

BAB II

TINJAUAN PUSTAKA

II.1 Pengertian Sistem

Istilah sistem berasal dari bahasa Yunani yaitu “*Systema*” yang berarti kesatuan. Dilihat dari sudut katanya sistem berarti sekumpulan objek yang bekerja bersama-sama untuk menghasilkan suatu kesatuan, metode, prosedur, teknik yang digabungkan dan diatur sedemikian rupa sehingga menjadi satu kesatuan yang berfungsi untuk mencapai suatu tujuan.

Menurut Nurdiana R. Lila Sinaga dan Rayendra Khresna MR Brahmana (2005 : 1) *Sistem* dapat didefinisikan sebagai suatu jaringan prosedur yang berinterrelasi dalam satu kesatuan terjadinya aktivitas untuk pencapaian tujuan yang diinginkan. Misalnya aktivitas pembelian barang, akan melibatkan pelaku baik pemasok, persediaan barang (*inventory*), Bank, bagian pembayaran akibat terjadinya transaksi pembelian, penyimpanan dan pembayaran barang yang dipasok tersebut. Masuknya barang dari pemasok sebagai asupan (*input*) akan diproses melalui prosedur yang saling terkait dimulai dari proses pembelian, penyimpanan dan pembayaran barang melalui cara tunai ataukah melalui intermediasi bank. Luaran (*Output*) dari sistem tersebut adalah informasi berupa laporan bahwa barang telah dibeli serta di tempatkan pada gudang persediaan barang, laporan tentang pembayaran barang yang dibeli tersebut. Dalam hubungan ini terjadi inter relasi prosedur yang merupakan aktivitas yang terkait dengan pembelian, penyimpanan dan pembayaran barang untuk mendapatkan informasi

mengenai berapa banyak barang yang dibeli serta tersimpan dalam gudang sebagai persediaan barang, disamping tunai atau giro melalui intermediasi bank (Nurdiana R. Lila Sinaga dan Rayendra Khresna MR Brahmana ; 2005 : 1).

II.2. Pengertian Informasi

Informasi merupakan data yang diolah yang dapat memberikan manfaat bagi penerima agar dapat digunakan sebagai dasar atau kebijaksanaan untuk informasi yang baik dan dibutuhkan.

II.3. Pengertian Sistem Informasi

Sistem informasi merupakan suatu sistem yang melibatkan manusia dalam pengumpulan data, serta proses melibatkan manusia dalam pengumpulan data, serta proses pemeliharaan dan pengolahan data tersebut sebagai informasi dan teknologi informasi untuk menopang dan meningkatkan pengoperasian bisnis sekaligus memberikan solusi terhadap manajemen dan pengguna. Menurut Nurdiana R.Lila Sinaga dan Rayendra Khresna MR Brahmana (2005 : 9).

Manusia yang merupakan stakeholders dari sistem informasi adalah :

1. Pemilik sistem, pihak manajemen yang memiliki kepentingan pembangunan sistem dan yang membiayai pengembangan tersebut.
2. Pengguna sistem, yang menggunakan sistem baik untuk diakses, memasok data, memutakhirkan data, menghapus data atau sekedar membaca data. Data dalam hal ini termasuk yang dikumpulkan, dipelihara, diolah menjadi sumber informasi.

3. Desainer sistem, yang melakukan perancangan sistem dari segi aplikasi perangkat lunak.
4. Pembangunan sistem, yang membangun kebutuhan sistem informasi terutama dari segi perangkat keras sesuai dengan kebutuhan teknologi informasi dan komunikasi. Pembangunan sistem ikut membangun, menguji dan membuat sistem siap dioperasikan.
5. Analisis sistem, yang membangun sistem termasuk manfaat, kelayakan dukungan teknologi informasi, ekonomi dan kualitas sumber daya manusia untuk dibangunnya suatu sistem. Dalam hal ini juga dianalisis struktur pengguna sistem termasuk sistem protokol keamanan sistem dan arsitektur dari sistem yang dikembangkan. Analisis sistem melakukan kebutuhan sistem dengan melakukan kerjasama dengan pemilik, pengguna, desainer dan pengembang sistem.
6. Vendor dan konsultan yang menjual perangkat keras, perangkat lunak dan pelayanan bisnis yang terkait dengan kebutuhan sistem informasi (Nurdiana R. Lila Sinaga dan Rayendra Khresna MR Brahmana ; 2005 : 9).

II.4. Pengertian Sistem Informasi Akuntansi

Sistem Informasi Akuntansi adalah sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan dengan transaksi-transaksi keuangan. Misalnya, salah satu input dari Sistem Informasi Akuntansi pada sebuah toko baju, adalah transaksi penjualan, memproses transaksi dengan mencatat penjualan tersebut kedalam jurnal

penjualan, mengklasifikasikan transaksi dengan menggunakan kode rekening, dan memposting transaksi ke dalam jurnal. Kemudian secara periodik Sistem Informasi Akuntansi akan menghasilkan output berupa laporan keuangan yang terdiri dari Neraca dan Laporan Laba Rugi (Anastasia & Lilis ; 2011 : 4).

Sistem Informasi akuntansi merupakan sebuah sistem informasi yang mengubah data transaksi bisnis menjadi informasi keuangan yang berguna bagi pemakainya (Kusrini & Andi ; 2007 : 10).

Manfaat atau Tujuan dari Sistem Informasi Akuntansi adalah sebagai berikut :

1. Mengamankan harta / kekayaan perusahaan. Meliputi kas perusahaan, persediaan barang dagangan, termasuk aset tetap perusahaan.
2. Mendukung beragam informasi untuk pengambilan. Misalnya, pengelola toko swalayan memerlukan informasi mengenai barang yang diminati oleh konsumen
3. Menghasilkan informasi untuk pihak *external*, setiap pengelola usaha memiliki kewajiban untuk membayar pajak, besarnya pajak yang dibayar tergantung pada omset penjualan (jika pengelola memilih menggunakan norma dalam perhitungan pajaknya) atau tergantung pada laba rugi usaha.
4. Menghasilkan informasi untuk penilaian kinerja karyawan atau divisi. Sistem informasi dapat juga dimanfaatkan untuk penilaian kinerja karyawan atau divisi, sebagai contoh, pengelola toko swalayan dapat memanfaatkan data penjualan untuk menilai kinerja kasir, kasir mana yang lebih cepat dan lebih cermat dalam melayani pelanggan

5. Menyediakan data masa lalu untuk kepentingan audit (pemeriksaan). Data yang tersimpan dengan baik sangat memudahkan proses audit (pemeriksaan), satu hal yang penting, audit bukan eksklusif milik perusahaan publik (Anastasia & Lilis ; 2011 : 5),

Komponen-komponen yang terdapat dalam sistem informasi akuntansi adalah sebagai berikut:

1. Orang-orang yang mengoperasikan sistem tersebut.
2. Prosedur-prosedur, baik manual maupun yang terotomatisasi, yang dilibatkan dalam pengumpulan, pemrosesan dan penyimpanan data aktivitas-aktivitas organisasi.
3. Data tentang proses-proses bisnis.
4. Software yang dipakai untuk memproses data organisasi.
5. Infrastruktur teknologi informasi.

Didalam organisasi, sistem informasi akuntansi berfungsi untuk :

1. Mengumpulkan dan menyimpan aktivitas yang dilaksanakan disuatu organisasi, sumber daya yang dipengaruhi oleh aktivitas-aktivitas tersebut dan para pelaku aktivitas tersebut.
2. Mengubah data menjadi informasi yang berguna bagi manajemen.
3. Menyediakan pengendalian yang memadai (Kusrini & Andi ; 2007 : 10).

Sistem informasi akuntansi merupakan pendukung aktivitas organisasi, yang termasuk pendukung aktivitas organisasi adalah:

1. Infrastruktur perusahaan : akuntansi, hukum, administrasi umum.

2. Sumber daya manusia : perekrutan, pengontrolan, pelatihan dan kompensasi kepada pegawai.
3. Teknologi : peningkatan produk dan jasa (penelitian).
4. Pembelian.

Sementara itu aktivitas utamanya adalah:

1. *Inbound Logistics* : Penerimaan, penyimpanan dan distribusi bahan-bahan masukan.
2. Operasi : aktivitas untuk mengubah masukan menjadi barang atau jasa.
3. *Outbound Logistics* : distribusi produk ke pelanggan.
4. Pemasaran dan Penjualan.
5. Pelayanan : Dukungan purna jual dan *maintenance*.

Dari penjelasan tersebut dapat diambil kesimpulan bahwa akuntansi adalah proses pengidentifikasian, pengukuran dan melaporkan informasi ekonomi untuk memungkinkan adanya pembuatan pertimbangan dan keputusan bagi yang menggunakan informasi (Kusrini & Andi ; 2007 : 10).

II.5. Pengertian Persediaan

Menurut AL. Haryono Jusup (2005 : 99) *Persediaan barang dagang* adalah elemen yang sangat penting dalam penentuan harga pokok penjualan pada perusahaan dagang eceran, maupun perusahaan dagang partai besar. Persediaan berpengaruh terhadap neraca maupun laporan laba/rugi. Dalam neraca sebuah perusahaan dagang atau perusahaan manufaktur, persediaan seringkali merupakan bagian yang sangat besar dari keseluruhan aktiva lancar yang dimiliki perusahaan.

Meskipun demikian, jumlah dan persentasenya berbeda-beda antara perusahaan yang satu dengan yang lainnya. Pada perusahaan tertentu, kadang-kadang persediaan menggambarkan 70% dari keseluruhan aktiva lancar. Angka persentase ini merupakan bukti betapa pentingnya kegiatan pembelian dan penjualan persediaan dalam operasi perusahaan semacam itu. Dalam laporan rugi/laba, persediaan memegang peranan sangat vital dalam penentuan hasil operasi perusahaan untuk suatu periode (AL. Haryono Jusup ; 2005 : 99). Menurut Soemarso S.R (2009 : 384) *Persediaan barang dagang (merchandise inventory)* adalah barang-barang yang dimiliki perusahaan untuk dijual kembali. Untuk perusahaan pabrik, termasuk dalam persediaan adalah barang-barang yang akan digunakan untuk proses produksi selanjutnya. Persediaan dalam perusahaan pabrik terdiri dari persediaan bahan baku, persediaan dalam proses dan persediaan barang jadi (Soemarso S.R ; 2009 : 384).

Menurut Mulyadi (2008 : 553) Dalam perusahaan manufaktur, persediaan terdiri dari : persediaan produk jadi, persediaan produk dalam proses, persediaan bahan baku, persediaan bahan penolong, persediaan bahan habis pakai pabrik, persediaan suku cadang. Dalam perusahaan dagang, persediaan hanya terdiri dari satu golongan, yaitu persediaan barang dagangan, yang merupakan barang yang dibeli untuk tujuan dijual kembali (Mulyadi ; 2009 : 553).

II.6. Sejarah Visual Basic

Diluncurkan *VB.Net 2002* (Pemrograman berbasis Teknologi.Net), dapat dianggap sebagai pertanda berakhirnya pengembangan *Visual Studio 6*, karena

secara *defacto* VB6 sudah tidak dikembangkan lagi oleh *Microsoft*. Walaupun VB6 tidak dikembangkan lagi, tetapi kesinambungan masa hidup VB6 masih dijaga oleh *Microsoft*. Contohnya dengan diluncurkannya *Service Pack 5* untuk *Visual Studio 6* agar bahasa pemrograman yang terdapat pada *visual Studio 6* dapat mengakses *database SQL Server 2005*.

Kini teknologi pemrograman telah dialihkan ke pemrograman berbasis .NET, teknologi yang intinya adalah menulis program agar dapat dijalankan pada piranti lunak maupun keras yang berbeda tanpa harus mengubah program yang sudah dikompilasi. Misalnya, program yang dibuat pada sistem operasi *windows*, nantinya dapat dijalankan pada sistem operasi *Unix/Linux* dan sebaliknya, dimana hal ini dimungkinkan dengan diperkenalkannya konsep *.NET Framework*.

Service Pack yang dikeluarkan oleh *Microsoft* lebih pada penyediaan fasilitas piranti tambahan sebagai media untuk mengakses/membuat hubungan dengan piranti *software* yang baru, misalnya *SQL Server 2005*. jadi walaupun anda menginstal *Service Pack*, tidak berarti *Visual Studio 6* (VB 6) anda akan lebih canggih. itulah yang dimaksud dengan VB 6 tidak dikembangkan lagi oleh *Microsoft* (Harip Santoso ; 2006 : 3).

Visual Basic 6 diluncurkan pada tahun 1998 dan terdiri dari tiga versi, yaitu *Learning Edition*, *Professional Edition*, dan *Enterprise Edition*. Adapun inti peningkatan yang terdapat pada VB 6 dibandingkan dengan versi sebelumnya adalah, peningkatan akses data, penambahan *tools*, dan kontrol baru untuk mengakses database, misalnya *ADODC*.

Awal sejarah *Visual Basic* dimulai pada pertengahan tahun 1991 dengan diluncurkannya VB versi 1.0. yang merupakan pengembangan dari *Quick Basic*, yaitu bahasa pemrograman *Basic* yang dimiliki oleh *Microsoft* untuk penulisan program pada sistem operasi *DOS*. *QuickBasic* masih bersifat teks program, dan belum bersifat grafik. Singkatnya, piranti mouse belum dapat digunakan dalam kaitan dengan interaksi antara pengguna dan aplikasi program

VB 2.0 diluncurkan pada tahun 1992, dimana pada masa itu mulai diperkenalkan *ODBC (Open Database Connectivity)* sebagai metode untuk pengaksesan data, perkembangan VB 2.0 tidak terlalu nampak pada masa itu, karena bahasa pemrograman berbasis *xBase (dbase, foxpro, clipper)* mendominasi penulisan aplikasi program. Disamping itu sistem operasi *DOS* masih bergandengan erat dengan sistem operasi *Netware* yang dimiliki oleh *Novell* sebagai sistem operasi dasar pada aplikasi berbasis *LAN (Local Area Network)*. (Harip Santoso ; 2006 : 4).

II.7. Microsoft SQL Server

Sejarah *SQL Server* berbeda jauh dengan sejarah *Visual Basic*. Bila *Visual Basic* berasal dari pengembangan *QuickBasic* yang juga merupakan produk *Microsoft*, maka *SQL Server* adalah hasil kerja sama antara *Microsoft* dengan *Sybase* untuk memproduksi sebuah software penyimpanan data (database) yang bekerja pada sistem operasi *OS/2*.

Sistem Operasi *OS/2* merupakan sistem operasi baru dari hasil kerja sama antara *Microsoft* dengan *IBM*. Sistem operasi *OS/2* mengenal bentuk-bentuk

perintah DOS, sekaligus memiliki kemampuan *multitasking*. Untuk mendapat pengakuan pasar, maka *Microsoft* bekerja sama dengan *Ashton-Tate* yang telah dikenal dengan produk *dBase*. Sayangnya, kerja sama ini tidak berlangsung lama, karena setelah peluncuran *SQL Server* versi 1.0 pada tahun 1989 kerja sama itupun berakhir.

Peluncuran *SQL Server* 1.0 dilanjutkan dengan peluncuran *SQL Server* versi 1.1 pada tahun 1990. Adapun fitur terpenting dari produk ini adalah dukungan untuk *platform* baru dari sisi client yang dikenal sebagai sistem operasi *Windows 3.1*

Pada tahun 1991, *SQL Server* versi 1.11 diluncurkan dan berisi fasilitas perawatan database. Pada tahun yang sama, *Microsoft* mengembangkan sendiri sistem operasi *multi user* yang dikenal sebagai *Windows NT*, *Microsoft SQL Server* versi 4.2 dirilis pada tahun 1992 dan berisi tool administrator database berbasis *GUI-Windows* (Harip Santoso ; 2006 : 5).

II.8. Database

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* (Budi Raharjo ; 2011 : 3).

II.8.1. Pemodelan Data

Pada perancangan konseptual diperlukan suatu pendekatan yang digunakan untuk menggambarkan hubungan antar data. Hubungan tersebut dapat dinyatakan dalam bentuk E-R, karena model E-R adalah dasar penting dalam merancang database maka akan dijelaskan tentang gambaran tentang model E-R, penjelasan mengenai komponen-komponen yang menyusun model E-R, hingga cara penyusunan model E-R (Abdul Kadir ; 2009 : 30).

II.8.2. Model E-R

Model E-R adalah suatu model yang digunakan untuk menggambarkan data dalam bentuk entitas, atribut dan hubungan antar entitas. Huruf E sendiri menyatakan entitas dan R menyatakan hubungan (dari kata *Relationship*). Model ini dinyatakan dalam bentuk diagram, itulah sebabnya model E-R sering disebut sebagai diagram E-R.

Model E-R melibatkan sejumlah notasi, beberapa notasi dasar dalam model E-R ditunjukkan pada gambar II.1, notasi-notasi tersebut diberikan hanya untuk memberikan suatu pengetahuan dasar (Abdul Kadir ; 2009 : 30).



Gambar II.1 : Sejumlah notasi pada model E-R

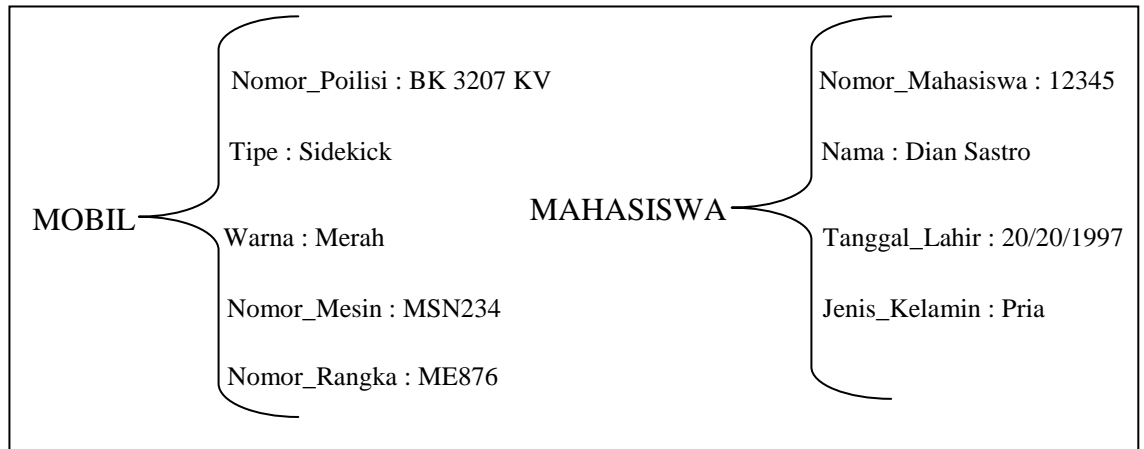
Sumber : Abdul Kadir (2009:31)

1. Entitas

Yang dimaksud dengan entitas adalah sesuatu dalam dunia nyata yang keberadaannya tidak bergantung pada yang lain. Sebagai contoh, setiap pegawai dalam sebuah organisasi adalah sebuah entitas. Entitas dapat berupa sesuatu yang nyata ataupun abstrak (berupa suatu konsep). Secara lebih rinci dijelaskan bahwa entitas dapat berupa seseorang, sebuah tempat, sebuah objek, sebuah kejadian atau suatu konsep.

2. Atribut

Setiap entitas dinyatakan dalam sejumlah atribut. Atribut adalah properti atau karakteristik yang terdapat pada setiap entitas. Sebagai contoh, pada gambar II.2, terdapat entitas MOBIL yang mengandung atribut Nomor_Polisi, Tipe, Warna, Nomor_Rangka dan Nomor_Mesin. Selain itu terdapat entitas MAHASISWA yang mengandung atribut Nomor_Mahasiswa, Nama, Tanggal_Lahir, dan Jenis_Kelamin.

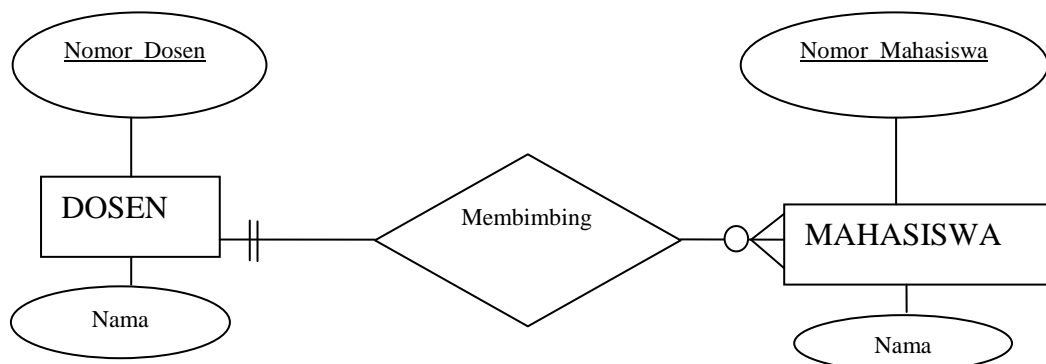


Gambar II.2 : Contoh Entitas dan Atribut

Sumber : Abdul Kadir (2009:32)

3. Hubungan (*Relationship*)

Hubungan (*Relationship*) menyatakan ketertarikan antara beberapa tipe entitas. Sebagai contoh , tipe entitas MAHASISWA dan DOSEN mempunyai hubungan yang mencerminkan bahwa seorang mahasiswa memiliki dosen pembimbing akademis. Gambar II.3 menunjukkan hubungan tersebut.



Gambar II.3 : Contoh Hubungan antara tipe entitas

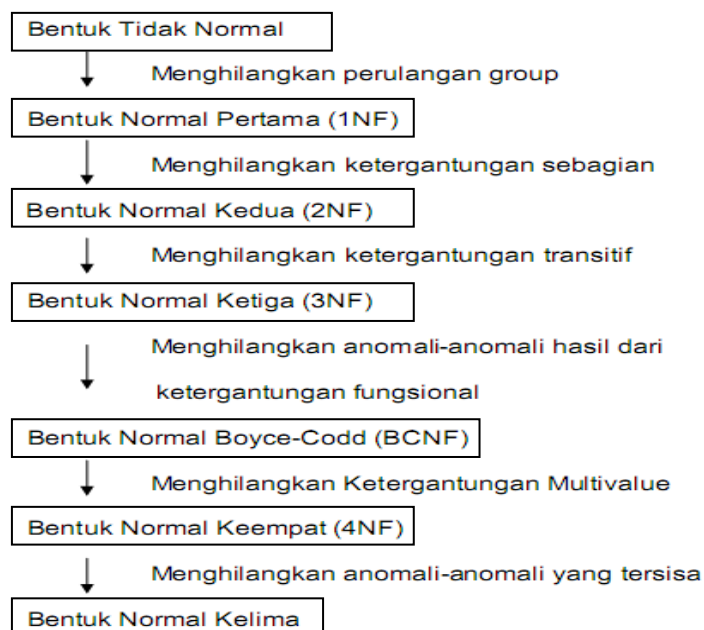
Sumber : Abdul Kadir (2009:45)

II.8.3. Normalisasi

Normalisasi adalah proses yang digunakan untuk menentukan pengelompokan atribut-atribut dalam sebuah relasi sehingga diperoleh relasi yang berstruktur baik. Dalam hal ini yang dimaksud dengan relasi yang berstruktur baik adalah relasi yang memenuhi dua kondisi berikut

1. Mengandung redundansi sedikit mungkin, dan
2. Memungkinkan baris-baris dalam relasi disisipkan, dimodifikasi dan dihapus tanpa menimbulkan kesalahan atau ketidakkonsistenan.

Normalisasi sendiri dilakukan melalui sejumlah langkah. Setiap langkah berhubungan dengan bentuk normal (*normal form*) tertentu. Dalam hal ini yang disebut bentuk normal adalah suatu keadaan relasi yang dihasilkan oleh penerapan aturan-aturan sederhana yang berhubungan dengan dependensi fungsional terhadap relasi tersebut. Gambar II.4 berikut ini akan memperlihatkan hubungan keenam bentuk normal tersebut (Abdul Kadir ; 2009 : 116).



Gambar II.4 : Langkah-langkah dalam normalisasi

Sumber : Abdul Kadir (2009:118)

Contoh dari bentuk–bentuk normalisasi antara lain adalah sebagai berikut, tabel II.1 menerangkan tabel tidak normal:

Tabel II.1 : Contoh Tabel Tidak Normal

isbn	judul	tebal	kpen	nama	alm	kdtopik	topik
1111	wifi	70	aa	asta karya	bdg	t2,t3	Network,security
2222	database	120	bb	bina bakti	pdg	t1,t4	Sql,database
3333	proses basis data	87	cc	cipta jaya	bali	t4,t3	Database,security
4444	keamanan database	69	bb	bina bakti	pdg	t3,t4	Security,database
5555	dasar-dasar LAN	50	ee	news	jkt	t2	network

1. Bentuk normal pertama adalah suatu bentuk relasi dimana atribut bernilai banyak (multivalued attribute) telah dihilangkan sehingga kita menjumpai nilai tunggal. Maka bentuk 1 NF dapat dilihat pada tabel II.1 menerangkan tabel normal pertama :

Tabel II.2 : Contoh Tabel Normal Pertama

isbn	Judul	tebal	kpen	nama	alm	kdtopik	topik
1111	Wifi	70	aa	asta karya	bdg	t2	network
1111	Wifi	70	aa	asta karya	bdg	t3	security
2222	Database	120	bb	bina bakti	pdg	t1	sql
2222	Database	120	bb	bina bakti	pdg	t4	database
3333	proses basis data	87	cc	cipta jaya	bali	t4	database
3333	proses basis data	87	cc	cipta jaya	bali	t3	security
4444	keamanan database	69	bb	bina bakti	pdg	t3	security
4444	keamanan database	69	bb	bina bakti	pdg	t4	database
5555	dasar-dasar LAN	50	ee	news	jkt	t2	network
6666	pengenalan internet	35	aa	asta karya	bdg	t2	network

2. Bentuk Normal kedua.. Syarat menjadi 2 NF harus memenuhi 1 NF dan memenuhi salah satu kondisi berikut :

- a. Kunci primer hanya mengandung 1 atribut (ex:NIP)
- b. Tidak ada atribut bukan kunci hadir di relasi
- c. Setiap atribut bukan kunci bergantung penuh pada sekumpulan atribut yang berfungsi sebagai kunci primer. Maka table tersebut di dekomposisi menjadi :

Tabel II.3 : Contoh Tabel Normal Kedua Tabel ISBN

isbn	Judul	tebal	kpen	nama	alm
1111	Wifi	70	aa	asta karya	bdg
2222	Database	120	bb	bina bakti	pdg
3333	proses basis data	87	cc	cipta jaya	bali

Tabel II.4 : Contoh Tabel Normal Kedua Tabel TOPIK

isbn	kdtopik	topik
1111	t2	network
1111	t3	security
2222	t1	sql
2222	t4	database
3333	t4	database
3333	t3	security
4444	t3	security
4444	t4	database
5555	t2	network
6666	t2	network

3. Bentuk normal tahap tiga (3 NF)

Syarat menjadi 3 NF adalah sebagai berikut

- a. Jika sebuah relasi sudah berada di 2 NF dan tidak dijumpai ketergantungan transitif .

- b. Ketergantungan transitif : ketergantungan fungsional antara 2 atau lebih atribut yang bukan kunci

Pada tabel ISBN nama dan alamat tidak tergantung pada ISBN tapi pada kpen, sedangkan pada table topik tidak tergantung kepada kodetopik , hal ini belum memenuhi 3 NF Maka table tersebut dapat dipecah lagi menjadi :

Tabel II.5 : Contoh Tabel Normal Ketiga Tabel ISBN

isbn	Judul	tebal	kpen
1111	Wifi	70	aa
2222	database	120	bb
3333	proses basis data	87	cc
4444	keamanan database	69	bb
5555	dasar-dasar LAN	50	ee
6666	pengenalan internet	35	aa

Tabel II.6 : Contoh Tabel Normal Ketiga Tabel PENERBIT

kpen	Nama	Alm
aa	asta karya	Bdg
bb	bina bakti	Pdg
cc	cipta jaya	Bali
bb	bina bakti	Pdg
ee	News	Jkt
aa	asta karya	Bdg

Tabel II.7 : Contoh Tabel Normal Ketiga Tabel TOPIK

kdtopik	topik
t2	network
t3	security
t1	sql
t4	database
t4	database
t3	security

Tabel II.8 : Contoh Tabel Normal Ketiga Tabel ISBN_TOPIK

isbn	kdtopik
1111	t2
1111	t3
2222	t1
2222	t4
3333	t4
3333	t3
4444	t3

II.9. Unified Modeling Language (UML)

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 (Rosa A.S & M. Shalahuddin ; 2011 : 118).

UML diaplikasikan untuk maksud tertentu, biasanya antara lain :

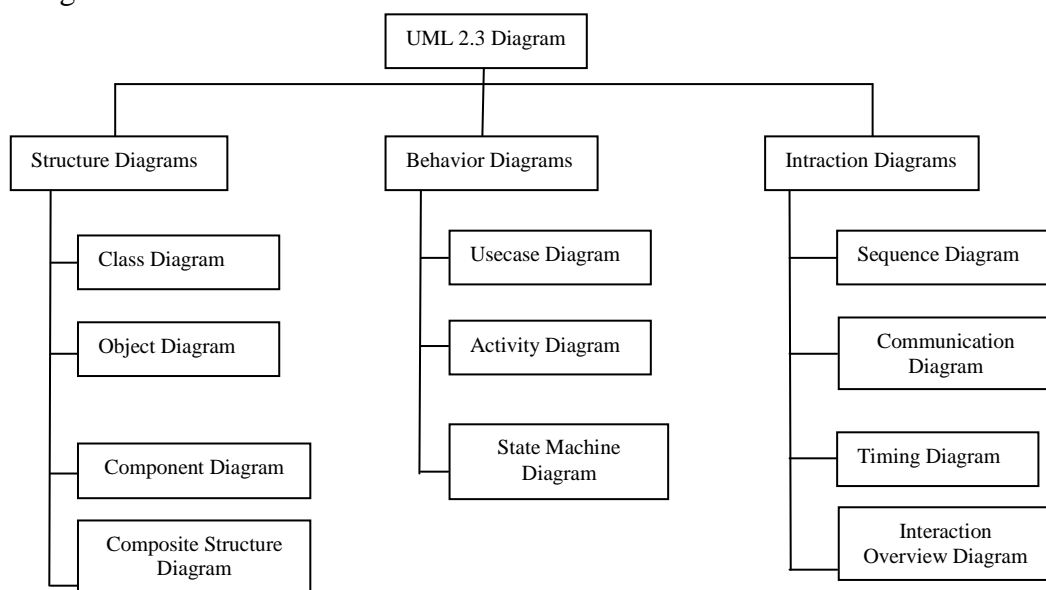
1. Merancang perangkat Lunak.
2. Sarana Komunikasi antara perangkat lunak dengan proses bisnis.

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 (Prabowo Pudjo Widodo & Herlawati ; 2011 : 6),

II.9.1 Diagram-Diagram UML

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.5 di bawah ini



Gambar II.5 : 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 (Rosa A.S & M. Shalahuddin ; 2011 : 120).

A. *Class Diagram*

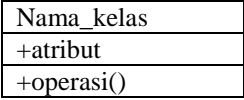

Diagram kelas atau *Class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.


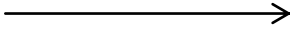
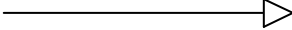
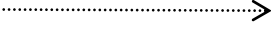
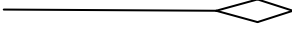
Kelas memiliki apa yang disebut atribut dan metode atau operasi.

- 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.9 menerangkan simbol-simbol pada diagram kelas :

Tabel II.9 : 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>)

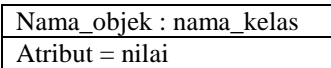

Sumber : Rosa A.S & M. Shalahuddin (2011 : 124)

B. *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.10 menerangkan simbol-simbol diagram objek

Tabel II.10 : Diagram Objek

Simbol	Deskripsi
Objek 	Objek dari kelas yang berjalansaat sistem dijalankan
Link 	Relasi antar objek

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

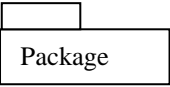
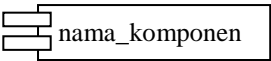
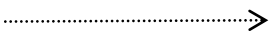


C. *Component Diagram*

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.11 yang menerangkan simbol-simbol yang ada pada diagram komponen

Tabel II.11 : Diagram Komponen

Simbol	Deskripsi
Package  Package	<i>Package</i> merupakan sebuah bungkus dari satu atau lebih komponen
Komponen  nama_komponen	Komponen Sistem
Kebergantungan / <i>dependency</i> 	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai
Antar muka / <i>interface</i>  nama_interface	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)

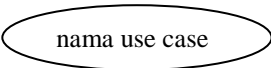

D. Use Case Diagram


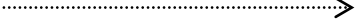
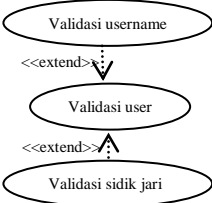

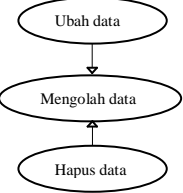

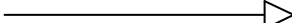
Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behaviour*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang 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*.

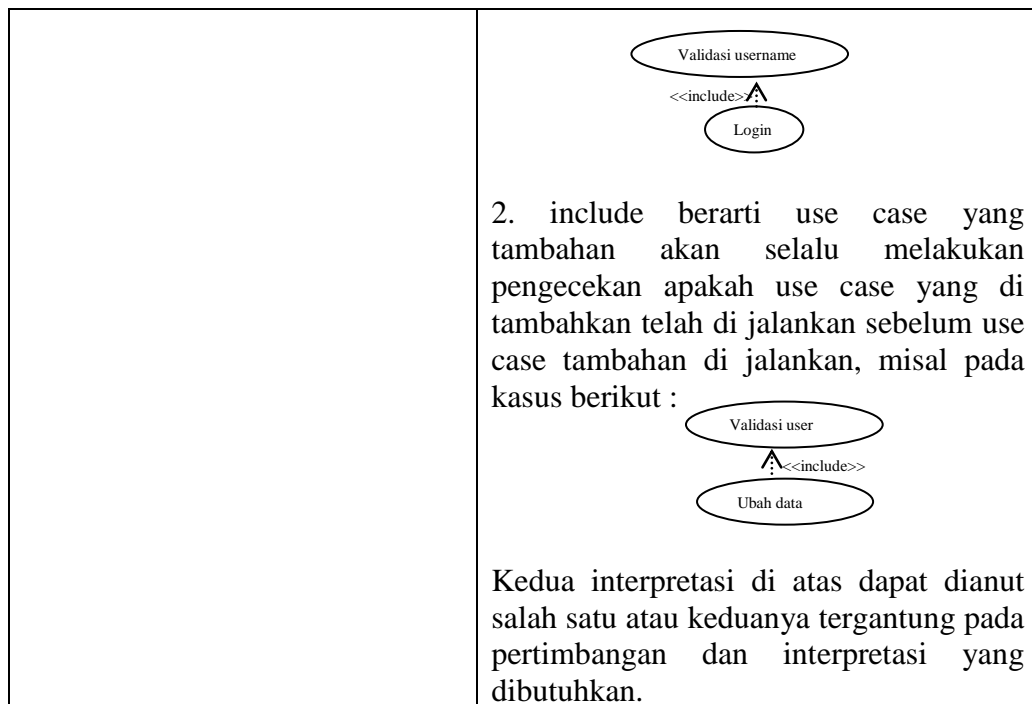
- 1) Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- 2) *Use case* merupakan fungsionalitas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

Berikut tabel II.12 menerangkan simbol-simbol pada diagram *use case*

Tabel II.2 : 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

<p>Asosiasi / <i>association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i>, atau <i>usecase</i> memiliki interaksi dengan aktor</p>
<p>Ekstensi / <i>extend</i></p> <p style="text-align: center;"><<extend>></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>  <p>arah panah mengarah pada <i>use case</i> yang ditambahkan</p>
<p>Generalisasi / <i>generalization</i></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>  <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;"><<uses>></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 menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini</p> <p>Ada 2 sudut pandang yang cukup besar mengenai <i>include</i> di <i>usecase</i></p> <ol style="list-style-type: none"> 1. <i>include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> dijalankan misal pada kasus berikut :



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


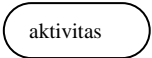



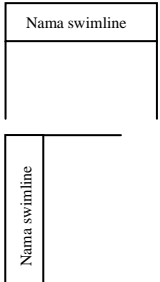
E. *Activity Diagram*

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.13 yang menggambarkan simbol-simbol yang ada pada diagram aktivitas :

Tabel II.13 : 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 / decesion 	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


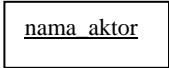

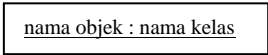

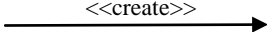
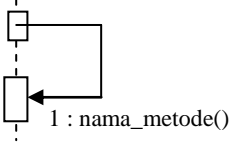
Sumber : Rosa A.S & M. Shalahuddin (2011 : 134)

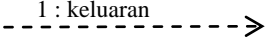
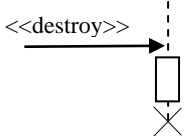
F. Sequence Diagram

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 didefenisikan interaksi jalanya pesan sudah dicakup dapa diagram sekuen sehingga semakin banyak use case yang didefenisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Berikut adalah tabel II.14 yang menerangkan simbol-sombol yang ada pada diagram sekuen :

Tabel II.14 : Diagram Sequence

Simbol	Deskripsi
<p>Aktor</p>  <p>nama aktor</p> <p>atau</p>  <p>nama aktor</p> <p>tampa waktu aktif</p>	<p>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>
<p>Garis hidup / lifeline</p> 	<p>Menyatakan kehidupan suatu objek</p>
<p>Objek</p>  <p>nama objek : nama kelas</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>1 : nama_metode()</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</p>

	dikirim
Pesan tipe return 	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
Pesan tipe destroy 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy

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