

Use Case Diagram

Analisis Desain Sistem D





Nama Anggota

Anita Puspitasari – 20081010015

Alif Fermansyah – 20081010030

Aktavan Karunia Rahman – 20081010036

Nora Mina – 20081010042

Muhammad Daffa Fitriansyah – 20081010095

Fahrul Firmansyah – 20081010099

Putra Dwi Wira Gardha Yuniahans – 20081010103

Nabila Sya'bani Wardana – 20081010114

Azka Avicenna R – 20081010115

Abdul Wasi Al-Afif – 20081010165

Raja Aulia Edward – 20081010166

Chycik Ayu Winata – 20081010197

Ardiana Deka Maharani – 20081010243

Syalum Marsya Pruista – 20081010244

Mendefinisikan Kebutuhan

Dalam membuat sebuah sistem, langkah awal yang perlu dilakukan adalah menentukan kebutuhan.

Terdapat dua jenis kebutuhan :

1. Kebutuhan fungsional adalah kebutuhan pengguna dan stakeholder sehari-hari yang akan dimiliki oleh sistem, dimana kebutuhan ini akan digunakan oleh pengguna dan stakeholder.
2. Kebutuhan nonfungsional adalah kebutuhan yang memperhatikan hal-hal berikut yaitu performansi, kemudahan dalam menggunakan sistem, kehandalan sistem, keamanan sistem, keuangan, legalitas, dan operasional.

Kebutuhan fungsional akan digambarkan melalui sebuah diagram yang dinamakan diagram use case.

USE CASE DIAGRAM

- Diagram Use Case menggambarkan apa saja aktifitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. Yang menjadi persoalan itu apa yang dilakukan bukan bagaimana melakukannya.
- Diagram Use Case dekat kaitannya dengan kejadian-kejadian. Kejadian (scenario) merupakan contoh apa yang terjadi ketika seseorang berinteraksi dengan sistem.
- Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, mengcreate sebuah daftar belanja, dan sebagainya.
- Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Manfaat Use Case Diagram

Singkatnya, diagram use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut

Diagram Use Case berguna dalam tiga hal :

1. Menjelaskan fasilitas yang ada (requirements). Use Case baru selalu menghasilkan fasilitas baru ketika sistem dianalisa, dan design menjadi lebih jelas.
2. Komunikasi dengan klien. Penggunaan notasi dan simbol dalam diagram Use Case membuat pengembang lebih mudah berkomunikasi dengan kliennya.
3. Membuat test dari kasus-kasus secara umum. Kumpulan dari kejadiankejadian untuk Use Case bisa dilakukan test kasus layak untuk kejadiankejadian tersebut.

KOMPONEN USE CASE DIAGRAM

1. Aktor adalah seseorang atau apa saja yang berhubungan dengan sistem yang sedang dibangun.
 - Aktor sebaiknya diberinama dengan kata benda.
 - Ketika memberi nama actor, gunakan nama peranan dan jangan nama posisi. Seorang individu dapat memainkan beberapa peranan.
 - Dalam UML direpresentasikan dengan notasi berikut ini:



KOMPONEN USE CASE DIAGRAM

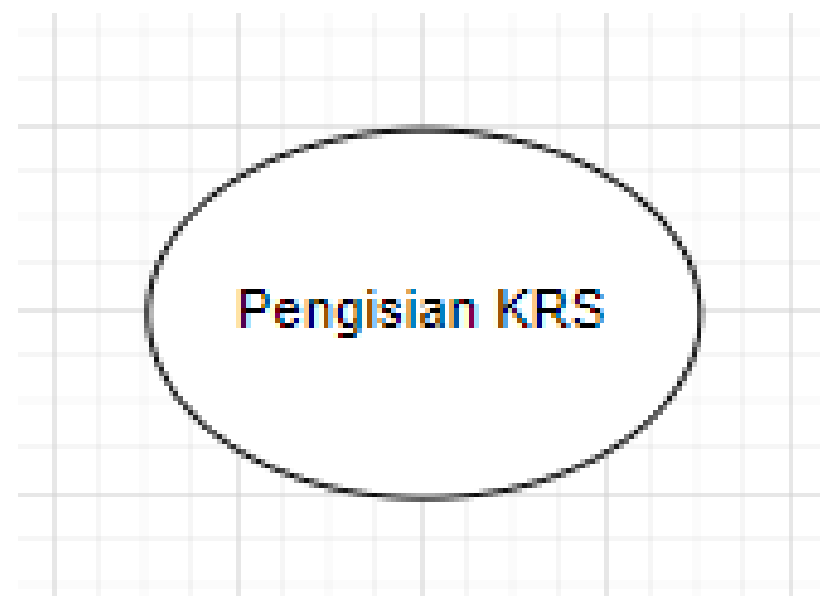
Tipe-Tipe Aktor

- Pengguna sistem. Merupakan actor secara fisik atau seorang pengguna, gambaran secara umum dan selalu ada pada setiap sistem
- Sistem yang lain dan berhubungan dengan sistem yang dibangun. Misalkan pada sebuah sistem Informasi Puskesmas memerlukan koneksi dengan aplikasi sistem yang lain, semisal SIM rumah sakit. Maka dalam kasus ini, SIM rumah sakit adalah actor.
- Waktu. Dapat menjadi actor jika seiring perjalanan waktu dapat memicu event/kejadian dalam sistem.

KOMPONEN USE CASE DIAGRAM

2. Use Case. Adalah bagian fungsionalitas tingkat tinggi yang disediakan oleh sistem.

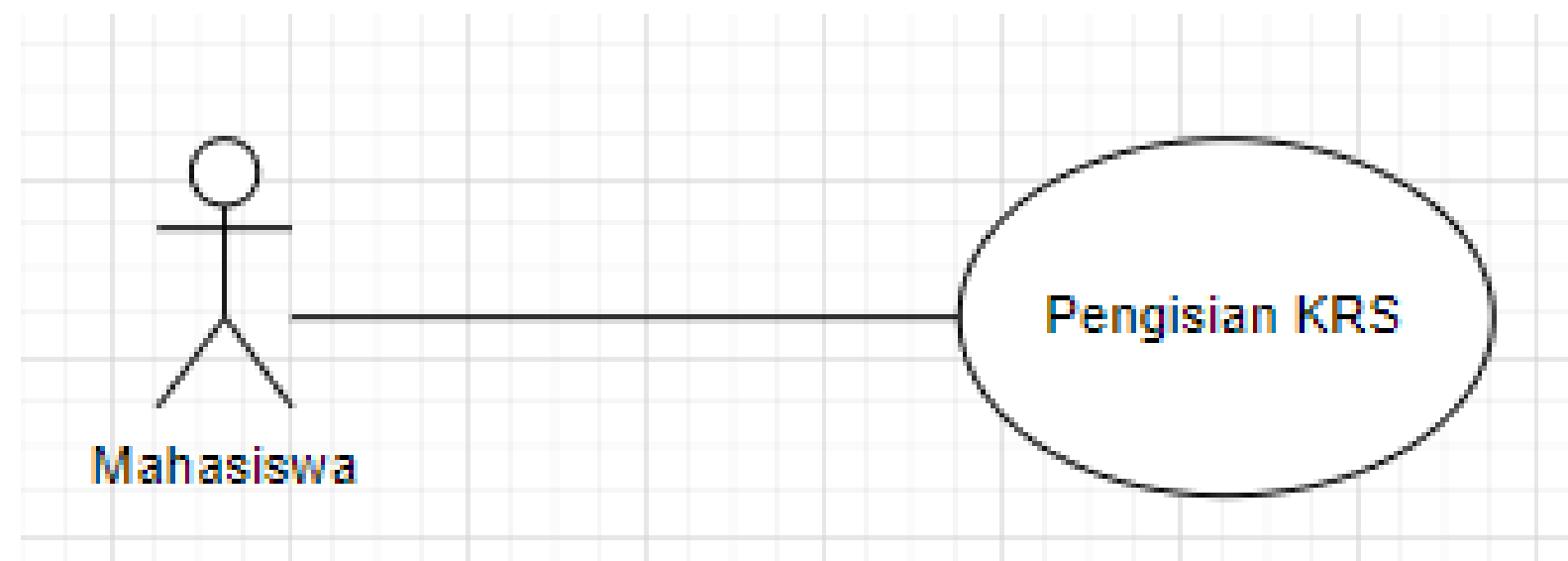
- Dengan kata lain, use case menggambarkan bagaimana seseorang menggunakan sistem.
- Use Case dalam UML dinotasikan dengan simbol:



KOMPONEN USE CASE DIAGRAM

3. Relasi yang menghubungkan antara actor dan use case

- Arah panah menunjukkan siapa yang mengawali komunikasi.
- Dengan mengecualikan use case dalam relasi include dan relasi extend, setiap use case harus diinisialisasi oleh actor
- Relasi dinotasikan seperti gambar berikut :





RELASI DALAM USE CASE

- Association
- Generalization
- Dependency
- Aggregation

ASSOCIATION

Teknik mengidentifikasi interaksi yang dilakukan oleh aktor tertentu dengan use case tertentu pula. Hal ini digambarkan dengan garis antara aktor terhadap use case tersebut. Asosiasi ini biasanya ditandai dengan garis 1 arah (ditandai dengan garis anak panah), jika terjadi komunikasi 2 arah, maka tanda panah tidak akan diperlukan.

Generalization

Teknik mengidentifikasi relasi antara 2 aktor can juga 2 Use case, dimana salah satunya akan meng- inherit dan juga menambah atau override sifat dari perangkat lainnya. Untuk teknik penggambarannya menggunakan garis bermata panah yang kosong. Garis akan diambil dari yang meng- inherit kemudian mengarah ke yang di- inherit.



Dependency

Dependency ini terbagi menjadi 2 macam, yaitu :

- Include
- Extend

Include

- Berfungsi untuk mengidentifikasi hubungan antara 2 use case, dimana use case yang satu akan memanggil use case yang lainnya.
- Apabila ada beberapa use case dengan aktifitas yang sama, maka bagian aktifitas tersebut. akan dijadikan aktifitas tersendiri, dengan relasi dependensi use case semula ke use case yang baru.
- Biasanya digambarkan dengan garis putus-putus dengan mata panah notasi include yang pada garis. Kemudian arahkan mata panah sesuai dengan arah yang memanggil.



Extend

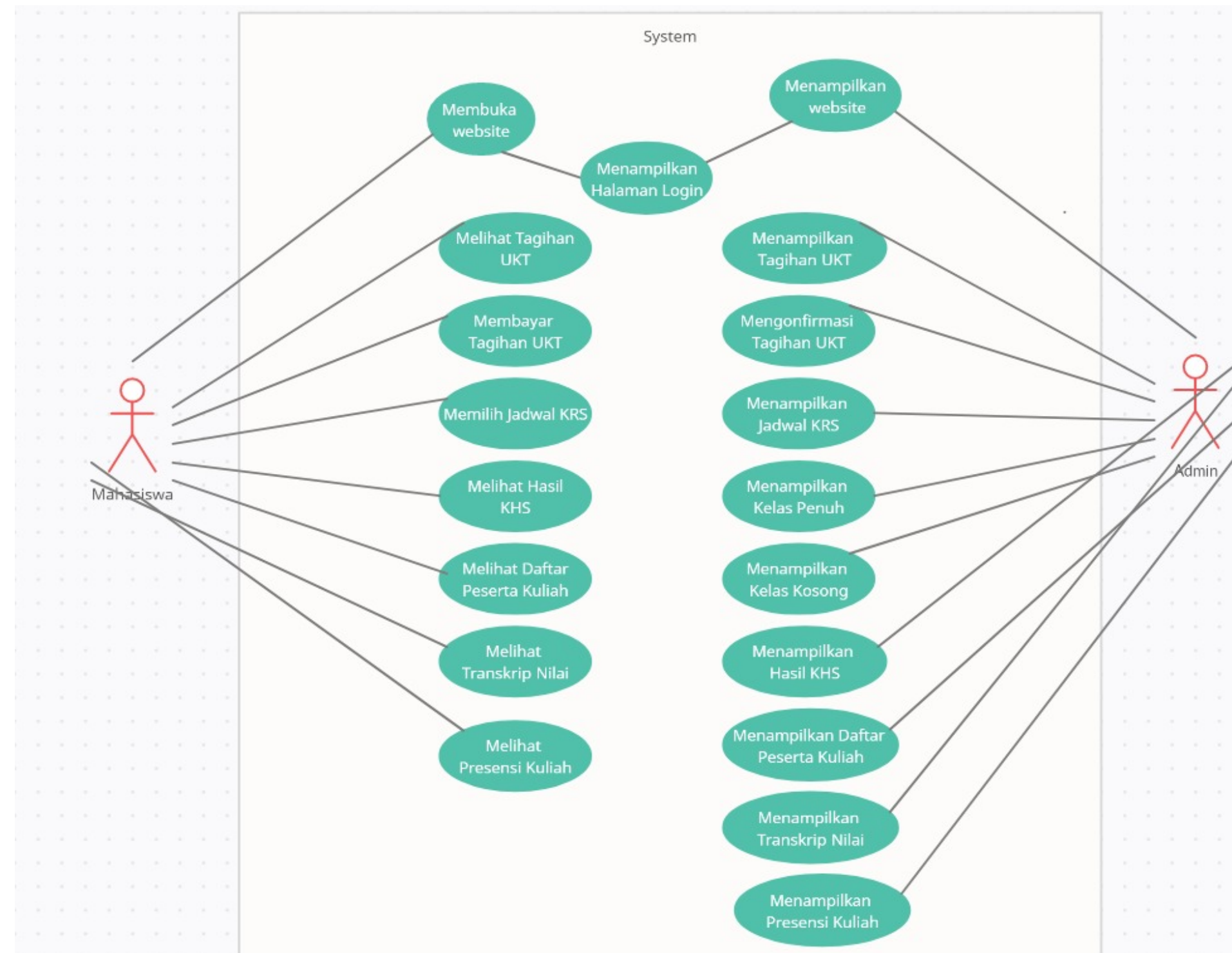
- Apabila pemanggilan, memerlukan kondisi tertentu maka akan berlaku dependensi.
- Konsep extend ini berbeda dengan konsep extend dalam program Java.
- Dependensi ini akan digambarkan dengan dependensi include namun dengan arah panah yang berlawanan.



Studi Kasus

Use Case Diagram pada Siamik UPN "VETERAN" JAWA TIMUR.

Use Case Diagram





Terima Kasih

analisis audiens yang menjelaskan siapa audiens yang dituju oleh proyekmu dan alasannya.