi Akademik Berbasis *Web Mobile* pada Pondok Pesantren Dar El Amir dengan Metode *Waterfall*”. Sistem ini digunakan untuk mempermudah pengelolaan data menjadi lebih cepat, efektif, dan efisien serta membantu dalam proses penyebaran informasi yang berhubungan dengan kegiatan akademik pondok pesantren. Data yang dikelola pada sistem ini meliputi data santri, data guru, data pegawai, data wali kelas, data rapor [3].

Penelitian serupa dilakukan oleh Firio Amazon pada tahun 2021, dengan judul “Rancang Bangun Sistem Informasi Akademik Fakultas Matematika Dan Ilmu Pengetahuan Alam Berbasis Website”. Sistem ini digunakan sebagai media informasi yang memfasilitasi dosen dan mahasiswa dalam mengetahui kegiatan akademik yang dilaksanakannya. Sistem ini mengelola data akademik yang meliputi pengelolaan data mata kuliah, nilai, dan pengisian krs. [4]

Perbedaan penelitian ini dengan peneliti-peneliti sebelumnya yaitu tidak ada yang menjelaskan data mengenai nilai ekstrakurikuler, sehingga hal ini dirasa masih belum cukup detail menjelaskan mengenai data induk

siswa. Selain itu dari sisi tampilan, beberapa sistem tidak menerapkan konsep responsive pada tampilan. Sehingga apabila data diakses melalui *smartphone* tidak dapat menyesuaikan resolusi layar. Penelitian yang peneliti buat, juga menerapkan konsep pendekatan *object oriented programming*, sehingga penelitian ini lebih terstruktur.

## **2.2. Landasan Teori**

Dalam penelitian ini diperlukan adanya teori yang mendasar dalam menunjang proses penelitian, berikut antara lain :

### **2.2.1. Sistem Informasi**

Sistem informasi adalah suatu sistem didalam organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi, dan menyediakan pihak luar tertentu dengan laporan – laporan yang diperlukan. Sistem informasi merupakan salah satu hal terpenting dalam suatu perusahaan, dengan adanya sistem informasi maka organisasi atau perusahaan dapat menjamin kualitas informasi yang disajikan dan dapat mengambil keputusan berdasarkan informasi yang cepat, tepat dan akurat [5].

Adapun definisi sistem informasi menurut beberapa ahli yaitu [6]:

1. Menurut Mc leod, sistem informasi merupakan sistem yang mempunyai kemampuan untuk mengumpulkan informasi dari semua sumber dan menggunakan berbagai media untuk menampilkan informasi.
2. Menurut O'Brien, sistem informasi adalah suatu kombinasi teratur apapun dari *people* (orang), *hardware* (perangkat keras), *software* (piranti lunak), *computer networks* dan *data communications* (jaringan komunikasi), dan *database* (basis data) yang mengumpulkan, mengubah dan menyebarkan informasi di dalam suatu bentuk organisasi.

### **2.2.2. Metode Pengujian Sistem**

Pengujian sistem yang digunakan peneliti adalah *Black-Box Testing*. *Black box* testing adalah menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat

mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian black box testing harus dibuat dengan kasus benar dan kasus salah [7].

### **2.2.3. Pemrograman Berorientasi Objek**

Metodelogi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan informasi yang diberlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metodologi berorientasi objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Metodologi berorientasi objek merupakan rangkaian aktivitas analisis orientasi objek, pemrograman berorientasi objek, dan pengujian berorientasi objek [8]. Beberapa konsep dasar yang harus dipahami dalam pemrograman berorientasi objek berikut ini :

#### **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 dari kelas tersebut. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (operasi/method), hubungan (relationship), dan arti. Suatu kelas dapat diwariskan ke kelas yang baru.

#### **2. Objek (object)**

Objek adalah abstraksi dan sesuatu yang mewakili dunia nyata seperti benda, manusia, satuan 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 sebaiknya bersifat privat untuk menjaga konsep enkapsulasi.

5. Abstraksi (abstraction)

Prinsip untuk merepresentasikan 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. Sebuah kelas dapat mengimplementasikan lebih dari satu antarmuka dimana kelas ini akan mendeklarasikan metode pada antarmuka yang dibutuhkan oleh kelas itu sekaligus mendefinisikan isinya pada kode program kelas itu.

9. Reusability

Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut.

10. Generalisasi dan Spesifikasi

Menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus.

#### 11. Komunikasi Antar Obyek

Komunikasi antar-objek dilakukan lewat pesan (message) yang dikirim dan satu objek ke objek lainnya.

#### 12. Polimorpisme (polymorphism)

Kemampuan suatu objek untuk digunakan di banyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.

#### 13. Package

Package adalah sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam package yang berbeda.

### 2.2.4. Basis Data

Basis data atau database adalah sekumpulan relasi data logika, dan deskripsi dari data yang dirancang untuk memenuhi kebutuhan informasi organisasi. Basis data atau database adalah kumpulan data yang mewakili berbagai macam entitas dan hubungannya yang dapat digunakan secara bersamaan oleh banyak pengguna dan dirancang untuk memenuhi kebutuhan informasi organisasi [9]. Basis data memungkinkan tempat penyimpanan datayangbesar dan dapat digunakan secara bersamaan oleh banyak departemen dan pengguna. Database mewakili entitas, atribut, dan hubungan logis antara entitas. Basis data terdiri dari kumpulan data yang terorganisir, relasi antar data, dan objektifnya. Objektif utama adalah kecepatan dan kemudahan berinteraksi dengan data yang dikelola atau diolah. Selain itu terdapat pengertian bahwa basis data adalah sekumpulan data persisten yang digunakan oleh aplikasi sistem dari perusahaan.

#### A. DBMS (Database Management System)

DBMS (Data Base Management System) adalah koleksi terpadu dari aplikasi program (system software) yang dapat digunakan dalam melakukan definisi membuat melakukan akses dan memelihara database [10]. DBMS menyediakan berbagai fasilitas yaitu :

1. DDL ( Data Definition Language) merupakan perintahperintah yang biasa digunakan administrator database untuk

mendefinisikan skema dan subskema database. Perintah yang termasuk di dalamnya yaitu [11]:

- a. **CREATE** : Digunakan untuk membuat, termasuk diantaranya membuat database dan tabel baru.
  - b. **ALTER** : Digunakan untuk mengubah struktur tabel yang telah dibuat.
  - c. **DROP** : Digunakan untuk menghapus database dan tabel.
2. **DML (Data Manipulation Language)** merupakan merupakan perintah-perintah yang memungkinkan pengguna melakukan akses dan manipulasi data sebagaimana yang telah diorganisasikan sebelumnya dalam model data yang tepat, (**Data Manipulation Language**) digunakan untuk memanipulasi database yang telah didefinisikan dengan **DDL**. Perintah yang termasuk **DML**:
- a. **INSERT** : Digunakan untuk menyisipkan atau memasukan dalam tabel
  - b. **SELECT**: Untuk mengambil data atau menampilkan data dari suatu tabel atau beberapa tabel.
  - c. **UPDATE** : Digunakan untuk memperbaharui data lama menjadi data terkini
  - d. **DELETE**: Digunakan untuk menghapus data dari tabel

#### B. **SQL (Structured Query Language)**




**SQL (Stuctured Query Language)** adalah bahasa yang digunakan untuk mengelola data pada **RDBMS**, **SQL** awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus [12]. **MySQL** merupakan basis data yang dikembangkan dari bahasa **SQL (structure query language)**. **MySQL** dapat dikatakan sebagai relational database management system (**RDBMS**), yaitu hubungan antar table yang berisi data data pada suatu database dengan demikian dapat.

#### C. **Entity Relationship Diagram (ERD)**


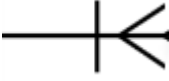

**Entity Relationship Diagram** atau **ERD** adalah sebuah diagram struktural yang digunakan untuk merancang sebuah database. Sebuah **ERD** mendeskripsikan data yang akan disimpan dalam sebuah sistem

maupun batasannya. Komponen utama yang terdapat di dalam sebuah ERD adalah entity set, relationship set, dan juga constraints [13]. Berikut ini symbol-simbol yang digunakan dalam ERD dapat dilihat pada Tabel 2.4.

**Tabel 2.1** Simbol dan Fungsi ERD [14]

No	Gambar Simbol	Nama	Keterangan
1.		<i>Relasi</i>	Hubungan yang terjadi antara 1 entitas atau lebih yang tidak mempunyai fisik tetapi hanya sebagai konseptual. Dan untuk mengetahui jenis hubungan yang ada antara 2 file.
2.		<i>Atribut</i>	Atribut ialah karakteristik dari entitas atau relasi yang menyediakan penjelasan detail tentang entitas atau relasi tersebut. Berfungsi untuk memperjelas atribut yang dimiliki oleh sebuah entitas. Atribut memiliki bentuk lingkaran lebih tepatnya elips..
3.		<i>One to One</i>	Alur memiliki fungsi untuk menghubungkan atribut dengan entitas dan entitas dengan relasi. Dan berbentuk garis.



4.		<i>Realization</i>	<i>One to One</i> , digunakan untuk menghubungkan antar entitas dengan hubungan satu ke satu
5.		<i>One to Many</i>	<i>One to Many</i> atau <i>Many to One</i> , digunakan untuk menghubungkan antar entitas dengan hubungan satu ke banyak atau sebaliknya
6.		<i>(Entity)</i>	Entitas ialah suatu objek yang dapat dibedakan dengan objek lainnya. Entitas berfungsi untuk memberikan identitas pada entitas yang memiliki label dan nama. Entitas memiliki bentuk persegi panjang.

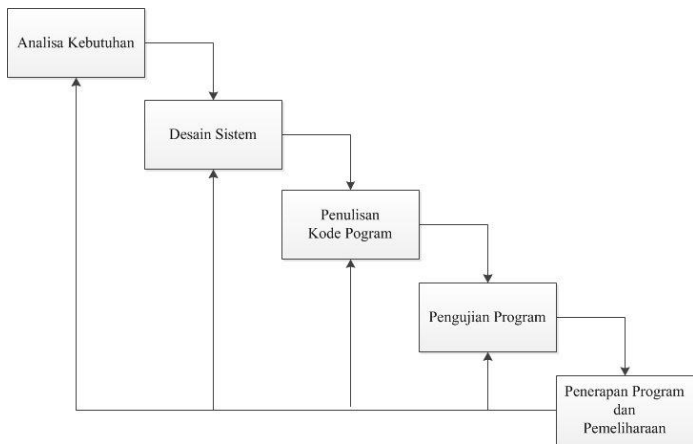
### 2.2.5. Rekayasa Perangkat Lunak

Perangkat lunak adalah sebuah perintah program dalam sebuah computer, yang apabila di eksekusi oleh user akan memberikan fungsi dan untuk bekerja seperti yang diharapkan oleh user. Rekayasa Perangkat Lunak (RPL) merupakan proses kegiatan perangkat lunak guna mengembangkan, memelihara, dan membangun kembali dengan menggunakan prinsip rekayasa untuk menghasilkan perangkat lunak yang dapat bekerja lebih efektif dan efisien untuk user [15].

### 2.2.6. *Waterfall*

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode *Waterfall*. Metode *waterfall* merupakan metode yang memiliki ciri khas dari pengerjaannya yaitu setiap fase harus diselesaikan terlebih dahulu sebelum melanjutkan ke fase selanjutnya yang artinya metode *waterfall* ini fokus terhadap masing-masing fase sehingga lebih

maksimal [3].



**Gambar 2. 1** Metode Waterfall Menurut Sommerville

Berikut tahapan yang dilakukan sesuai dengan pengembangan sistem pada metode waterfall.

### 1. Analisis kebutuhan

Analisis kebutuhan merupakan tahap pertama yang menjadi dasar proses pembuatan perangkat lunak maupun sistem. Dalam tahapan ini dilakukan analisa dan pengumpulan kebutuhan sistem yang akan dibangun. Analisa pada tahap ini didasarkan pada data yang diperoleh pada tahap pengumpulan data dengan metode observasi. Kelancaran proses pembuatan sistem secara keseluruhan sangat tergantung pada hasil analisa kebutuhan ini.

### 2. Desain sistem

Desain merupakan tahapan penyusunan rancangan yang matang mengenai setiap proses, data, aliran proses dan hubungan antara data dalam sistem sesuai dengan hasil tahapan analisis kebutuhan. Dalam tahapan ini disusun desain interface, arsitektur sistem dan database ERD. Dokumentasi yang dihasilkan dari tahap desain sistem ini antara lain : use case diagram, diagram aktivitas dan sequence diagram, Data Flow Diagram DFD dan Entity Relationship Diagram ERD.

### 3. Coding

*Coding* merupakan tahapan penerjemahan desain sistem yang telah dibuat ke dalam bentuk perintah-perintah atau kode program yang dimengerti komputer dengan menggunakan bahasa pemrograman, middleware dan database tertentu.

#### 4. Implementasi dan *Testing*

Merupakan tahapan untuk menyatukan komponen-komponen yang dihasilkan pada tahap sebelumnya menjadi satu kesatuan sistem yang utuh. Kemudian dilakukan pengujian untuk memastikan bahwa sistem yang dibuat telah sesuai dengan desainnya dan semua fungsi dapat dipergunakan dengan baik tanpa ada kesalahan.

#### 5. Pemeliharaan Sistem Operational and Maintenance

Merupakan tahap dimana sistem yang telah dibuat diterapkan dalam keseharian. Dalam keseharian penggunaan sistem, tidak menutup kemungkinan adanya perubahan dan perbaikan. Perubahan dapat disebabkan karena seiring berjalannya waktu sistem harus disesuaikan dengan lingkungan pengguna yang mungkin meningkat kebutuhan fungsionalnya. Sementara perbaikan adalah memperbaiki kesalahan pada sistem yang mungkin belum ditemukan sebelumnya [4].

### 2.2.7. Buku Induk Siswa

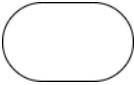
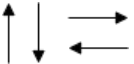
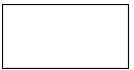
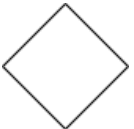

Buku induk siswa merupakan kumpulan daftar nama siswa yang belajar di sekolah tersebut. Catatan di dalam buku induk harus lengkap meliputi, biodata siswa dan hasil prestasi belajar siswa dari semester pertama hingga lulus. Biodata siswa tersebut terdiri dari nama, nis, tempat/tanggal lahir, nama orang tua/wali murid, alamat, pekerjaan, dan sebagainya. Catatan buku harus jelas dan merupakan tanggung jawab kepala sekolah yang bersangkutan, dan pengisiannya bisa ditugaskan pada petugas administrasi [16].

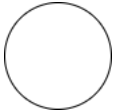
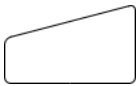

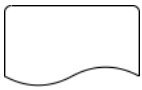
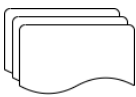
### 2.2.8. Flowchart

Flowchart merupakan penyajian yang sistematis tentang proses dan logika dari kegiatan penanganan informasi atau penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. Bagan alir (flowchart) adalah bagan (chart) yang menunjukkan alir (flow) di dalam program atau prosedur sistem secara logika. Bagan alir

digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi [17]. Berikut ini symbol-simbol yang digunakan dalam *Flowchart* dapat dilihat pada Tabel 2.3.

**Tabel 2.2** Simbol dan Fungsi Flowchart [18]

No	Gambar Simbol	Nama	Keterangan
1.		<i>TerminalPoint</i>	Simbol yang menyatakan awal atau akhir suatu proses.
2.		<i>Flow direction</i>	Simbol yang menyatakan alur proses kerja dan menghubungkan antar simbol.
3.		<i>Proses</i>	Simbol yang menyatakan suatu proses yang dilakukan oleh komputer.
4.		<i>Decision</i>	Digunakan untuk memilih keputusan dalam program sesuai dengan kondisi tertentu.
5.		<i>Input output</i>	Simbol yang menyatakan input atau output yang tidak tergantung jenis peralatannya.

7.		<i>Connector (On-page)</i>	Simbol yang menghubungkan dengan proses lainnya dalam satu halaman yang sama.
9.		<i>Manual Input</i>	Simbol memasukkan data secara manual menggunakan online keyboard.
11.		<i>Manual Operation</i>	Menunjukkan proses/kegiatan yang dilakukan tanpa menggunakan komputer.
12.		<i>Document</i>	Mencetak keluaran dalam bentuk dokumen (melalui printer).
13.		<i>Multiple document</i>	Sama seperti simbol dokumen hanya saja dokumen yang digunakan lebih dari satu.



### 2.2.9. Unified Modelling Language (UML)



UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem [19].

a) **Usecase Diagram**

*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu [20]. Berikut ini symbol-symbol yang digunakan dalam *Use Case Diagram* dapat dilihat pada Tabel 2.1.

**Tabel 2.3** Simbol dan Fungsi Use Case Diagram [14]


No	Gambar Simbol	Nama	Keterangan
1.		<i>Use Case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau actor. Biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
2.		<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 symbol dari actor adalah gambar orang, tapi actor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama actor.

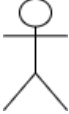


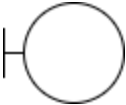



4.		<i>Association</i>	Komunikasi antara actor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor.
5.		<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

#### b) *Sequence Diagram*

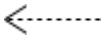
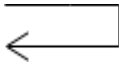
Diagram sequence menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sequence maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sequence juga dibutuhkan untuk melihat scenario yang ada pada *use case* [14]. Berikut ini symbol-simbol yang digunakan dalam *Sequence Diagram* dapat dilihat pada Tabel 2.2.

**Tabel 2.4** Simbol dan Fungsi Sequence Diagram [14]

No	Gambar Simbol	Nama	Keterangan
1.		<i>Object</i>	Menggambarkan sebuah class atau object.

2.		<i>Actor</i>	Menggambarkan pengguna yang berinteraksi dengan sistem.
3.		<i>Lifeline</i>	Menggambarkan garis titik-titik yang terhubung dengan objek.
4.		<i>Activation boxes</i>	Menggambarkan panjang waktu yang dibutuhkan sebuah objek dalam mengerjakan tugasnya.
5.		<i>Boundary</i>	Digunakan untuk menggambarkan sebuah form.
6.		<i>Control</i>	Digunakan untuk menghubungkan boundary dengan tabel.
7.		<i>Entity Class</i>	Menggambarkan sistem awal yang menjadi landasan untuk menyusun basis data.
8.		<i>Message</i>	Menggambarkan pesan atau interaksi antar objek.



9.		<i>Message to self</i>	Menggambarkan pesan balikan atau reaksi dari objek sebelumnya.
10.		<i>Message return</i>	Menggambarkan hasil dari pengiriman message.