

## DAFTAR SIMBOL

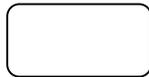
### A. Simbol UML (Unified Modeling Language)

#### 1. Activity Diagram



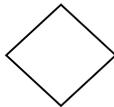
#### INITIAL/START POINT (NODE)

Digunakan untuk status awal aktivitas sistem.



#### ACTIVITY / ACTION

Digunakan untuk aktivitas yang dilakukan oleh sistem, aktivitas biasanya diawali dengan kata kerja.



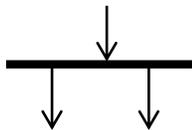
#### DECISION

Digunakan untuk mengambil keputusan. Asosiasi percabangan jika ada pilihan aktivitas lebih dari satu.



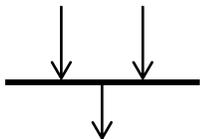
#### END POINT / ACTIVITY FINAL

Digunakan untuk status akhir yang dilakukan sistem.



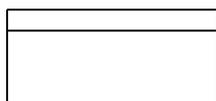
#### FORK/ PERCABANGAN

Digunakan untuk asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.



#### JOIN / PERCABANGAN

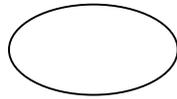
Digunakan untuk asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.



#### SWIMLANE

Digunakan untuk memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi atau sebuah cara untuk mengelompokkan activity berdasarkan actor.

## 2. Use Case Diagram

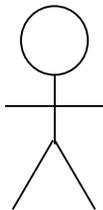


### USECASE

Fungsionalitas yang disediakan oleh sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama usecase.

### AKTOR

Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.

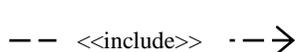


### ASOSIASI / ASSOCIATION

Komunikasi antara aktor dan usecase yang berpartisipasi pada usecase atau usecase memiliki interaksi dengan aktor.



### INCLUDE / USES



Relasi usecase dimana proses bersangkutan akan dilanjutkan keproses yang dituju.

### EKSTENSI / EXTEND



Relasi usecase tambahan sebuah usecase yang ditambahkan dapat berdiri sendiri walau tanpa usecase tambahan itu.

## GENERALISASI/GENERALIZATION



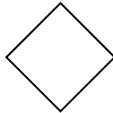
Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah usecase dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.

### 3. ERD (Entity Relationship Diagram)



#### ENTITAS

Suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.



#### RELASI

Menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.



#### ATRIBUT

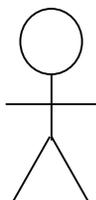
Berfungsi mendeskripsikan karakter entitas (atribut yang berfungsi sebagai *key* diberi garis bawah).



#### LINE

Relasi antara Entitas dan Atribut.

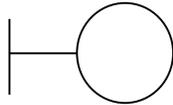
### 4. Sequence Diagram



#### ACTOR

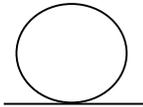
Orang, poses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi

yang akan dibuat itu sendiri, jadi walaupun simbol dari actor adalah gambar orang, tapi actor belum tentu merupakan orang.



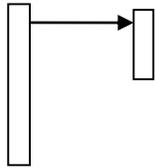
### **INTERFACE atau BOUNDARY OBJECT**

Sebuah obyek yang menjadi penghubung user dengan sistem.



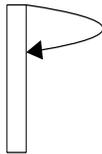
### **ENTITY OBJECT**

Suatu Objek yang berisi informasi kegiatan yang terkait yang tetap dan disimpan kedalam suatu database.



### **OBJECT MESSAGE**

Menggambarkan pesan/hubungan antar obyek, yang menunjukkan urutan kejadian yang terjadi.



### **MESSAGE to SELF**

Menggambarkan pesan/hubungan objek itu sendiri, yang menunjukkan urutan kejadian yang terjadi.



### **LIFELINE**

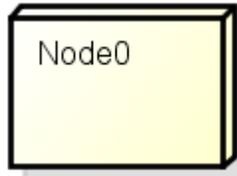
Garis titik-titik yang berhubungan dengan objek, sepanjang lifeline terdapat activation.



### **ACTIVATION**

Mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.

## 5. Deployment Diagram

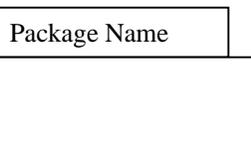


### **NODE**

Sumber daya yang digunakan pada saat aplikasi dijalankan.

### **ASSOCIATION**

Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity.



### **PACKAGE**

Package merupakan sebuah bungkusan dari node atau lebih.

### **KEBERGANTUNGAN/DEPENDENCY**

Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity*.

