

## DAFTAR SIMBOL

### 1. Simbol *Activity Diagram*



#### ***Start Point (Initial Node)***

Merupakan simbol untuk memulai *activity diagram*.



#### ***End Point (Activity Final Node)***

Merupakan simbol untuk mengakhiri *activity diagram*



#### ***Transition***

Menggambarkan aliran perpindahan kontrol antara *activity*.



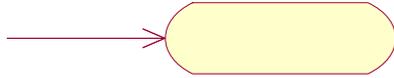
#### ***Activity (Aktivitas)***

Menggambarkan proses bisnis dan dikenal sebagai *activity state*. *Activity* juga merupakan proses komputasi atau perubahan kondisi yang bisa berupa kata kerja atau ekspresi.



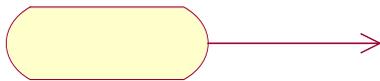
#### ***Swimlane***

Menggambarkan pemisahan atau pengelompokan aktivitas berdasarkan *actor*.



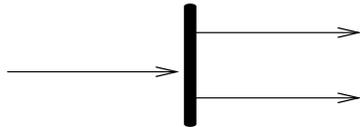
### ***Black Hole Activities***

Adanya masukan dan tidak ada keluaran, biasanya digunakan jika dikehendaki ada 1 atau lebih transisi.



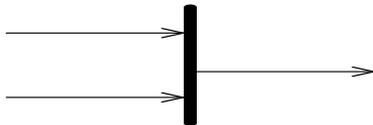
### ***Miracle Activities***

Tidak ada masukan dan ada keluaran, biasanya dipakai pada waktu *start point* dan dikehendaki ada 1 atau lebih transisi.



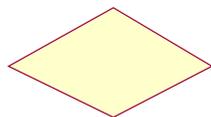
### ***Fork (Percabangan)***

Mempunyai 1 transisi masuk dan 2 atau lebih transisi keluar.



### ***Join (Penggabungan)***

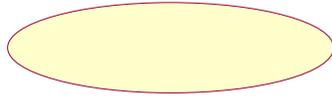
Mempunyai 2 atau lebih transisi masuk dan hanya 1 transisi keluar.



### ***Decision***

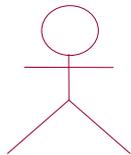
Merupakan cara untuk menggabungkan ketika ada lebih dari 1 transisi yang masuk atau pilihan untuk mengambil keputusan.

## 2. Simbol Use Case Diagram



### *Use case*

Gambaran fungsionalitas dari suatu sistem, sehingga pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.



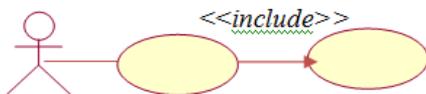
### *Actor*

Sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.



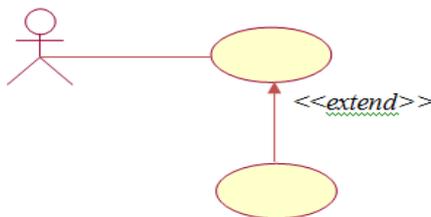
### *Association*

Merupakan abstraksi berupa garis tanpa panah yang menghubungkan antara aktor dan *use case*.



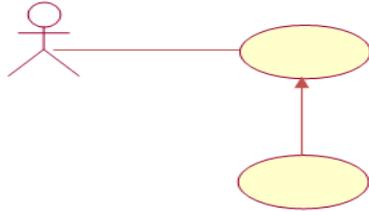
### *Include*

Menunjukkan bahwa suatu *use case* seluruhnya merupakan fungsionalitas dari *use case* lainnya.



### *Extend*

Menunjukkan suatu *use case* merupakan tambahan fungsional dari *use case* lainnya jika suatu kondisi terpenuhi.



### ***Generalization***

Disebut juga *inheritance* (pewarisan), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya.

### ***Packages***

Digambarkan sebagai sebuah direktori yang berisikan model-model elemen. *Packages* digunakan untuk mengorganisasikan sebuah diagram yang besar menjadi beberapa diagram kecil.



## **3. Simbol *Entity Relationship Diagram (ERD)***

### ***Entity***

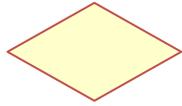
Dapat berupa orang, tempat, objek, atau kejadian yang dianggap penting bagi perusahaan atau instansi, sehingga segala atributnya harus dicatat dan disimpan dalam basis data.



### ***Attribute***

Elemen data yang dimiliki sebuah entitas. Atribut berfungsi mendeskripsikan karakteristik entitas (atribut yang berfungsi sebagai *key* diberi garis bawah).

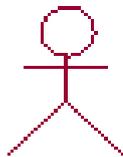




### **Relasi**

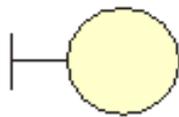
Menggambarkan hubungan yang ada diantara himpunan entitas

## **4. Simbol *Sequence Diagram***



### ***Actor***

Menggambarkan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem.



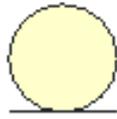
### ***Boundary***

Menggambarkan interaksi antara satu atau lebih *actor* dengan sistem, memodelkan bagian dari sistem yang bergantung pada pihak lain disekitarnya dan merupakan pembatas sistem dengan dunia luar.



### ***Control***

Menggambarkan “perilaku untuk mengatur atau kegiatan mengontrol”, mengkoordinasikan perilaku sistem dan dinamika dari suatu sistem, menangani tugas utama dan mengontrol alur kerja suatu sistem.



**Entity**

Menggambarkan informasi yang harus disimpan oleh sistem (struktur data dari sebuah sistem).



**Object**

Menggambarkan abstraksi dari sebuah entitas nyata/tidak nyata yang informasinya harus disimpan.



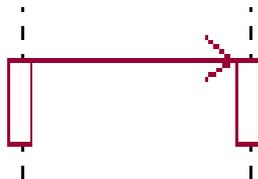
**Activation**

Menunjukkan periode selama suatu *object* atau *actor* sedang melakukan suatu tindakan.



**Message**

Pesan yang dikirim untuk dirinya sendiri.



**Object Message**

Menggambarkan pesan/hubungan antar objek yang menunjukkan urutan kejadian yang terjadi.



**Looping logic**

Menggambarkan dengan sebuah *frame* dengan label *loop* dan sebuah kalimat yang mengindikasikan pengulangan dan *interaction operator loop*.