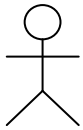
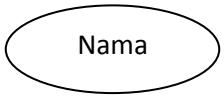
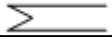
DAFTAR SIMBOL

USE CASE DIAGRAM

Notasi	Keterangan	Simbol
Actor	Actor adalah pengguna sistem. Actor tidak terbatas hanya manusia saja, jika sebuah sistem berkomunikasi dengan aplikasi lain dan membutuhkan input atau memberikan output, maka aplikasi tersebut juga bisa dianggap sebagai actor.	 Aktor
Use Case	Use case digambarkan sebagai lingkaran elips dengan nama use case dituliskan didalam elips tersebut.	
Association	Asosiasi digunakan untuk menghubungkan actor dengan use case. Asosiasi digambarkan dengan sebuah garis yang menghubungkan antara Actor dengan Use Case.	

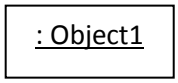
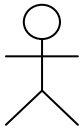
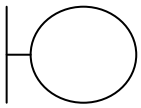
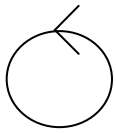
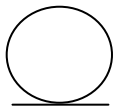
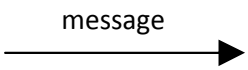
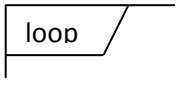
(Sumber : Modul Workshop UML Bab 2)

ACTIVITY DIAGRAM

Notasi	Keterangan	Simbol
Initial	Titik awal, untuk memulai suatu aktivitas.	
Final	Titik akhir, untuk mengakhiri aktivitas.	
Activity	Activity	
Decision	Pilihan Untuk mengambil Keputusan	
Fork/Join	Digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan peralel menjadi satu.	
Rake	Menunjukkan adanya dekomposisi	
Time	Tanda Waktu	
Send	Tanda pengiriman	
Receive	Tanda penerimaan	
Flow final	Untuk mengkhiri suatu aliran	

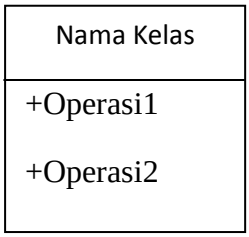
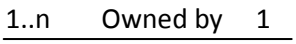
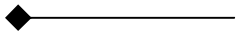
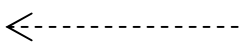
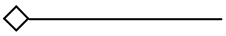
(Sumber : *User Guide Enterprise Architect* 7.0, Modul *Workshop* UML Bab 2)

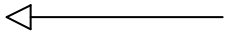
SEQUENCE DIAGRAM

Notasi	Keterangan	Simbol
Object (Partisipan)	*Object* atau biasa juga disebut partisipan merupakan instance dari sebuah class dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah class (kotak) dengan nama objek didalamnya yang diawali dengan sebuah titik koma.	
Actor	*Actor* juga dapat berkomunikasi dengan *object*, maka *actor* juga dapat diurutkan sebagai kolom.	
Lifeline	*Lifeline* mengindikasikan keberadaan sebuah object dalam basis waktu. Notasi untuk *Lifeline* adalah garis putus-putus vertikal yang ditarik dari sebuah objek.	
Activation	*Activation* dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah *lifeline*. *Activation* mengindikasikan sebuah objek yang akan melakukan sebuah aksi.	
Boundary	Boundary terletak di antara sistem dengan dunia sekelilingnya. Semua form, laporan-laporan, antar muka ke perangkat keras seperti printer atau scanner dan antar muka ke sistem lainnya adalah termasuk dalam kategori.	
Control	Control berhubungan dengan fungsionalitas seperti pemanfaatan sumber daya, pemrosesan terdistribusi, atau penanganan kesalahan.	
Entity	*Entity* digunakan menangani informasi yang mungkin akan disimpan secara permanen. *Entity* bisa juga merupakan sebuah tabel pada struktur basis data.	
Massage	*Message*, digambarkan dengan anak panah horizontal antara *Activation*. Message mengindikasikan komunikasi antara *object-object*.	
Self-Message	Self-message atau panggilan mandiri mengindikasikan komunikasi kembali kedalam sebuah objek itu sendiri.	
Loop	Operator loop adalah *fragmen* yang dapat mengeksekusi berulang kali dan penjaga menunjukkan dasar iterasi.	

(Sumber : *Martin Fowler* UML *Distilled* 2005, Modul *Workshop* UML Bab 2)

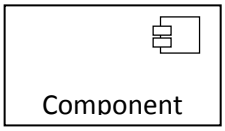
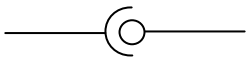
***CLASS* DIAGRAM**

Notasi	Keterangan	Simbol
Class	Class adalah blok - blok pembangun pada pemrograman berorientasi objek. Sebuah class digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari class. Bagian tengah mendefinisikan property/atribut class. Bagian akhir mendefinisikan method-method dari sebuah class.	
Assosiation	Sebuah asosiasi merupakan sebuah relationship paling umum antara 2 *class*, dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 *class*. Garis ini bisa melambangkan tipe-tipe relationship dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah relationship (Contoh: One-to-one, one-to-many, many-to-many).	
Composition	Jika sebuah class tidak bisa berdiri sendiri dan harus merupakan bagian dari class yang lain, maka class tersebut memiliki relasi Composition terhadap class tempat dia bergantung tersebut. Sebuah relationship composition digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi/solid.	
Dependency	Kadangkala sebuah class enggunakan class yang lain. Hal ini disebut dependency. Umumnya penggunaan dependency digunakan untuk menunjukkan operasi pada suatu class yang menggunakan class yang lain. Sebuah dependency dilambangkan sebagai sebuah panah bertitik-titik.	
Aggregation	Aggregation mengindikasikan keseluruhan bagian relationship dan biasanya disebut sebagai relasi "mempunyai sebuah" atau "bagian dari". Sebuah *aggregation* digambarkan sebagai sebuah garis	

	dengan sebuah jajaran genjang yang tidak berisi/tidak solid.	
Generalization	Sebuah relasi *generalization* sepadan dengan sebuah relasi *inheritance* pada konsep berorientasi objek. Sebuah *generalization* dilambangkan dengan sebuah panah dengan kepala panah yang tidak solid yang mengarah ke kelas "*parent*"-nya/induknya.	

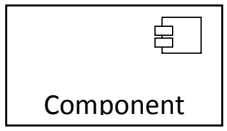
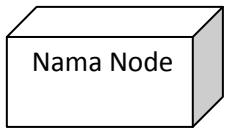
(Sumber : Modul *Workshop* UML Bab 2)

COMPONENT DIAGRAM

Notasi	Keterangan	Simbol
Component	Sebuah komponen melambangkan sebuah entitas software dalam sebuah sistem. Sebuah komponen dinotasikan sebagai sebuah kotak segiempat dengan dua kotak kecil tambahan yang menempel disebelah kirinya.	
Assembly	Sebuah konektor diantara dua komponen yang salah satunya menyediakan layanan ke komponen lain yang memerlukannya.	
Association	Sebuah *association* digambarkan sebagai sebuah garis yang menghubungkan dua komponen yang mengindikasikan jalur komunikasi antara element-elemen software.	
Depedency	Sebuah *dependency* digunakan untuk menotasikan relasi antara dua komponen. Notasinya adalah tanda panah putus-putus yang diarahkan kepada komponen tempat sebuah komponen itu bergantung.	

(Sumber : *User Guide Enterprise Architect* 7.0, Modul *Workshop* UML Bab 2)

DIGRAM *DEPLOYMENT*

Notasi	Keterangan	Simbol
Component	Pada deployment diagram, komponen-komponen yang ada diletakkan didalam node untuk memastikan keberadaan posisi mereka.	
Node	Node menggambarkan bagian-bagian hardware dalam sebuah sistem. Notasi untuk node digambarkan sebagai sebuah kubus 3 dimensi.	
Association	Sebuah association digambarkan sebagai sebuah garis yang menghubungkan dua node yang mengindikasikan jalur komunikasi antara element-elemen hardware.	
Depedency	Sebuah *dependency* digunakan untuk menotasikan relasi antara dua *node*. Notasinya adalah tanda panah putus-putus yang diarahkan kepada *node* tempat sebuah *node* itu bergantung.	

(Sumber : *User Guide Enterprise Architect* 7.0, Modul *Workshop* UML Bab 2)