**Praktikum Rekayasa Perangkat Lunak**

**Modul 3**

**Use Case Diagram**

**Tugas Pendahuluan**

1. Jelaskan pengertian dari Use case dan jenis-jenisnya! (sertakan sumbernya, ex: sumber modul)

Jawaban :

Use case adalah gambaran fungsionalitas dari suatu sistem, sehingga customer atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.

Sebuah use case merupakan sekumpulan urutan dari aksi atau langkah-langkah, termasuk variant-nya (skenario lain yang mungkin terjadi), yang dilakukan oleh sistem untuk memberikan hasil yang dapat diamati dan diukur oleh user. Secara grafis use case dilambangkan dengan bentuk elips.

Sebuah use case hendaknya spesifik namun fungsionalitas yang dilakukan juga tidak boleh terlalu kecil. Untuk menamai sebuah use case dapatmenggunakan kata kerja aktif yang menggambarkan apa yang dilakukan oleh use case tersebut.

Use Case dibagi menjadi dua kategori, yaitu:

1. Use case konkrit

Use case yang dibuat sesuai kebutuhan actor

2. Use case abstrak

Use case yang tidak bisa berdiri sendiri

1. Jelaskan pengertian dari actor! (sertakan sumbernya, ex: sumber modul)

Dalam context use case, user dari suatu use case disebut aktor. Seorang aktor merupakan sekumpulan peran yang berkaitan yang dimainkan pada saat berinteraksi dengan use case yang bersangkutan. Biasanya seorang aktor mewakili peran yang dimainkan oleh manusia, perangkat keras, atau bahkan sistem lain.

Seorang aktor dapat melakukan banyak use case, sebaliknya sebuah use case dapat dilakukan oleh banyak aktor. Selain itu seorang user dapat berperan menjadi lebih dari satu aktor

Aktor hanya dapat berhubungan dengan use case dalam bentuk asosiasi. Sebuah asosiasi antara aktor dan use case mengindikasikan bahwa aktor dan use case saling berkomunikasi dan masing masing memungkinkan untuk mengirim dan menerima pesan. Aktor hanya berinteraksi dengan use case, tetapi tidak memiliki kontrol akan use case tersebut. Secara grafis seorang aktor dilambangkan dengan *stick man.*

Sebuah *actor* mungkin hanya memberikan informasi inputan pada sistem, hanya menerima informasi dari sistem atau keduanya menerima, dan memberi informasi pada sistem. *Actor* dapat digambarkan secara secara umum atau spesifik, dimana untuk membedakannya kita dapat menggunakan *relationship.*

Ada beberapa kemungkinan yang menyebabkan *actor* tersebut terkait dengan sistem antara lain:

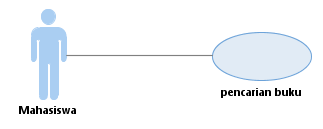
* Yang berkepentingan terhadap sistem dimana adanya arus informasi baik yang diterimanya maupun yang dia inputkan ke sistem.
* Orang ataupun pihak yang akan mengelola sistem tersebut.
* *External resource* yang digunakan oleh sistem.
* Sistem lain yang berinteraksi dengan sistem yang akan dibuat.

Jadi, hanya yang berinteraksi dengan sistem saja yang akan dianggap sebagai aktor. Sementara orang ataupun sesuatu yang tidak berinteraksi dengan sistem tidak bisa dianggap sebagai aktor.

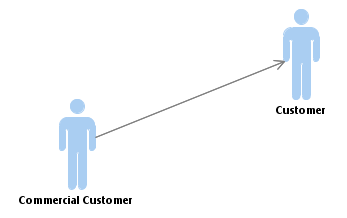
1. Sebutkan dan jelaskan berbagai macam jenis relasi pada use case diagram! (sertakan sumbernya, ex: sumber modul)

Ada beberapa relasi yang terdapat pada *use case diagram*:

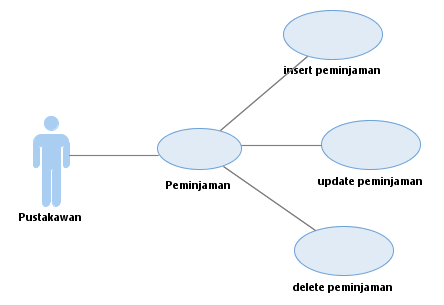
1. *Association*, Menghubungkan elemen dengan proses pertukaran informasi. Dilambangkan dengan garis tegas tanpa panah. Contoh :



1. *Generalization*, Generalisasi disini berarti inheritance (pewarisan), dimana sebuah elemen (use case atau aktor) dapat merupakan spesialisasi dari elemen lainnya. Dilambangkan dengan garis tegas yang memiliki panah tertutup.



1. *Aggregation*, bentuk assosiation dimana sebuah elemen berisi elemen lainnya.



1. *Dependency*,merupakan ketergantungan elemen terhadap elemen lain, dependency dalam diagram use case secara umum memiliki tiga stereotype yang dilambangkan dengan garis putus-putus, antara lain:
2. **<<include>>**, yaitu kelakuan yang harus terpenuhi agar sebuah *event* dapat terjadi, dimana pada kondisi ini sebuah *use case* adalah bagian dari *use case* lainnya. Atau dengan kata laen, use case yang harus dilakukan sebelum melakukan use case yang di include.
3. **<<extends>>**, kelakuan yang hanya berjalan di bawah kondisi tertentu seperti menggerakkan alarm. Jadi sifatnya kondisional, mungkin dilakukan dan mungkin tidak.
4. <<**communicates**>>, mungkin ditambahkan untuk asosiasi yang menunjukkan asosiasinya adalah *communicates association* . Ini merupakan pilihan selama asosiasi hanya tipe *relationship* yang dibolehkan antara *actor* dan *use case*.
5. Rancanglah use case sebuah system ATM!
6. Buat Skenario use case dari use case ATM yang telah anda buat sebelumnya!

Note:

Soal no 4 gunakan **RSA!**