**UML**

Use Case Diagram

UNIFIED MODELLING LANGUAGE(UML)

Bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dengan menggunakan Teknik pemrograman berorientasi objek.

UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari system perangkat lunak.

VIEW DAN DIAGRAM UML

UML dibagi menjadi beberapa view atau pandangan.

View adalah bagian yang simple dari konstruksi pemodelan UML yang merepresentasikan aspek dari sebuah system.

Pada level atas, view dibagi menjadi 3 area:

- a. Klasifikasi structural : mendeskripsikan hubungan segala hal yang ada di dalam system
- b. Kelakuan dinamik : mendeskripsikan kelakuan system atau urutan perubahan yang dialami system
- c. Pengelolaan model : mendeskripsikan keterkaitan organisasi dengan hirarki unit yang ada di dalam sistem

VIEW VS DIAGRAM UML

Area Mayor	View	Diagram
Struktural	Static View View yang tidak bergantung pada waktu	Diagram Kelas
	Use Case View View atau pandangan dari segi fungsionalitas sistem	Diagram Use case
	Implementation View View dari segi komponen implementasi sistem	Diagram Komponen
	Deployment View Pandangan dari segi node tempat komponen di deploy	Diagram deployment

VIEW VS DIAGRAM UML

Area Mayor	View	Diagram
Dinamik	State Machine View View dari segi status yang dialami system berdasarkan objek-objek sistem	Diagram Status
	Activity View View dari segi aktivitas yang dilakukan oleh sistem	Diagram Aktivitas
	Diagram Interaksi	Diagram Sekuen Diagram Kolaborasi

VIEW VS DIAGRAM UML

Area Mayor	View	Diagram
Pengelolaan model (Model Management)	Model-management view View dari segi pengelolaan model sistem	Diagram Kelas

USE CASE

- Dalam membuat sebuah system perlu ditentukan kebutuhan fungsional dan kebutuhan nonfungsional.
- Kebutuhan fungsional : kebutuhan pengguna dan stakeholder sehari-hari yang akan dimiliki oleh system, dimana kebutuhan ini akan digunakan oleh pengguna dan stakeholder.
- Kebutuhan nonfungsional : kebutuhan yang memperhatikan performansi, kemudahan dalam menggunakan system, keuangan, legalitas dan operasional.
- Kebutuhan fungsional akan digambarkan melalui sebuah diagram yang dinamakan **Diagram Use Case**.

USE CASE

- Pemodelan untuk menggambarkan kelakuan (behaviour) system yang akan dibuat.
- Mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan system yang akan dibuat.
- Digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah system dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.
- Bukan menggambarkan tampilan **antarmuka (user interface)**, **arsitektur dari system**, **kebutuhan nonfungsional** dan **tujuan performansi**.

SIMBOL USE CASE

Simbol	Deskripsi
Use Case 	Fungsionalitas yang disediakan system sebagai unit-unit yang saling bertukar pesan antar unit atau actor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case
Aktor 	Orang, Proses atau system lain yang berinteraksi dengan system informasi yang akan dibuat di luar system informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari actor adalah gambar orang, tapi actor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
Asosiasi / association 	Komunikasi antara actor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan actor.

SIMBOL USE CASE

Simbol	Deskripsi
Ekstensi / extend 	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; biasanya use case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan.
Generalisasi / Generalization 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Include 	Relasi Use Case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini.

MENEMUKAN AKTOR

- SIAPA yang akan menggunakan system?
- APAKAH system tersebut akan memberikan NILAI bagi actor?

- Tidak semua actor adalah MANUSIA.

CONTOH PENCARIAN AKTOR

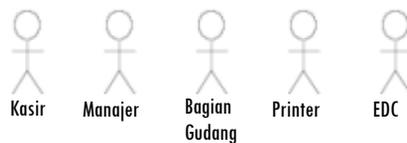
SISTEM PENCATATAN PENJUALAN DI SUPERMARKET

Pertanyaan	Analisis
Siapa sajakah yang berinteraksi dengan system pencatatan penjualan di supermarket ?	<ul style="list-style-type: none"> • Bagian yang akan mencatat penjualan barang • Bagian yang ingin tahu berapa besar keuntungan yang didapatkan • Bagian yang ingin tahu berapa banyak produk yang berkurang
Peran apa saja yang terlibat?	Kasir, Manajer, bagian Gudang.
Nilai apa sajakah yang akan diberikan system kepada actor?	<p>Nilai bagi kasir :</p> <ul style="list-style-type: none"> • Ia akan mendapatkan struk belanja • Lama aktivitas kerja akan terekam kedalam system <p>Nilai bagi manajer</p> <ul style="list-style-type: none"> • Ia perlu mengetahui laporan keuntungan dalam rentang waktu tertentu <p>Nilai bagi bagian Gudang</p> <ul style="list-style-type: none"> • Ia perlu mengetahui produk apa saja yang berkurang.

CONTOH PENCARIAN AKTOR

SISTEM PENCATATAN PENJUALAN DI SUPERMARKET

Pertanyaan	Analisis
Apakah system pencatatan penjualan bergantung pada sesuatu?	Printer <ul style="list-style-type: none"> • Untuk mencetak Struk EDC <ul style="list-style-type: none"> • Untuk menarik sejumlah uang pada account seseorang



MENEMUKAN USE CASE

Untuk menemukan use case, lihat dari sudut pandang actor :

- Informasi apa sajakah yang akan didapatkan actor dari system?
- Apakah ada kejadian dari system yang perlu diberitahukan ke actor?

Kemudian dari sudut pandang system :

- Apakah ada informasi yang perlu disimpan atau diambil dari system?
- Apakah ada informasi yang harus dimasukkan oleh aktor?

USE SCENARIO

- Nama use case yaitu penamaan use case yang menggunakan kata kerja
- Deskripsi yaitu penjelasan mengenai tujuan use case dan nilai yang akan didapatkan oleh aktor.
- Kondisi sebelum (pre-condition) yaitu kondisi-kondisi yang sudah dipenuhi ketika use case sudah dilaksanakan
- Kondisi sesudah (post-condition) yaitu kondisi-kondisi yang sudah dipenuhi ketika use case sudah dilaksanakan.
- Alur Dasar (basic flow) yaitu alur yang menceritakan jika semua aksi yang dilakukan adalah benar atau proses yang harusnya terjadi.
- Alur alternative (Alternatif Flow) yaitu alur yang menceritakan aksi alternative yang berbeda dari alur dasar.



SELESAI
TERIMA KASIH