

# USE CASE

Ama Fariza, S.Kom, M.Kom

The document name can go here Company Proprietary and Confidential

## MATERI 2

- Pendahuluan
- Menangkap persyaratan menggunakan use case
- Membangun blok dari diagram use case
- Aktor
- Dependency
- Generalisasi

COMPANY

## PENDAHULUAN

3

- Untuk memodelkan sistem, aspek yang paling penting adalah menangkap perilaku dinamis.
- Perilaku dinamis berarti perilaku dari sistem pada saat berjalan / beroperasi (running)
- Sehingga perilaku statis saja tidak cukup untuk memodelkan sistem, karena perilaku dinamis lebih penting daripada perilaku statis.
- Pada UML terdapat 5 diagram untuk memodelkan perilaku dinamis dan use case adalah salah satunya.
- Diagram use case terdiri dari faktor internal dan eksternal yang melakukan interaksi, Agen internal dan eksternal disebut aktor
- Diagram use case terdiri dari aktor, use case dan relasinya
- Diagram digunakan untuk memodelkan sistem/sub sistem dari aplikasi. Satu diagram use case menjelaskan fungsional tertentu dari sistem, jadi keseluruhan sistem terdiri dari sejumlah diagram use case.

COMPANY

## MENANGKAP PERSYARATAN MENGGUNAKAN USE CASE

4

- Diagram use case menangkap aspek dinamis dari sistem, definisi ini terlalu umum untuk menggambarkan tujuan use case, karena 4 diagram lain (aktifitas, sequence, collaboration dan statechart) juga mempunyai tujuan yang sama.
- Yang membedakan, diagram use case digunakan untuk menggambarkan persyaratan sistem yang terdiri dari internal dan eksternal yang terlibat
- Ketika sistem dianalisa untuk menentukan fungsionalitas, dipersiapkan use case dan diidentifikasi aktor nya
- Jadi tujuan diagram use case adalah:
  - Menggambarkan persyaratan sistem
  - Mendapatkan *outside view* dari sistem
  - Identifikasi faktor eksternal dan internal yang terlibat dalam sistem
  - Menunjukkan interaksi pada persyaratan oleh aktor

COMPANY

## BUILDING BLOCK DARI DIAGRAM USE CASE

5

- UML menggambarkan sistem real time sehingga sangat penting membuat model konseptualnya dan kemudian memprosesnya secara bertahap.
- Model konseptual dari UML dapat dibangun dengan mempelajari elemen berikut:
  - Building Block UML
  - Aturan untuk menghubungkan building blok
  - Mekanisme umum dari UML
- Building block dari UML dinyatakan sebagai:
  - Things
  - Relationships
  - Diagrams

COMPANY

## (1) THINGS

6

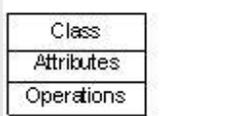
- Things adalah building block dari UML yang paling penting, dapat berupa:
  - Structural
  - Behavioral
  - Grouping
  - Annotational

COMPANY

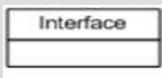
## STRUCTURAL THINGS #1

7

- Merupakan elemen konseptual dan fisik
- **Class**: kumpulan obyek yang mempunyai tanggung jawab yang sama



- **Interface**: kumpulan operasi yang menentukan tanggung jawab dari sebuah class.



COMPANY

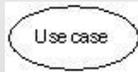
## STRUCTURAL THINGS #2

8

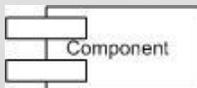
- **Collaboration**: menyatakan interaksi antar elemen.



- **Use case**: menyatakan kumpulan aksi-aksi yang dibentuk oleh sistem untuk tujuan khusus.



- **Component**: menggambarkan bagian fisik dari sistem.

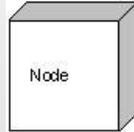


COMPANY

## STRUCTURAL THINGS #3

9

- **Node:** menyatakan elemen fisik yang tersedia pada saat aplikasi berjalan.



COMPANY

## BEHAVIOUR THINGS

10

- Terdiri dari bagian dinamis dari model UML
- **Interaction:** perilaku yang terdiri dari sekumpulan pertukaran pesan antar elemen untuk menyelesaikan pekerjaan tertentu.



- **State machine:** menyatakan urutan state dari obyek yang melalui event. Event adalah faktor eksternal yang mengubah state.

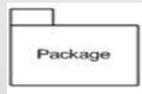


COMPANY

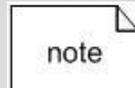
## GROUPING THINGS

11

- Menyatakan mekanisme untuk mengelompokkan elemen dari model UML
- **Package**: menggabungkan structural things dan behaviour things



- **Annotational things**: menyatakan mekanisme pemberian catatan, deskripsi dan komentar dari elemen model UML, menggunakan **Note**.



COMPANY

## (2) RELATIONSHIP # 1

12

- Merupakan building block UML yang sangat penting juga
- Menunjukkan bagaimana elemen dihubungkan dan asosiasi ini menggambarkan fungsionalitas dari aplikasi
- **Asosiasi**: kumpulan link yang menghubungkan elemen dari model UML, juga menggambarkan berapa banyak obyek yang ber-relasi.

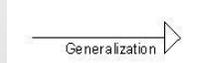


COMPANY

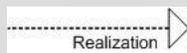
## (2) RELATIONSHIP #2

13

- **Generalisasi:** menyatakan relasi yang menghubungkan elemen tertentu dengan elemen generalisasi. Untuk menggambarkan relasi pewarisan (inheritance) dari obyek.



- **Realisasi:** menyatakan relasi antara 2 elemen yang berhubungan. Satu elemen menyatakan beberapa tanggung jawab yang tidak diimplementasikan dan elemen lain mengimplementasikannya. Relasi ini terjadi pada interface.



COMPANY

## (3) UML DIAGRAM

14

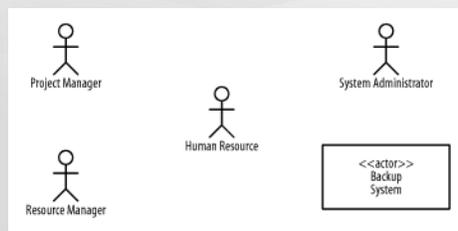
- UML terdiri dari 9 diagram yaitu:
  1. Diagram Class
  2. Diagram Obyek
  3. Diagram Use Case
  4. Diagram Sequence
  5. Diagram Collaboration
  6. Diagram Aktifitas
  7. Diagram Statechart
  8. Diagram Deployment
  9. Diagram Komponen

COMPANY

## AKTOR

15

- Aktor adalah user atau sistem eksternal yang berinteraksi
- Contoh: sistem manajemen proyek melibatkan beberapa user yaitu manajer proyek, manajer sumber daya, sumber daya manusia dan administrator sistem. Semua user tsb adalah aktor
- Aktor eksternal yang berinteraksi dengan sistem, berupa orang atau sistem lain yang mempunyai tujuan dan interaksi dengan sistem



COMPANY

## USE CASE UNTUK MODEL USE CASE #1

16

- Diagram use case adalah analisa persyaratan sistem level tinggi, jadi fungsionalitas sistem tergambar dalam use case.
- Untuk merencanakan diagram use case harus diidentifikasi
  - Fungsionalitas yang dinyatakan dengan use case
  - Aktor
  - Relasi antara use case dan aktor
- Pedoman untuk membuat diagram use case yang efektif
  - Nama use case sangat penting, mengidentifikasi bagaimana fungsi dijalankan.
  - Berikan nama yang tepat untuk aktor
  - Menunjukkan relasi dan dependensi secara jelas
  - Jangan memasukkan semua tipe relasi karena tujuan utama dari diagram adalah mengidentifikasi persyaratan

COMPANY

## USE CASE UNTUK MODEL USE CASE #2

17

- Contoh diagram use case untuk sistem manajemen pemesanan. Terdapat 3 use case (Order, SpecialOrder and NormalOrder) dan satu aktor yaitu customer. Use case *SpecialOrder* dan *NormalOrder* merupakan extend dari use case *Order*. Aktor *Customer* berada diluar sistem sebagai user eksternal.

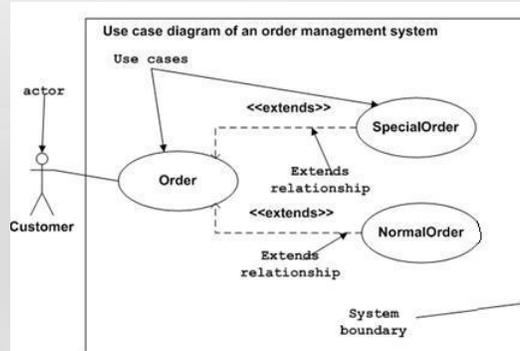


Figure: Sample Use Case diagram

COMPANY

## DEPENDENCY

18

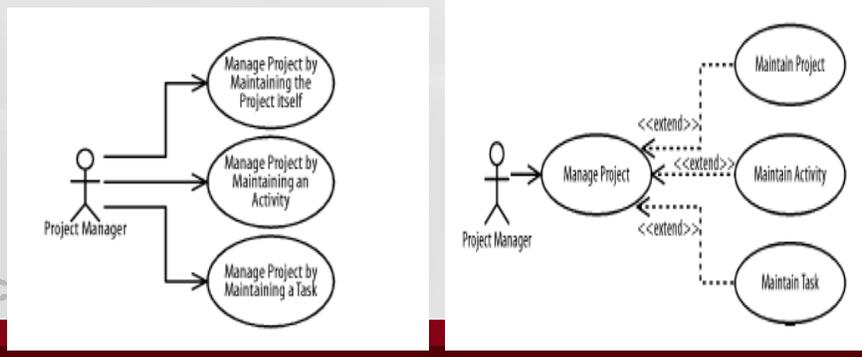
- Sebuah model bisa terdiri dari beberapa use case, bagaimana kita mengorganisasi use case yang menyatakan apa yang sistem kerjakan?
- Jenis dependency ada 2 yaitu include dan extend

COMPANY

## DEPENDENCY: INCLUDE

19

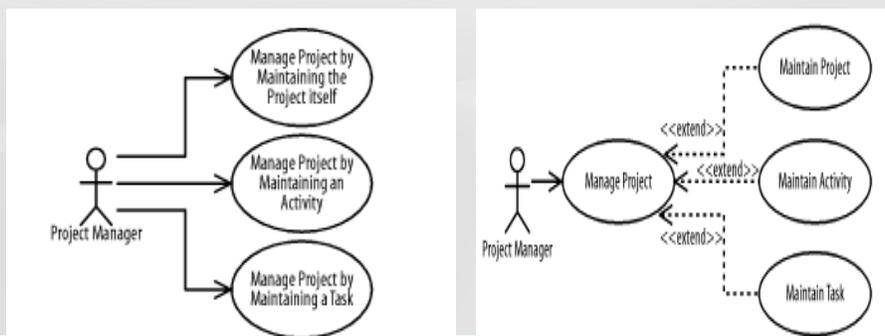
- Aktifitas project managers, resources managers, dan system administrators berinteraksi pada project management system. Use case dengan perilaku umum.
- Dependency include menyatakan use case basis akan memasukkan atau memanggil use case lain.



## DEPENDENCY: EXTEND

20

- Dependency extend dari satu use case (disebut use case extend) ke use case lain (disebut use case basis) mengindikasikan use case extend akan menyisipkan dan menambah use case basis.



## GENERALISASI

21

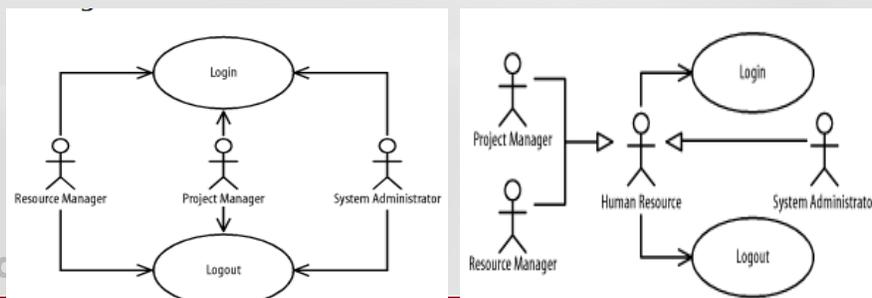
- Aktor ada kemungkinan sama dalam satu sistem, misalnya manajer proyek, manajer sumber daya dan administrator sistem bisa masuk dan keluar dari sistem manajemen proyek.
- Use case ada kemungkinan sama secara fungsionalitas yang dilakukan user, contohnya, manajer proyek memasukkan status proyek dengan 2 cara yaitu membangkitkan laporan ke printer dan ke website pada server proyek.
- Untuk mengorganisasi aktor dan use case yang sejenis digunakan generalisasi. Terdapat 2 jenis generalisasi yaitu generalisasi aktor dan generalisasi use case.

COMPANY

## GENERALISASI AKTOR

22

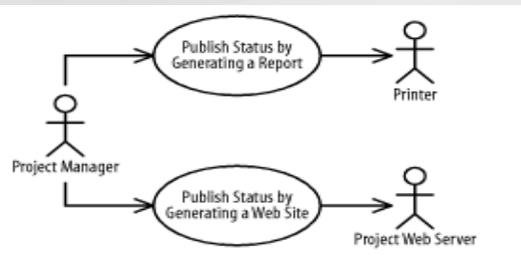
- project managers, resource managers dan system administrators dapat log in dan log out project management system.
- Generalisasi aktor membentuk satu aktor human resource yang melakukan use case Login and Logout. Project managers, resource managers dan system administrators adalah human resources.



## GENERALISASI USE CASE

23

- project manager melaporkan status proyek dengan 2 cara: membangkitkan laporan ke printer atau menampilkan pada website.
- Penerbitan status proyek dan semua proses yang terlibat dalam pengumpulan dan penyiapan data untuk publikasi adalah hal yang umum dalam use case
- Generalisasi use case dari use case spesifik (spesifikasi use case) ke use case yang lebih general (generalisasi) menyatakan bahwa use case spesifik menerima atau mewariskan aktor, urutan perilaku dan ekstensi dari use case yang lebih general dan anggota dari use case yang spesifik dihasilkan dari anggota yang lebih general.



COMPANY