

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Menurut Asbon Hendra (2012 : 157) Sistem merupakan kumpulan Kumpulan dari unsur atau elemen elemen yang saling berkaitan/berinteraksi dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu

II.1.1. Data

Menurut Asbon Hendra (2012 : 157) Data merupakan *raw material* untuk suatu informasi. Perbedaan data dan informasi sangat relatif , tergantung pada nilai gunanya bagi manajemen yang memerlukan.

II.1.2. Informasi

Menurut Asbon Hendra (2012 : 167) informasi merupakan Data yang telah di proses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat.

Menurut Asbon Hendra (2012 : 169) Ada 4 oprasi dasar sistem informasi, yaitu mengumpulkan, mengelolah, menyimpan, dan menyebarkan informasi. Informasi mungkin dikumpulkan dari lingkungan dalam atau luar dan memungkinkan didistribusikan ke dalam atau keluar organisasi. Contoh sebuah sistem informasi penjualan : pengumpulan data transaksi dan faktur penjualan. Sistem ini memanfaatkan perangkat keras dan perangkat lunak komputer, prosedur manual, model manajemen, dan basis data.

II.1.3. Kualitas Informasi

Menurut Asbon Hendra (2012 : 168) Kualitas informasi adalah bias terhadap eror karena kesalahan cara pengukuran dan pengumpulan, kegagalan mengikuti prosedur pemrosesan, kehilangan atau data tidak di proses, kesalahan perekaman atau koreksi data, kesalahan file histori/master, kesalahan prosedur pemrosesan, serta ketidakberfungsian sistem.

Kualitas informasi tergantung dari 3 hal, yaitu informasi harus :

1. **Akurat** , berarti informasi harus bebas dari kesalahan-kesalahn dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya.
2. **Tepat pada waktunya**, berarti informasi yang datang pada penerima tidak boleh terlambat.
3. **Relevan**, berarti informasi tersebut mempunyai manfaat untuk pemakainya.relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda.

II.1.4. Sistem Informasi

Menurut Asbon Hendra (2012 : 170) suatu sistem informasi dapat diklasifikasikan sebagai sistem informasi formal dan informal. Sistem informasi formal adalah sistem informasi yang memiliki prosedur dan kebijaksanaan tertulis dalam dokumen. Contohnya undang-undang pajak penjualan.Sistem informasi informal adalah sistem informasi yang memiliki prosedur dan kebijaksanaan tidak tertulis dalam dokumen, contohnya pemberian diskon penjualan secara tidak terduga.

Komponen fisik sistem informasi dibagi menjadi :

1. Perangkat keras komputer : CPU, *Storage*, perangkat *input/output*, terminal untuk interaksi, serta media komunikasi data.
2. Perangkat lunak komputer : perangkat lunak sistem (sistem operasi dan *utility*-nya), perangkat lunak umum aplikasi (bahasa pemrograman), dan perangkat lunak aplikasi (aplikasi akuntansi).
3. Basis data : penyimpanan data pada media penyimpanan komputer.
4. Prosedur : langkah-langkah penggunaan sistem.
5. Personel untuk pengelolaan operasi (SDM).
6. *Clerical personnel* (untuk menangani transaksi dan pemrosesan data dan melakukan inquiry = operator).
7. *First level manager*, untuk mengelola pemrosesan data di dukung dengan perencanaan, penjadwalan, identifikasi situasi *out-of-control*, dan pengambilan keputusan level menengah ke bawah.
8. *Staff specialist*, digunakan untuk analisis perencanaan dan pelaporan.
9. *Management*, untuk pembuatan laporan berkala, permintaan khusus, analisis khusus, laporan khusus, serta pendukung identifikasi masalah dan peluang.

II.2. Plaza

Menurut Edwin Japariato (2012 : 1) Dalam perebutan penguasaan pangsa pasar diperlukan pembentukan positioning yang unggul dalam persepsi konsumen, karena perusahaan yang berada pada posisi unggul dibandingkan

dengan pesaingnya produk yang dihasilkan menjadi pilihan utama konsumen. Salah satu bidang industri yang saat ini menunjukkan peningkatan persaingan adalah bisnis retail. Persaingan yang dimulai pada tahun 1998, yaitu pada saat krisis ekonomi mulai melanda semua bidang kehidupan sehingga disebut sebagai krisis multi-dimensional, dan peningkatan tersebut terus terlihat hingga tahun 2008 ini. Hal ini dapat dilihat dengan semakin menjamurnya jaringan *shopping centre* di kalangan masyarakat sekarang ini. Beberapa tahun terakhir ini, sedang marak muncul *shopping centre* yang menjadi *leisure* dan *entertainment* bagi masyarakat. Fenomena seperti itulah yang sering kali dijadikan sebagai ajang persaingan bisnis seperti yang telah disebutkan sebelumnya.

Persaingan yang semakin ketat ini membuat para pemain pasar dituntut untuk selalu inovatif dalam melakukan pembaruan pemasaran sehingga tetap menarik minat pengunjung di antara jajaran jaringan *shopping centre* yang telah ada sebelumnya, terwujudlah banyak proyek besar pengembang untuk mendirikan pusat perbelanjaan. Seiring dengan perkembangan jaman, untuk terus menarik minat pengunjung di antara jajaran jaringan *shopping centre* terwujudlah sebuah proyek besar dari Trisensa dan PT Dwijaya Manunggal, yang merupakan anak perusahaan Pakuwon Group, yang kemudian dikenal dengan nama *Royal Plaza*.

II.3. Sistem Informasi Geografis

Menurut Adam Suseno & Ricky Agus (2012 : 1) Sistem informasi yang memiliki kepaduan antara teknologi informasi dan aktifitas dari orang yang menggunakan teknologi itu untuk mengembangkan dan mengaplikasikan dalam mendukung sebuah operasi atau manajemen di bidang geografis, merupakan bagian dari perkembangan di ilmu sistem informasi geografis.

Dalam pengembangannya, sistem informasi ini dibuat dengan tujuan memanfaatkan teknologi informasi. Hal ini tidak terlepas dari semakin banyaknya software yang dibuat untuk membantu dalam pengerjaannya khususnya dalam sistem informasi geografis. Sesuatu yang berhubungan dengan sistem informasi tentunya tidak terlepas dari hubungan dengan sistem data dan aktifitas lain dalam penggunaan *software* nya. Oleh karena itu dibutuhkan sebuah software yang mendukung dalam sistem informasi geografis.

Menurut Adam Suseno & Ricky Agus (2012 : 2) Sistem Informasi Geografis atau disingkat SIG dalam bahasa Inggris *Geographic Information System* (disingkat GIS) merupakan sistem informasi khusus yang mengelola data yang memiliki informasi spasial (bereferensi keruangan). Atau dalam arti yang lebih sempit adalah sistem komputer yang memiliki kemampuan untuk membangun, menyimpan, mengelola dan menampilkan informasi bereferensi geografis atau data geospasial untuk mendukung pengambilan keputusan dalam perencanaan dan pengelolaan suatu wilayah, misalnya data yang diidentifikasi menurut lokasinya, dalam sebuah database. Para praktisi juga memasukkan orang yang membangun dan mengoperasikannya dan data sebagai bagian dari sistem ini.

Menurut Adam Suseno & Ricky Agus (2012 : 2) Teknologi Sistem Informasi Geografis dapat digunakan untuk investigasi ilmiah, pengelolaan sumber daya, perencanaan pembangunan, kartografi dan perencanaan rute. Misalnya, Sistem informasi geografis bisa membantu perencana untuk secara cepat menghitung waktu tanggap darurat saat terjadi bencana alam, atau Sistem informasi geografis dapat digunakan untuk mencari lahan basah (*wetlands*) yang membutuhkan perlindungan dari polusi atau dapat digunakan mencari informasi sebuah tempat khusus dan banyak manfaat lain yang dapat dikembangkan dalam sistem informasi geografis ini.

II.3.1. Komponen Sistem Informasi Geografis

Menurut Adam Suseno & Ricky Agus (2012 : 5) Komponen-komponen pendukung Sistem informasi geografis terdiri dari lima komponen yang bekerja secara terintegrasi yaitu perangkat keras (hardware), perangkat lunak (software), data, manusia, dan metode yang dapat di lihat pada gambar II.1 sebagai berikut :



Gambar II.1 : Komponen Sistem Informasi Geografis

(Sumber : Adam Suseno & Ricky Agus ; 2012 : 6)

1. Perangkat Keras (*hardware*)

Perangkat keras Sistem informasi geografis adalah perangkat-perangkat fisik yang merupakan bagian dari sistem komputer yang mendukung analisis goegrafi dan pemetaan. Perangkat keras Sistem informasi geografis mempunyai kemampuan untuk menyajikan citra dengan resolusi dan kecepatan yang tinggi serta mendukung operasioperasi basis data dengan volume data yang besar secara cepat. Perangkat keras Sistem informasi geografis terdiri dari beberapa bagian untuk menginput data, mengolah data, dan mencetak hasil proses. Berikut ini pembagian berdasarkan proses :

- a. Input Data : mouse, digitizier, scanner
- b. Olah Data : harddisk, processor, RAM, VGA card
- c. Output Data : plotter, printer, screening

2. Perangkat Lunak (*software*)

Perangkat lunak digunakan untuk melakukan proses menyimpan, menganalisa, memvisualkan data-data baik data spasial maupun non-spasial. Perangkat lunak yang harus terdapat dalam komponen software Sistem informasi geografis adalah :

- a. Alat untuk memasukkan dan memanipulasi data sistem informasi geografis
- b. *Data Base Management System* (DBMS)
- c. Alat untuk menganalisa data-data
- d. Alat untuk menampilkan data dan hasil analisa

3. Data

Pada prinsipnya terdapat dua jenis data untuk mendukung sistem informasi geografis yaitu :

1. Data Spasial

Data spasial adalah gambaran nyata suatu wilayah yang terdapat di permukaan bumi. Umumnya direpresentasikan berupa grafik, peta, gambar dengan format digital dan disimpan dalam bentuk koordinat x,y (vektor) atau dalam bentuk image (raster) yang memiliki nilai tertentu.

2. Data Non Spasial (Atribut)

Data non spasial adalah data berbentuk tabel dimana tabel tersebut berisi informasi- informasi yang dimiliki oleh obyek dalam data spasial. Data tersebut berbentuk data tabular yang saling terintegrasi dengan data spasial yang ada.

4. Manusia

Manusia merupakan inti elemen dari sistem informasi geografis karena manusia adalah perencana dan pengguna dari sistem informasi geografis. Pengguna sistem informasi geografis mempunyai tingkatan seperti pada sistem informasi lainnya, dari tingkat spesialis teknis yang mendesain dan mengelola sistem sampai pada pengguna yang menggunakan sistem informasi geografis untuk membantu pekerjaannya sehari-hari.

5. Metode

Metode yang digunakan dalam Sistem informasi geografis akan berbeda untuk setiap permasalahan. Sistem informasi geografis yang baik tergantung pada aspek desain dan aspek realnya.

II.4. PHP

Menurut Minarni (2013 : 40) PHP adalah bahasa pemrograman *script* yang paling banyak dipakai saat ini. PHP paling banyak dipakai untuk memogram situs web dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakai lainnya. contoh terkenal dari aplikasi php adalah PhpBB dan mediaWiki (software dibelakang Wikipedia).

II.5. MySQL

Menurut Minarni (2013 : 40) MySQL merupakan susunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structure Query Language*). kendala dari suatu system database (DBMS) dapat diketahui dari cara kerja optimezernya dalam melakukan perintah-perintah sql yang dibuat user ataupun program-program aplikasinya sebagai database server, MYSQL dapat dikatakan lebih unggul dibanding database server lainnya dalam query data.. MYSQL adalah salah satu dari sekian banyak sistem database yang merupakan terobosan solusi yang tepat dalam aplikasi database.

II.6. Database

Menurut Budi Raharjo (2011 : 3) istilah database banyak memiliki definisi. Untuk sebagian kalangan sederhana *database* diartikan sebagai kumpulan data (buku, nomor telepon, daftar pegawai, dan lain sebagainya). Ada juga yang menyebut *database* dengan definisi lain yang lebih formal dan tegas. *Database* didefinisikan sebagai kumpulan data yang terintegrasi dan diatur sedemikian rupa sehingga data tersebut dapat dimanipulasi, diambil dan dicari secara cepat.

Selain berisi data, *database* juga berisi *metadata*. Metadata adalah data yang menjelaskan tentang struktur dari data itu sendiri. Sebagai contoh, Anda dapat memperoleh informasi tentang nama-nama kolom dan tipe yang ditampilkan tersebut disebut *metadata*.

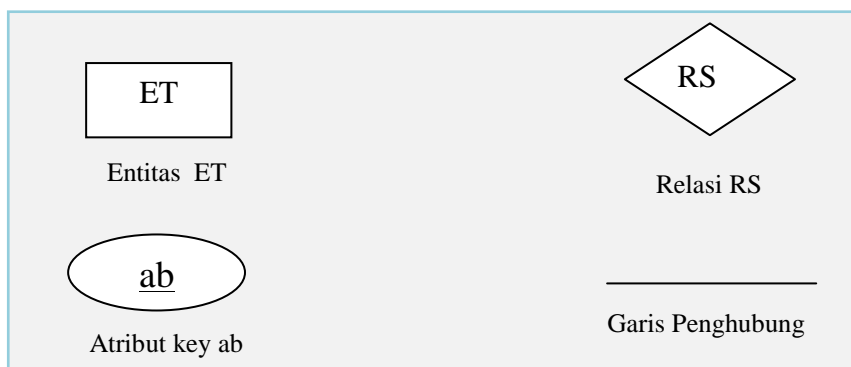
II.6.1. Pemodelan Data

Menurut Yudi Priyadi (2014 : 10) Terdapat beberapa penjelasan mengenai pemodelan basis data. Suatu basis data dapat digunakan secara bebas untuk menggambarkan dan memberikan deskripsi mengenai kumpulan informasi yang tersimpan dalam *data storage* komputer. Secara sederhana, definisi untuk model basis data adalah sekumpulan notasi atau simbol untuk menggambarkan data dan relasinya, berdasarkan suatu konsep dan aturan tertentu suatu pemodelan.

II.6.2. Notasi Diagram E-R

Menurut Yudi Priyadi (2014 : 20) Pemodelan basis data dengan menggunakan diagram relasi antar entitas, dapat dilakukan dengan menggunakan suatu pemodelan basis data yang bernama Diagram *Entity-Relational* (selanjutnya disingkat Diagram E-R). Pada Gambar II.2, terdapat suatu simbol/notasi dasar

yang digunakan pada Diagram E-R,yaitu entitas,relasi, atribut, dan garis penghubung.



Gambar II.2 : Notasi Dasar Diagram E-R

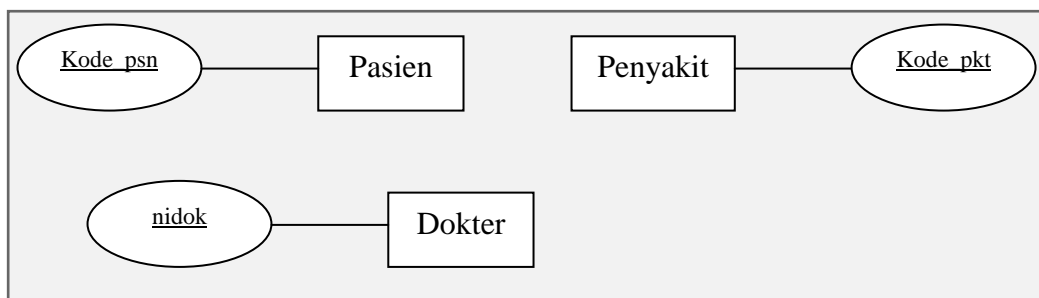
(Sumber : Yudi Priyadi ; 2014 : 20)

1. Entitas

Merupakan notasi untuk mewakili suatu objek dengan karakteristik sama, yang dilengkapi oleh atribut, sehingga pada suatu lingkungan nyata setiap objek akan berbeda dengan objek lainnya. Pada umumnya, objek dapat berupa benda, pekerjaan, tempat dan orang.

2. Atribut

Merupakan notasi yang menjelaskan karakteristik suatu entitas dan juga relasinya. Atribut dapat sebagai key yang bersifat unik, yaitu *Primary Key* atau *Foreign Key*. Selain itu, atribut juga dapat sebagai atributdeskriptif saja, yaitu sebagai pelengkap deskripsi suatu entitas dan relasi.

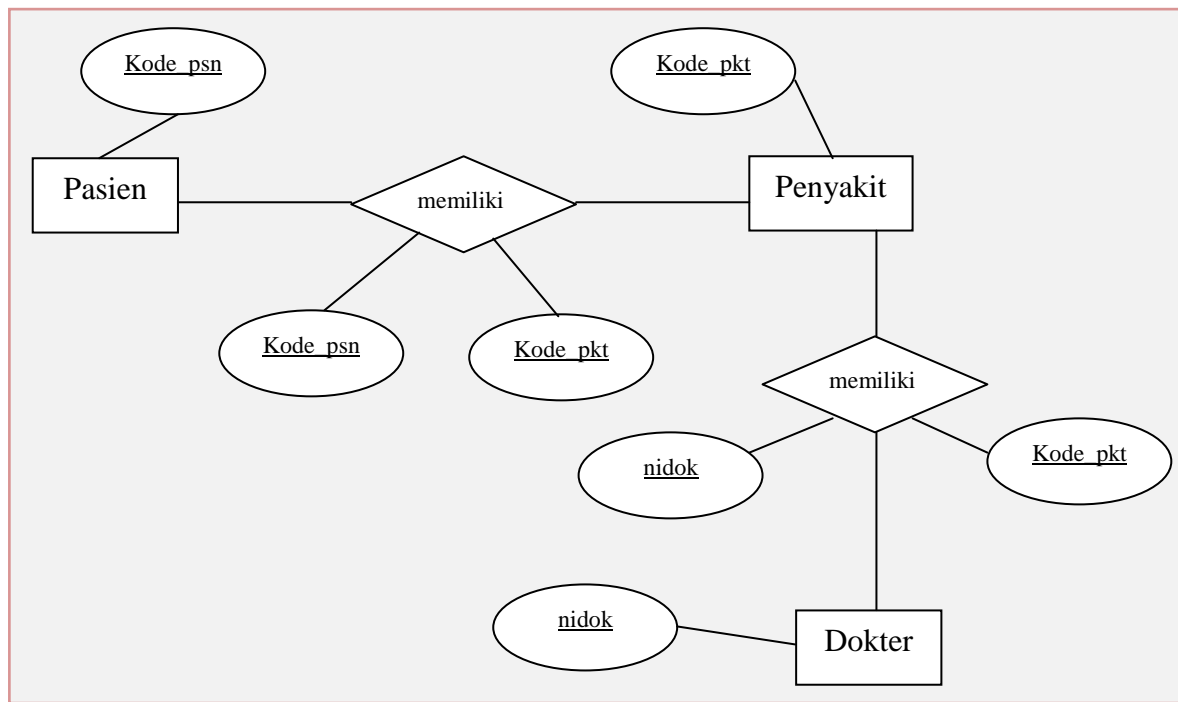


Gambar II.3 : Atribut Key pada Entitas

(Sumber : Yudi Priyadi ; 2014 : 23)

3. Relasi

Merupakan notasi yang digunakan untuk menghubungkan beberapa entitas berdasarkan fakta pada suatu lingkungan.



Gambar II.4 : Pemilihan Relasi untuk Entitas

(Sumber : Yudi Priyadi ; 2014 : 25)

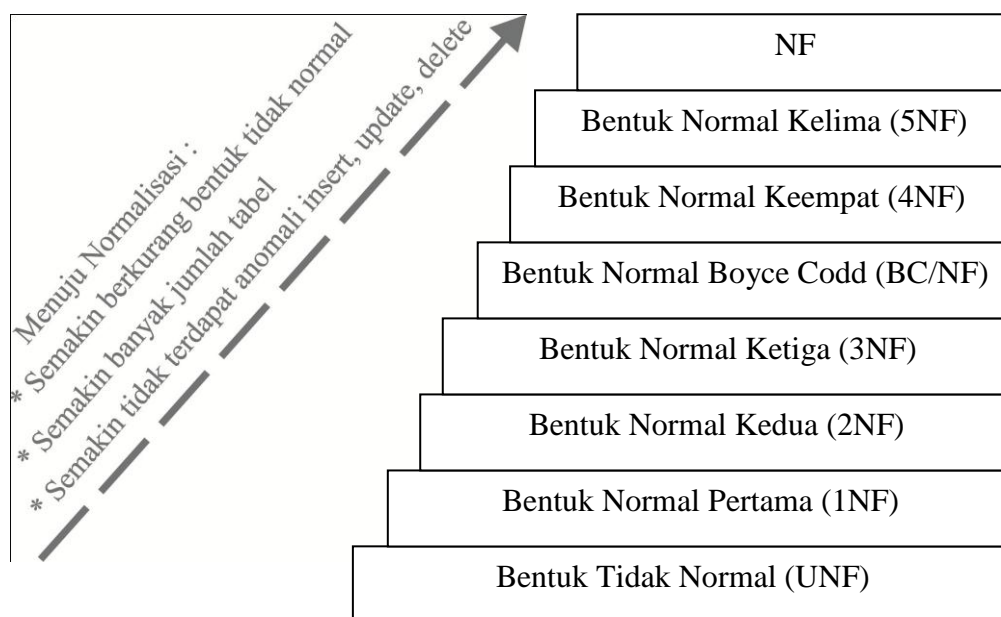
4. Garis penghubung

Merupakan notasi untuk merangkaikan keterkaitan antar notasi yang digunakan dalam Diagram E-R, yaitu entitas, relasi dan atribut.

II.6.3. Normalisasi

Menurut Yudi Priyadi (2014 : 67) Normalisasi merupakan proses sistematis yang dilakukan pada struktur tabel basis data menjadi struktur tabel yang memiliki integritas data, sehingga tidak memiliki data anomali pada saat melakukan *insert*, *delete*, dan *update*. Pada Gambar II.4, tahapan proses sistematis yang dilakukan

mulai dari bentuk tidak normal menjadi bentuk normal memiliki suatu syarat yang harus dipenuhi pada saat menuju suatu bentuk yang lebih baik (*well structured relation*).



Gambar II.5 : Tahapan Proses Bentuk Normalisasi

(Sumber : Yudi Priyadi ; 2014 : 67)

Setiap syarat dalam tahapan suatu bentuk normal memiliki keterkaitan, hal ini disebabkan karena pada setiap bentuk normal mengalami penyempurnaan untuk bentuk normal selanjutnya. Bentuk tidak normal akan semakin berkurang, setelah melalui tahapan perubahan bentuk normalisasi, sehingga berdampak pada jumlah tabel yang semakin banyak, tetapi menuju perbaikan ke dalam bentuk *well structured relation*. Hal ini terjadi akibat dari pengelompokan data suatu tabel agar memiliki ketergantungan secara fungsional.

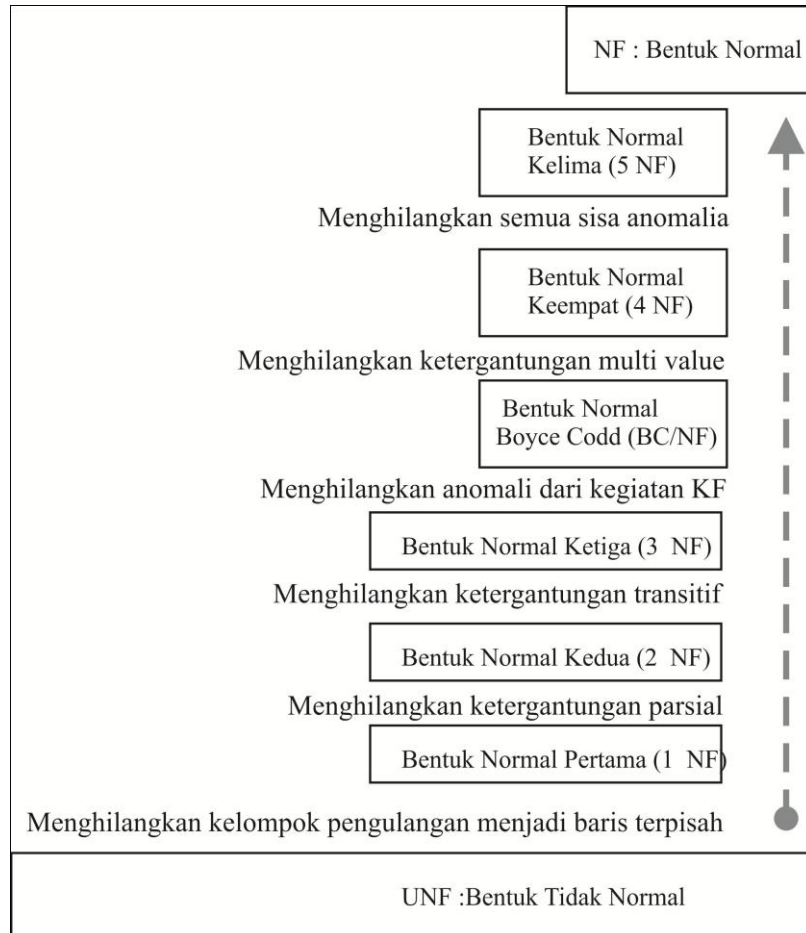
II.6.4. Aturan Proses Normalisasi

Menurut Yudi Priyadi (2014 : 68) Secara sederhana, kegiatan normalisasi adalah melakukan dekomposisi atau penguraian tabel beserta datanya, menjadi tabel yang normal menurut konsep RDBMS. Merujuk pada gambar II.6, dekomposisi diawali dengan melakukan analisis pada suatu tabel atau beberapa contoh formulir yang sudah memiliki data lengkap dalam basis data, tetapi masih dalam bentuk yang tidak normal (UNF). Oleh karena itu agar dapat memenuhi syarat bentuk normal pertama (1NF), pada setiap barisnya diisikan suatu *value* dengan kelompok data yang sama, berdasarkan suatu atribut key. Dengan demikian, kelompok pengulangan dalam suatu baris dapat dihilangkan, karena sudah tidak terdapat *value* yang kosong untuk setiap *field* dan *recordnya*

Setelah memenuhi syarat bentuk normal pertama (1NF), proses berikutnya adalah menghilangkan ketergantungan secara parsial, yaitu dengan cara melakukan dekomposisi tabel menjadi beberapa kelompok tabel berdasarkan field yang memiliki status sebagai key. Hal ini dapat dilakukan oleh salah satu field saja, dengan tetap tidak mengubah arti relasi dan ketergantungannya. Oleh sebab itu, disebut ketergantungan fungsional sebagian (*partially functional*), sehingga syarat bentuk normal kedua (2NF) sudah tercapai.

Bentuk normal kedua (2NF) merupakan syarat yang harus dimiliki untuk menuju bentuk normal ketiga (3NF). Pada proses ini, dilakukan dengan menghilangkan ketergantungan secara transitif, yaitu suatu konsep untuk tabel dari hasil relasi yang didalamnya terdapat ketergantungan secara tidak langsung pada beberapa atributnya. Pada umumnya proses normalisasi sudah dapat tercapai

pada bentuk normal ketiga (3NF), yaitu dengan menghasilkan tabel yang tidak mengalami anomali basis data pada saat proses *insert*, *delete*, dan *update*.



Gambar II.6 : Tahapan Aturan Proses Normalisasi

(Sumber : Yudi Privadi : 2014 : 69)

II.7. Unified Modeling Language (UML)

Menurut Rosa A.S & M. Shalahuddin (2011 : 118) Pada perkembangan teknik pemrograman berorientasi objek, muncullah 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 menspesifikasikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan 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 metode berorientasi objek.

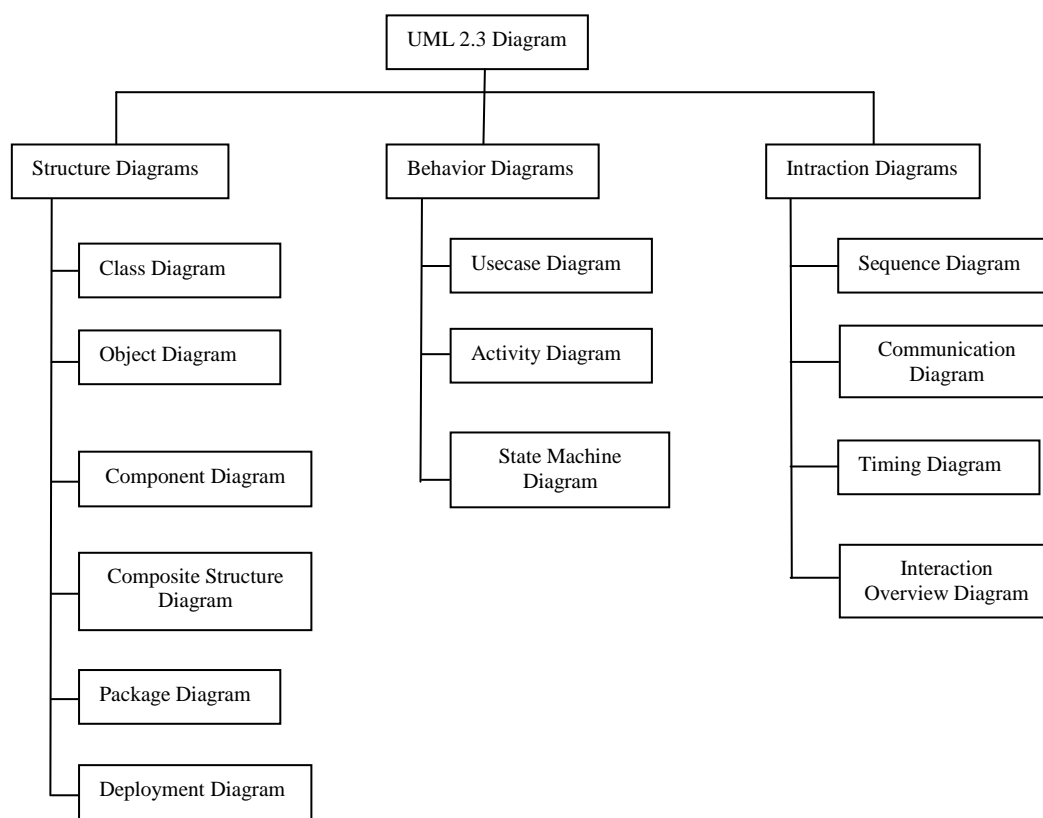
Menurut Prabowo Pudjo Widodo & Herlawati (2011 : 6) UML diaplikasikan untuk maksud tertentu, biasanya antara lain :

1. Merancang perangkat Lunak.
2. Sarana Komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

Blok pembangunan utama UML adalah diagram. Beberapa diagram ada yang rinci (jenis *timing diagram*) dan lainnya ada yang bersifat umum (misalnya diagram kelas). Para pengembang sistem berorientasi objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka rancang. UML memungkinkan para anggota team untuk bekerja sama dengan bahasa model yang sama dengan mengaplikasikan beragam sistem. Intinya UML merupakan alat komunikasi yang konsisten dalam mendukung para pengembang sistem saat ini.

II.7.1 Diagram-Diagram UML

Menurut Rosa A.S & M. Shalahuddin (2011 : 120) 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 II.7 di bawah ini



Gambar II.7 : Diagram UML

(Sumber : Rosa A.S & M. Shalahuddin ; 2011 : 121)

Berikut ini penjelasan singkat dari pembagian kategori tersebut

1. *StructureDiagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.

2. *Behavior Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

1. *Class Diagram*

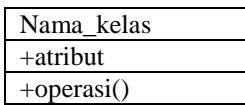



Diagram kelas atau *Class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

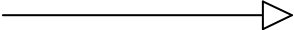
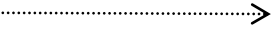
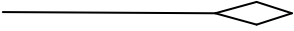
Kelas memiliki apa yang disebut atribut dan metode atau operasi.

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

Berikut Tabel II.I menerangkan simbol-simbol pada diagram kelas :

Tabel II.1 : Diagram Kelas

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi / <i>association</i>  Asosiasi berarah / <i>directed association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi	Relasi antar kelas dengan makna

	generalisasi-spesialisasi (umum khusus)
Kebergantungan 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi / <i>aggregation</i> 	Semua bagian (<i>whole part</i>)

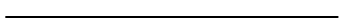
(Sumber : Rosa A.S & M. Shalahuddin ; 2011 : 123)

2. Object Diagram

Diagram objek menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem. Pada diagram objek harus dipastikan semua kelas yang sudah didefinisikan pada diagram kelas harus dipakai objeknya, karena jika tidak, pendefinisian kelas itu tidak dapat dipertanggungjawabkan. Untuk apa mendefinisikan sebuah kelas sedangkan pada jalannya sistem, objeknya tidak pernah dipakai.

Berikut adalah tabel II.2 menerangkan simbol-simbol diagram objek

Tabel II.2 : Diagram Objek

Simbol	Deskripsi		
Objek <table border="1" style="margin-left: 20px;"> <tr> <td>Nama_objek : nama_kelas</td> </tr> <tr> <td>Atribut = nilai</td> </tr> </table>	Nama_objek : nama_kelas	Atribut = nilai	Objek dari kelas yang berjalansaat sistem dijalankan
Nama_objek : nama_kelas			
Atribut = nilai			
Link 	Relasi antar objek		

(Sumber : Rosa A.S & M. Shalahuddin ; 2011 : 124)

3. Component Diagram

Diagram komponen atau component diagram dibuat untuk menunjukkan organisasi dan ketergantungan di antara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan

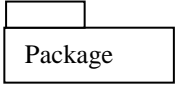
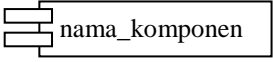
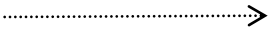


dan ada didalam sistem. Komponen dasar yang biasanya ada dalam suatu sistem adalah sebagai berikut :

- 1) Komponen *user interface* yang menangani tampilan
- 2) Komponen *bussiness procesiing* yang menangani fungsi-fungsi proses bisnis
- 3) Komponen data yang menangani manipulasi data
- 4) Komponen *security* yang menangani keamanan sistem

Komponen lebih terfokus pada penggolongan secara umum fungsi-fungsi yang diperlukan.

berikut Tabel II.3 yang menerangkan simbol-simbol yang ada pada diagram komponen

Tabel II.3 : Diagram Komponen

Simbol	Deskripsi
Package 	<i>Package</i> merupakan sebuah bungkusan dari satu atau lebih komponen
Komponen 	Komponen Sistem
Kebergantungan / <i>dependency</i> 	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai
Antar muka / <i>interface</i> 	Sama dengan konsep <i>interface</i> pada pemrograman berorientasi objek, yaitu sebagai antarmuka komponen agar tidak mengakses langsung komponen
Link 	Relasi antar komponen

(Sumber : Rosa A.S & M. Shalahuddin ; 2011 : 126)

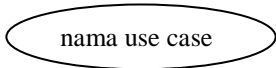

4. Use Case Diagram

Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behaviour*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

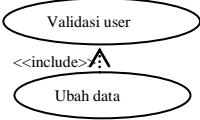
- 1) Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- 2) *Use case* merupakan fungsionalitas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

Berikut Tabel II.4 menerangkan simbol-simbol pada diagram *use case*

Tabel II.4 : Diagram Use case

Simbol	Deskripsi
Use case  nama use case	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>
Aktor / <i>actor</i>  nama aktor	Orang, proses, atau sistem yang lain berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan di buat itu sendiri
Asosiasi / <i>association</i>	Komunikasi antara aktor dan <i>use case</i>

	yang berpartisipasi pada <i>use case</i> , atau <i>usecase</i> memiliki interaksi dengan aktor
<p>Ekstensi / <i>extend</i></p> <p style="text-align: center;"><<extend>></p> <p style="text-align: center;">.....></p>	<p>Relasi usecase 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 inheritance pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan misal</p> <div style="text-align: center;"> </div> <p>arah panah mengarah pada <i>use case</i> yang ditambahkan</p>
<p>Generalisasi / <i>generalization</i></p> <p style="text-align: center;">—————></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 misalnya :</p> <div style="text-align: center;"> </div> <p>Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>
<p>Menggunakan / <i>include / uses</i></p> <p style="text-align: center;"><<include>></p> <p style="text-align: center;">.....></p> <p style="text-align: center;"><<uses>></p> <p style="text-align: center;">—————></p>	<p>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 menjalankannya atau sebagai syarat dijalankan <i>use case</i> ini</p> <p>Ada 2 sudut pandang yang cukup besar mengenai include di usecase</p> <ol style="list-style-type: none"> include berarti <i>use case</i> yang ditambahkan dipanggil saat <i>use case</i> dijalankan misal pada kasus berikut : <div style="text-align: center;"> </div>

	<p>2. include berarti use case yang tambahan akan selalu melakukan pengecekan apakah use case yang di tambahkan telah di jalankan sebelum use case tambahan di jalankan, misal pada kasus berikut :</p>  <p>Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>
--	---

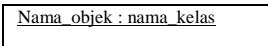

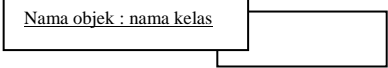
(Sumber : Rosa A.S & M. Shalahuddin ; 2011 : 131)



5. Communication Diagram

Diagram komunikasi mengelompokkan message pada kumpulan diagram sekuen menjadi sebuah diagram. Dalam diagram komunikasi yang dituliskan adalah operasi / metode yang di jalankan antara objek yang satu dengan objek lainnya secara keseluruhan, oleh karna itu dapat di ambil dari jalanya interaksi pada semua diagram sekuen.

Berikut adalah Tabel II.5 yang menerangkan simbol-simbol yang ada pada diagram komunikasi :

Tabel II.5 : Diagram Komunikasi

Simbol	Deskripsi
<p>Objek</p> 	<p>Objek yang melakukan interaksi pesan</p>
<p>Link</p> 	<p>Relasi antar objek yang menghubungkan objek satu dengan lainnya atau dengan dirinya sendiri</p> 

Arah pesan / stimulus 	Arah pesan yang terjadi, jika pada suatu link ada dua arah pesan yang berbeda, maka arah juga digambarkan dua arah pada dua sisi link 
--	---

(Sumber : Rosa A.S & M. Shalahuddin ; 2011 : 140)


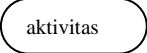


6. Activity Diagram


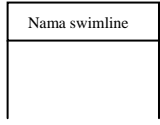
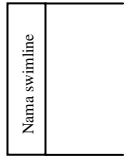
Diagram aktivitas atau activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

- 1) Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
- 2) Urutan atau pengelompokan tampilan dari sistem/user interface dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan
- 3) Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujiannya.

Berikut adalah Tabel II.6 yang menggambarkan simbol-simbol yang ada pada diagram aktivitas :

Tabel II.6 : Diagram Aktivitas

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / decision 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan / join 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu


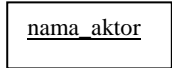
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Swimlane  atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi


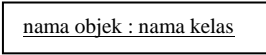

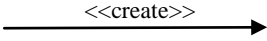
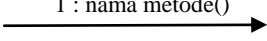
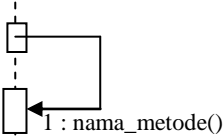
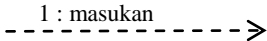
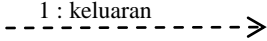
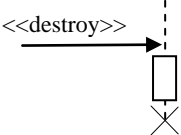
(Sumber : Rosa A.S & M. Shalahuddin ; 2011 : 134)

7. Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Banyaknya diagram objek yang digambarkan adalah sebanyak pendefinisian use case yang memiliki proses sendiri atau yang penting semua use case yang telah didefinisikan interaksi jalanya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak use case yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak. Berikut adalah Tabel II.7 yang menerangkan simbol-simbol yang ada pada diagram sequence :

Tabel II.7 : Diagram Sequence

Simbol	Deskripsi
Aktor  nama aktor atau  tampa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya di nyatakan menggunakan kata benda di awali frase nama aktor

<p>Garis hidup / lifeline</p> 	<p>Menyatakan kehidupan suatu objek</p>
<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi pesan</p>
<p>Pesan tipe create</p> 	<p>Objek yang lain, arah panah mengarah pada objek yang dibuat</p>
<p>Pesan tipe call</p> 	<p>Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri</p>  <p>Arah panah mengarah pada objek yang memiliki operasi / metode, karena ini memanggil operasi / metode maka operasi / metode yang di panggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi</p>
<p>Pesan tipe send</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>
<p>Pesan tipe return</p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian</p>
<p>Pesan tipe destroy</p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy</p>

(Sumber : Rosa A.S & M. Shalahuddin ; 2011 : 138)