

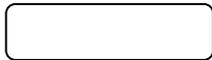
DAFTAR SIMBOL

A. Activity Diagram



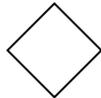
Initial/ Start Point

Status awal aktivitas system



Activity/Action

Yang dilakukan oleh sistem, aktivitas biasanya diawali dengan kata kerja.



Decision

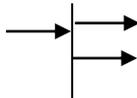
Pengambilan keputusan.

Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.



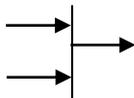
End Point / Activity Final

Status akhir yang dilakukan sistem



Fork

Percabangan dimana jika ada pilihan aktivitas lebih dari satu.



Join

Penggabungan dimana lebih dari satu aktivitas digabungkan.

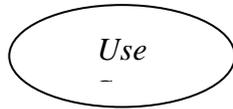


Swimlane

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

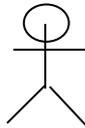
Sumber : Rosa A.S dan M. Shalahuddin

B. 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 diawal frase nama *usecase*.



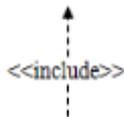
Aktor

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



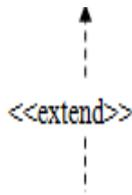
Asosiasi/Association

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



Include/uses/ Menggunakan

Relasi *usecase* tambahan ke sebuah *usecase* dimana *usecase* ini untuk menjalankan fungsinya atau syarat dijalankan *usecase* ini. Relasi *usecase* dimana proses bersangkutan akan dilanjutkan keproses yang dituju.



Ekstensi/Extend

- a. Relasi *usecase* tambahan ke sebuah *usecase* dimana *usecase* yang ditambahkan dapat berdiri sendiri walau tanpa *usecase* tambahan itu.
- b. Mirip dengan prinsip *inheritance* pada pemrograman berorientasi objek.
- c. Biasanya *usecase* tambahan memiliki nama depan yang sama dengan *usecase* yang ditambahkan.



Generalisasi/Generalization

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

Sumber : Martin Fowler, Rosa A.S dan M. Shalahuddin

C. Entity Relational Diagram (ERD)



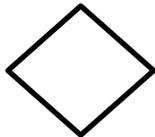
Entitas (*Entity*)

Digunakan untuk memberikan identitas pada entitas yang memiliki label dan nama



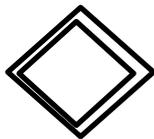
Weak Entity

Entitas yang kemunculannya tergantung dari entitas lain yang lebih kuat



Relasi (*Relationship*)

Digunakan untuk mengetahui jenis hubungan yang ada diantara dua entitas.



Identifying Relationship

Menghubungkan *strong entity* dan *weak entity*



Attribute

Digunakan untuk memperjelas *attribute* yang dimiliki oleh sebuah entitas.



Attribute Multi Value

Attribute yang mempunyai banyak nilai



Attribute Primary Key

Menentukan entitas secara unik



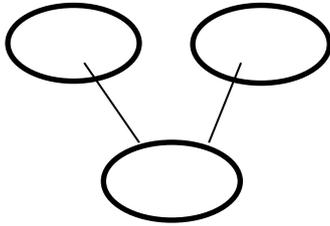
Attribute Derivatif

Attribute yang dihasilkan dari *attribute* lain

Alur



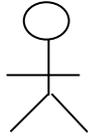
Digunakan untuk menghubungkan *atribute* dengan entitas dan entitas dengan relasi.



Atribute Composit

Suatu atribut yang terdiri dari beberapa atribut yang lebih kecil.

C. *Sequence Diagram*



Aktor

Menggambarkan orang yang berinteraksi dengan sistem



Boundary Class

Menggambarkan sebuah gambaran dari form



Entity Class

Menggambarkan hubungan yang akan dilakukan



Control Class

Menggambarkan penghubungan antara *boundary* dengan tabel

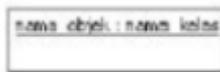
Lifeline

Menggambarkan tempat mulai dan berakhirnya pesan



Line Message

Menggambarkan pengiriman pesan



Objek

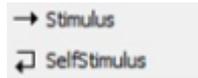
Menyatakan objek yang berinteraksi pesan

Gaya hidup (*lifeline*)

Menyatakan kehidupan suatu objek

Waktu Aktif

Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya (aktor tidak memiliki waktu aktif)



Stimulus

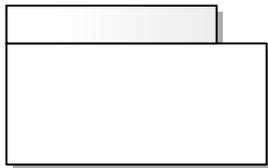
Menyatakan suatu objek mengirimkan pesan untuk menjalankan operasi yang ada pada objek lain

E. Deployment Diagram



Node

Biasanya mengacu pada perangkat keras, perangkat lunak yang tidak dibuat sendiri, jika di dalam *node* disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen



Package

Package merupakan sebuah bungkus dari satu atau lebih *node*



Kebergantungan/dependency

Kebergantungan antar *node*, arah panah mengarah pada *node* yang dipakai



Link

Relasi antar *node*

Sumber : Rosa A.S dan M. Shalahuddin