

## **BAB II LANDASAN**

### **TEORI**

#### **II.1. Sistem Informasi Geografis**

Sistem Informasi Geografis (SIG) itu sendiri merupakan suatu sistem informasi yang dirancang untuk bekerja dengan data yang terreferensikan secara spasial atau geografis. Dengan kata lain, Sistem Informasi Geografis merupakan sistem basisdata dengan kemampuan khusus dalam menangani data yang terreferensi secara spasial. Atau dapat dikatakan juga kalau Sistem Informasi Geografis merupakan sistem mendukung proses pengambilan keputusan spasial dan mampu menginterasikan deskripsi lokasi dengan karakteristik fenomena yang ditemukan.(Eddy Prahasta:2014;101).

Dan di dalam definisi lain Sistem Informasi Geografis merupakan suatu kesatuan formal yang terdiri dari berbagai sumber daya fisik dan logika yang berkenaan dengan objekobjek yang terdapat di permukaan bumi. Sistem Informasi Geografis juga merupakan sejenis perangkat lunak yang dapat digunakan untuk pemasukan, penyimpanan, manipulasi, menampilkan dan keluaran informasi geografis berikut atribut-atributnya. (I Komang Adi Paramarta. (2013). “Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI).”Sistem Informasi Geografis Rencana Tata Ruang Wilayah (RTRW) Kabupaten Buleleng Berbasis *Web*. 2. 244).

Sedangkan menurut Bern (1992) Sistem Informasi Geografis merupakan sistem komputer untuk memanipulasi data geografis. Sistem ini diimplementasikan dengan menggunakan perangkat keras dan perangkat lunak

komputer yang berfungsi untuk akuisisi dan verifikasi data, kompilasi data, penyimpanan data, perubahan dan updating data, manajemen dan pertukaran data, manipulasi data, pemanggilan dan presentasi data, serta analisa data. (Eddy Prahasta:2014;101).

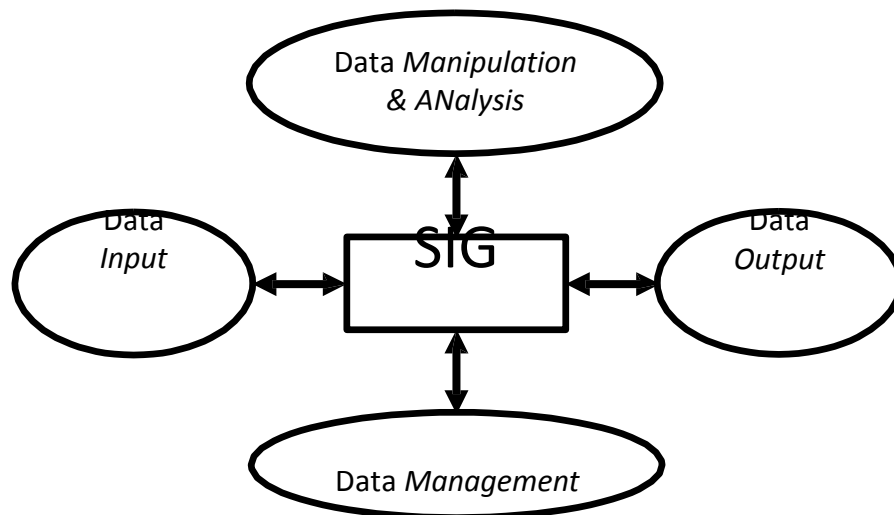
Dari definisi diatas maka dapat disimpulkan bahwa Sistem Informasi Geografis merupakan sebuah sistem yang dirancang secara khusus untuk memberikan informasi tentang geografis permukaan bumi atau letak suatu lokasi yang ada dipermukaan bumi dengan memanfaatkan data *vector* dan data *spasial* yang dipadukan dengan perangkat lunak dan perangkat keras untuk mengolah data *geografis* permukaan bumi agar tetap *terupdate*.

### **II.1.1. Sub Sistem Sistem Informasi Geografis**

Dari definisi diatas maka sistem informasi geografis dapat diuraikan menjadi beberapa sub-sistem sebagai berikut :

- a. Data Input : mengumpulkan, mempersiapkan, dan menyimpan data spasial dan atributnya. Sub-sitem ini bertanggung jawab dalam mengkonversikan format data aslinya ke dalam format SIG.
- b. Data Output : menampilkan, dan menghasilkan keluaran basisdata spasial, seperti table, grafik, report, peta, dan lain sebagainya.
- c. Data *Management* : mengorganisasikan data spasial dan table atribut ke dalam sistem basisdata hingga mudah untuk dipanggil kembali, di *update*, dan di *edit*.

- d. *Data Manipulation dan Analysis* : menentukan informasi yang dihasilkan oleh Sistem Informasi Geografis. Selain itu, sub-sistem ini memanipulasi data untuk menghasilkan informasi yang diharapkan.



**Gambar II.1. Ilustrasi Sub-Sistem SIG**  
(Sumber : Eddy Prahasta ; 2014: 103)

### II.1.2. Komponen Sistem Informasi Geografis

Sistem informasi geografis merupakan sistem kompleks yang umumnya terintegrasi dengan sistem computer lainnya ditingkat fungsional dan jaringan. Jika diuraikan maka SIG terdiri dari komponen dengan berbagai karakteristiknya :

a. Perangkat Keras

sistem informasi geografis tersedia di berbagai *platform* perangkat keras; mulai dari *PC desktop*, *workstations*, hingga *multi-user host*. Walaupun demikian, fungsional SIG tidak terikat ketat pada karakteristik fisik perangkat kerasnya hingga keterbatasan memori pada PC dapat diatasi. Adapun perangkat keras yang sering digunakan dalam

SIG adalah computer (PC/CPU), *mouse*, *keyboard*, *monitor* dan lain sebagainya.

b. Perangkat Lunak

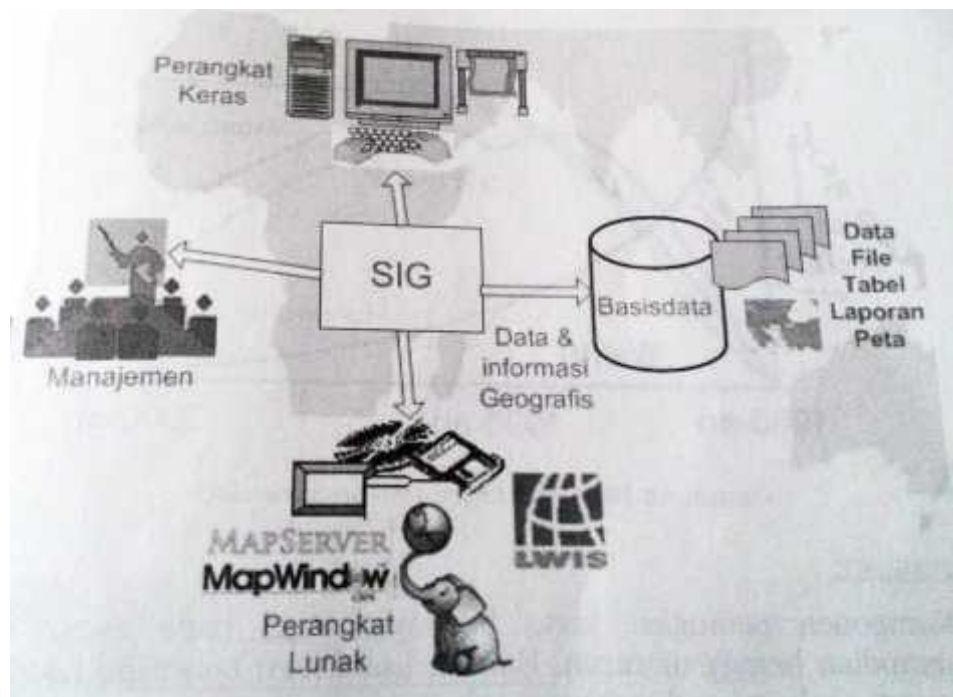
SIG merupakan sistem perangkat lunak dimana sistem basis datanya memegang peranan kunci.

c. Data dan Informasi Geografis

SIG dapat mengumpulkan dan menyimpan data/informasi yang diperlukan baik tidak langsung maupun langsung mendigitasi data spasialnya dari peta analog dan memasukkan data atributnya dari table/laporan dengan menggunakan keyboard.

d. Manajemen

proyek SIG akan berhasil jika dikelola dengan baik dan dikerjakan oleh orang yang memiliki keahlian yang tepat pada semua tingkatan.



**Gambar II.2. Komponen-Komponen SIG**  
(Sumber : Eddy Prahasta ; 2014: 105)

## II.2. PHP

PHP adalah bahasa pemrograman untuk dijalankan melalui halaman web, umumnya digunakan untuk mengolah informasi di internet. Sedangkan dalam pengertian lain PHP adalah singkatan dari PHP *hypertext preprocessor* yaitu bahasa pemrograman *web server-side* yang berbasis *open source* atau gratis, karena PHP merupakan script yang menyatu dengan HTML dan berada pada *server side* HTML *embedded scripting*. (Rulianto Kurniawan:2010;4).



**Gambar II.3. Logo PHP**  
(Sumber : Rulianto Kurniawan ; 2010: 2)

### II.2.1. Versi PHP

#### 1. PHP / FI

Pada saat PHP pertama kali diciptakan oleh Rasmus Lerdorf, yang dulu PHP bernama FI, PHP baru bisa digunakan untuk mengolah form web. Kemudian versi pertama ini banyak dikembangkan sejak diliris kepada khalayak umum.

#### 2. PHP Versi 2.0

Pada versi 2.0 php sudah terintegrasi dengan bahasa pemrograman seperti C dan dilengkapi dengan modul sehingga kualitas PHP juga sangat meningkat drastis.

#### 3. PHP Versi 3.0

Di versi 3.0 dilakukan perubahan nama dari awalnya PHP ( *Personal Home Page* ) menjadi PHP ( *Hypertext Preprocessor* ).

#### **4. PHP Versi 4.0**

Versi 4.0 merupakan versi yang paling banyak digunakan pada awal abad 21 karena PHP versi ini sudah mampu membangun web secara kompleks dengan stabilitas kecepatan yang tinggi pula.

#### **5. PHP Versi 5.0**

Pada versi PHP 5.0 terjadi perubahan yang sangat besar, dimana PHP kini telah berubah menjadi suatu yang berparadigma dengan obyek. Versi PHP 5.0 ini dikembangkan oleh perusahaan yang bernama Zend. Serta pada PHP 5.0 ini juga melahirkan *framework - framework* PHP yang ada.

#### **6. PHP Versi 6.0**

Versi 6.0 ini PHP sudah mendukung *Unicode*, perbaiki keamanan, memiliki fitur dan konstruksi baru, seperti contohnya dukungan terhadap *labeled breaks*, membangun perulangan untuk array multi dimensi dll. (Gilang's, Blog, 2013, *perkembangan versi PHP*, <http://blogger-klaten.blogspot.co.id/2013/11/perkembangan-versi-php.html>, diakses tanggal 26 April 2016).

### **II.3. MySQL**

*MySQL* adalah salah satu jenis *database server* yang sangat terkenal. *MySQL* termasuk jenis RDBMS (*Relational Database Management System*). *MySQL* ini mendukung bahasa pemrograman PHP. Dan *MySQL* juga mempunyai

*query* atau bahasa SQL (*Structured Query Language*) yang *simple* dan menggunakan *escape character* yang sama dengan *PHP*.



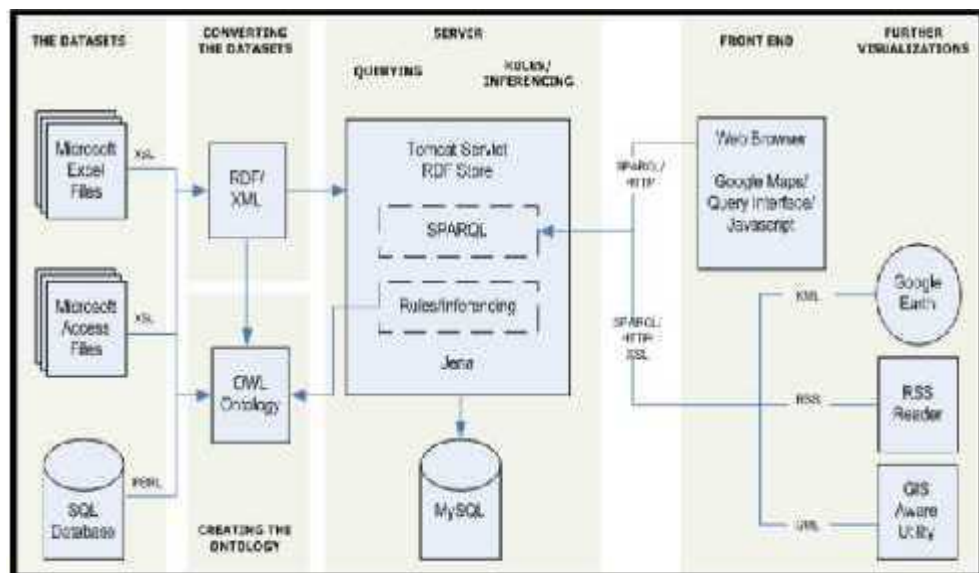
**Gambar II.4. Logo MySQL**  
(Sumber : Rulianto Kurniawan ; 2010: 16)

#### **II.4. Google Maps API**

*Google Maps* merupakan sebuah aplikasi pemetaan *online* yang cukup populer. Pada *google maps* titik koordinat suatu tempat ditunjukkan dengan sistem koordinat geografis. (Bernadus, Ester:2012;9). Atau *Google Maps* merupakan layanan gratis mapping online yang disediakan oleh *Google* melalui <http://maps.google.com>. Layanan ini memberikan informasi geografis hampir seluruh wilayah di bumi secara interaktif. *Google Maps* sendiri juga menyediakan API salah satunya berbentuk *library Javascript* untuk mengakses informasi geografis ini. Dengan adanya API ini, *web programmer* dapat membuat *webnya* sendiri yang memiliki fitur SIG dengan bantuan dari *Google Maps*. Sehingga dapat dilakukan penghematan biaya dan waktu untuk membangun SIG, *web programmer* cukup berkonsentrasi pada data-data utamanya, sementara data geografis dan peta diserahkan ke *Google Maps*. (Agus Qomaruddin Munir. (2014). "Jurnal Teknologi Informasi." Sistem Informasi Geografis Pemetaan Bencana Alam Menggunakan *Google Maps*. IX . 2430).

Dari pengertian diatas dapat disimpulkan bahwa *Google Maps API* adalah fungsi fungsi pemrograman yang disediakan oleh *Google Maps* agar *Google Maps*

bisa di integrasikan kedalam *Web* atau aplikasi yang sedang buat. *Google Maps* API dapat ditambahkan ke *website* atau aplikasi yang kita buat menggunakan *JavaScript*. API tersebut menyediakan banyak fasilitas dan utilitas untuk memanipulasi peta dan menambahkan konten ke peta melalui berbagai layanan.



**Gambar II.5. Arsitektur *Google Maps***  
(Sumber : Agus Qomaruddin Munir; 2014: Vol IX)

## II.5. *Macromedia Dreamweaver*

Macromedia dreamweaver 8 merupakan HTML editor professional yang berfungsi mendesain, melakukan editng dan mengembangkan aneka website. Salah satu kelebihan dreamweaver 8 yaitu ruang kerja dreamweaver 8 beserta *tools* yang tersedia dapat digunakan dengan sangat mudah dan cepat sehingga anda bisa membangun suatu website dengan cepat dan tanpa harus melakukan coding. Selain itu, dreamweaver 8 juga mempunyai integrasi dengan produk macromedia lainnya, seperti flash dan firework, flash sudah sangat terkenal sebagai sebagai program untuk membuat animasi yang berbasis *web* dengan

perkembangan kebutuhan dan teknologi, flash akhir-akhir ini juga digunakan untuk membuat animasi dan video. Fungsi coding dreamweaver 8 tidak hanya mendukung coding HTML tetapi juga CSS ( *Cascading Style Sheet* ), Javascript, Coldfusion, ASP ( *Active Server Sheet* ), JSP ( *Java Server Page* ), dan Dreamweaver juga memungkinkan anda membangun website dengan server berbahasa CFML ( *Colfusion Markup Language* ), ASP.net, dan PHP.

## **II.6. UML (*Unified Modelling Language*)**

### **II.6.1. Pengenalan UML**

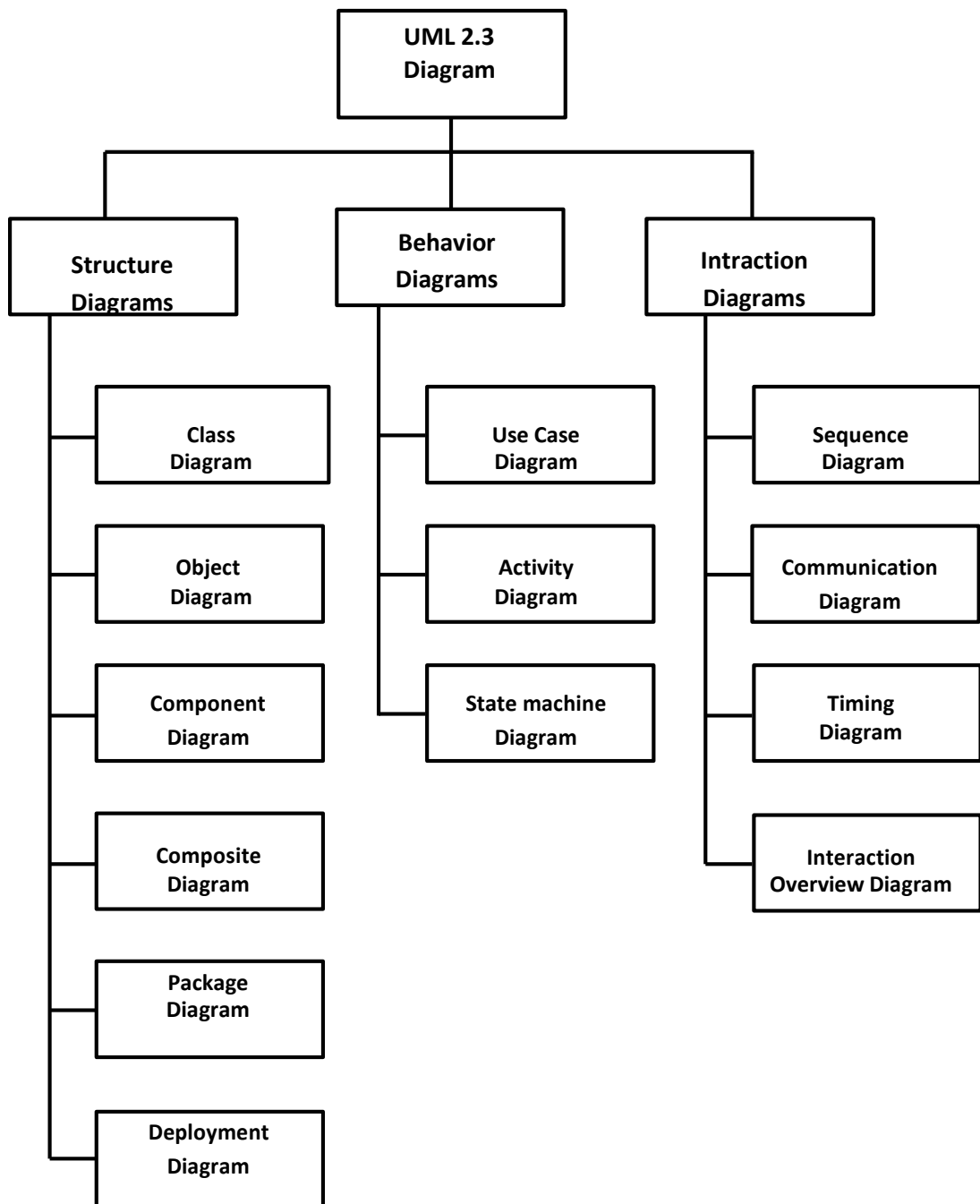
Sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasi, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML (*Unified Modeling Language*) merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek. (Rosa A. S dan Shalahuddin. M:2015;137-138).

### **II.6.2. Sejarah dan Diagram UML**

Bahasa pemrograman berorientasi objek yang pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967. Bahasa

pemrograman ini kurang berkembang. Konsep pemikiran Ivar Jacobson, dan beberapa konsep lainnya dimana James R. Rumbaigh, Grady Booch, dan Ivar Jacobson bergabung dalam sebuah perusahaan yang bernama *Rational Software Corporation* sehingga menghasilkan bahasa yang disebut dengan *Unified Modeling Language* (UML) Sehingga pada tahun 1996, *Object Management Group* (OMG) mengajukan proposal agar adanya standarisasi pemodelan berorientasi objek dan pada bulan September 1997 UML diakomodasi oleh OMG sehingga sampai saat ini UML telah memberikan kontribusinya yang cukup besar di dalam metodologi berorientasi objek dan hal-hal yang terkait dalamnya. Secara fisik, UML adalah sekumpulan spesifikasi yang dikeluarkan oleh OMG. UML yang terbaru adalah UML 2.3 yang terdiri dari 4 macam spesifikasi, yaitu Diagram Interchange Specification,,. UML Infrastructure, UML Superstructure, dan Object Constraint Language (OCL). (Rosa A. S dan Shalahuddin. M:2015;139-140).

Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar dibawah ini.



**Gambar II.6. Diagram UML**  
(Sumber : Rosa A . S dan Shalahuddin M ; 2015: 140)

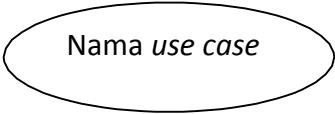
## 1. Use Case Diagram

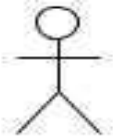


*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case diagram* yaitu pendefinisian apa yang disebut aktor dan *use case*.




1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut ini adalah simbol-simbol yang ada pada diagram *use case* :

**Tabel II.1. Komponen Use Case Diagram**

No	Simbol	Keterangan
1	<p style="text-align: center;"><i>Use Cases</i></p> 	<p>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>.</p>

2	<p>Aktor/<i>Actor</i></p> 	<p>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, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i>.</p>
3	<p><i>Association</i></p> 	<p>Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i>.</p>
4	<p>Ekstensi / <i>extend</i></p> <p>&lt;&lt;<i>extend</i>&gt;&gt;</p> 	<p>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; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; ditambahkan, misal arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>

5	<p>Generalisasi / generalization</p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>
6	<p>Menggunakan / <i>include</i> / <i>uses</i></p> <p>&lt;&lt; include &gt;&gt;</p>  <p>&lt;&lt;uses&gt;&gt;</p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i>.</p>

(Sumber : Rosa A . S dan Shalahuddin M ; 2015: 156)



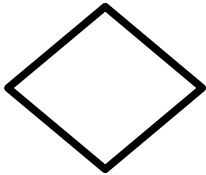


## 2. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”.

Berikut ini adalah simbol-simbol yang ada pada diagram activity :

**Tabel II.2. Komponen Activity Diagram**

No	Simbol	Keterangan
	Status Awal	Status awal aktivitas sistem, sebuah

1		diagram aktivitas memiliki sebuah status awal.
2	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3	Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
4	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

(Sumber : Rosa A . S dan Shalahuddin M ; 2015: 162)

### 3. 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 atau operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

1. Kelas main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2. Kelas yang menangani tampilan sistem (*view*)

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

3. Kelas yang diambil dari pendefinisian *use case* (*controller*)





Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.



4. Kelas yang diambil dari pendefinisian data (*model*)

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Berikut adalah simbol-simbol yang ada pada class diagram :

Tabel II.3. Class Diagram

No	Simbol	Keterangan
1	<p style="text-align: center;">Kelas</p> <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <p style="text-align: center;">Nama_kelas</p> <hr/> <p style="text-align: center;">+ atribut</p> <hr/> <p style="text-align: center;">+operasi()</p> </div>	Kelas pada struktur sistem
2	<p style="text-align: center;">Antarmuka</p> <div style="text-align: center; margin: 10px auto;">  </div> <p style="text-align: center;">Nama interface</p>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3	<p style="text-align: center;">Asosiasi / <i>association</i></p> <div style="text-align: center; margin: 10px auto;">  </div>	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4	<p style="text-align: center;">Asosiasi berarah / <i>directed association</i></p> <div style="text-align: center; margin: 10px auto;">  </div>	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5	<p style="text-align: center;">Generalisasi</p> <div style="text-align: center; margin: 10px auto;">  </div>	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).

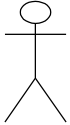


6	Kebergantungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antarkelas.
7	Agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua- bagian ( <i>whole-part</i> ).

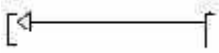

(Sumber : Rosa A . S dan Shalahuddin M ; 2015: 146-147)

#### 4. Squence Diagram

*Sequence diagram* atau diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirim dan diterima antar objek”. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Tabel II.4. Simbol Squence Diagram

No	Simbol	Keterangan
1	<p>Aktor</p>  <p>Atau</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;"> <p><b><u>Nama Aktor</u></b></p> </div> <p>Tanpa waktu aktif</p>	<p>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, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i>.</p>
2	<p>Objek <i>lifeline</i></p> 	<p>Objek <i>entity</i>, antarmuka yang saling berinteraksi.</p>
3	<p><i>Message</i></p> 	<p>Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang</p>

		terjadi
4	<p><i>Message</i></p> 	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
5	<p>Pesan tipe create</p> <p>&lt;&lt;create&gt;&gt;</p> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.

(Sumber : Rosa A . S dan Shalahuddin M ; 2015: 165-167)