# MODUL 9 :

# Specifying Control

## Daftar Isi

## 9.1. Pengantar Specifying Control

Notasi UML telah diperkenalkan untuk memodelkan persyaratan aplikasi *(use case),* struktur statik dari sebuah aplikasi *(class diagram),* cara obyek berinteraksi *(interaction diagram)* dan prosedur atau operasi yang dilakukan obyek-obyek *(activity diagram)*. Aspek penting lainnya dari sebuah aplikasi yang harus dimodelkan adalah cara sebuah obyek merespon *event* yang bervariasi tergantung pada berlalunya waktu dan event-event yang sudah terjadi. Contohnya adalah *vending machine* yang tidak akan mengeluarkan barang sampai jumlah uang yang sesuai dengan barang yang diinginkan dimasukkan. Variasi perilaku (*behaviour)* ini ditentukan oleh *state* dari *vending machine* – yang bergantung pada cukup tidaknya uang yang dimasukkan untuk membayar barang yang dipilih. Pada kenyataannya bisa saja meskipun uang yang dimasukkan sesesuai dengan harga barang yang dipilih tetapi tidak dapat mengeluarkan barang tsb karena barangnya tidak ada dalam stock. Penting untuk memodelkan *state dependent variation* pada behaviour seperti ini karena mewakili batasan-batasan pada cara suatu sistem seharusnya bertindak.

Obyek dapat memiliki variasi dalam perilaku mereka bergantung pada *state*nya. Variasi perilaku ini menggambarkan batasan batasan-batasan penting pada cara obyek bertindak, dan ditentukan oleh persyaratan sistem. UML menggunakan *state machine* untuk memodelkan *state* dan ketergantungan perilaku *state* untuk obyek dan interaksi.

## 9.2. State dan Event

Semua obyek memiliki state. Current State dari suatu obyek adalah hasil dari event-event yang telah terjadi terhadap obyek dan ditentukan oleh *current value* dari atribut-atribut obyek dan hubungan yang dimiliki dengan obyek lainnya. Misalnya kelas *StaffMember* memiliki atribut *StartDate* yang menentukan apakah objek StaffMember dalam status percobaan sementara tanggal seorang staf member mulai bekerja di Agate menentukan kapan periode percobaan berakhir (misalnya setelah 6 bulan).

Sebuah state menggambarkan suatu kondisi tertentu sehingga elemen yang dimodelkan (mis. objek ) dapat menempati selama periode waktu tertentu sambil menunggu beberapa event atau trigger.

*Possible State* yang dapat ditempati oleh sebuah objek dibatasi oleh kelasnya.

Objek dari beberapa kelas hanya memiliki satu state yang mungkin. Contohnya pada studi kasus Agate obyek *Grade* dapat ada ataupun tidak ada. Jika ada maka tersedia untuk digunakan dan jika tidak ada maka ini berarti tidak tersedia. Obyek dari kelas ini hanya memiliki satu state yang dapat diberi nama *Available*.

Obyek dari kelas ini memiliki lebih dari satu state yang mungkin. Contohnya obyek dari kelas GradeRate dapat berada oada satu dari beberapa state. State tsb mungkin *Pending,* jika tanggal saat ini lebih awal dari tanggal mulainya*. Active*, jika tanggal saat ini sama dengan atau lebih dari start date tetapi lebih awal dari tanggal selesai (diasumsikan tanggal selesai lebih belakangan dari tanggal mulai). Atau *lapsed* jika tanggal saat ini lebih belakangan dari tanggal selesai untuk Grade. Jika tanggal saat ini setidaknya setahun lebih belakangan dari tanggal selesai maka obyek dihapus dari sistem. Alternatif lainnya kelas GradeRate mungkin memiliki atribut tunggal (jenis enumerated – yang memiliki nilai integer untuk setiap state yang mungkin) dengan nilai yang menunjukkan *current state* dari suatu obyek. Secara konseptual, sebuah obyek tetap berada dalam satu state pada interval waktu tertentu.

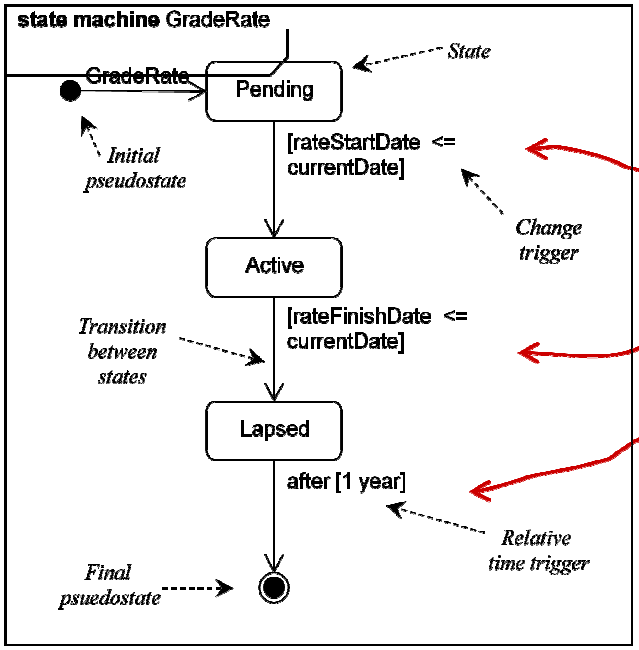
Suatu atribut yang menyimpan nilai current state dari suatu obyek disebut sebagai *state variable*. Penting untuk diperhatikan bahwa perpindahan dari satu state ke state lainnya untuk obyek GradeRate bergantung pada event yang terjadi dengan berlalunya waktu. Perpindahan dari satu state ke state lainnya disebut *transition*, ini diawali dengan sebuah *trigger*. *Trigger* adalah suatu event yang menyebabkan state berubah dan relevan dengan obyek (atau dengan elemen yang dimodelkan). Ketika sebuah event yang ditrigger mengahsilkan sebuah transition ini disebut *fire*. Sebuah transition ditunjukkan dengan panah terbuka dari state sumber ke state target. Contohnya penundaan suatu advert pada Agate adalah sebuah trigger yang akan mengubah state dari obyej Advert yang sedang ditunda.

### 9.2.1. State Machine

Current state dari obyek GradeRate dapat ditentukan oleh dua atribut rateStartDate dan rateFinishDate. Sebuah variabel state *enumerated* dapat digunakan untuk menyimpan state dari suatu obyek, nilai-nilai yang mungkin adalah Pending, Active atauLapsed.

# state machine

state machine for the class `GradeRate`.




Movement from one state to another is dependent upon events that occur with the passage of time.




## 6.3 Use Case Realization

### 6.3.1 Transformasi Use Case ke Communication dan Class Diagram

Dari model *Use Case* maka tahapan berikutnya adalah melanjutkan ke arah implementasi *software*. Proses pengembangan suatu model abstrak atau elemen yang lebih dekat ke arah implementasi di sebut dengan ***realization***.

***Use Case realization*** dilakukan dengan mencari sekumpulan *class* yang mungkin serta bagaimana interaksi *class* tersebut untuk mendapatkan fungsionalitas dari *use case*. Kumpulan dari *class* itu di sebut dengan ***collaboration***. *Collaboration* adalah realisasi *dari use case* tertentu.

Gambar 4. 1 transformasi Use case ke Diagram Class

Diagram komunikasi sangat berguna untuk melihat secara lebih rinci. Diagram ini juga memperlihatkan interaksi antara objek. Hal terpenting pada diagram komunikasi adalah untuk menunjukkan bagaimana objek berkollaborasi antara satu dengan yang lain. Hasil dari kolaborasi tersebut dapat di representasikan dalam bentuk *Class diagram*. *Class diagram* tidak menunjukkan interaksi antara objek namun lebih memperlihatkan struktur dari objek serta memperlihatkan fasilitas lain yang dimiliki oleh *class*.

### 6.3.2 Analisis stereotype dari class

Elemen model yang terdapat dalam model analisis disebut kelas analisis (*analysis class*)

*Boundary*, adalah kelas yang memodelkan interaksi antara satu atau lebih aktor dengan sistem

*Control*, digunakan untuk memodelkan "perilaku mengatur", khusus untuk satu atau beberapa use-case

*Entity*, memodelkan informasi yang harus disimpan oleh sistem

#### Boundary

Kelas boundary memodelkan bagian dari sistem yang bergantung pada pihak lain disekitarnya dan merupakan pembatas sistem dengan lingkungan luarnya.

Dapat Berupa :

***User Interface*** : sarana komunikasi sistem dengan user (contoh : *window* dalam GUI)

***System Interface*** : sarana komunikasi sistem dengan sistem lainnya (*communication protocol*)

***Device Interface*** : sarana komunikasi sistem dengan device/alat (misal : printer, sensor, dsb)

Gambar 4. 2 Simbol Class Boundary

#### Control

Kelas ***control*** menggunakan atau membuat isi dari kelas entity. Kelas control memasangkan kelas boundary dengan kelas entity. ***Control object*** (*instance* dari kelas *Control*) mengkoordinasi perilaku sistem dan menggambarkan dinamika sistem dan mengontrol alur kerja suatu sistem

Gambar 4. 3 Simbol Class Control

### Entity

Kelas entity memperlihatkan struktur data dari sebuah sistem. Digunakan untuk memahami apa yang kira-kira akan ditawarkan oleh sistem kepada *user*. *Entity object* (*instance* dari kelas *entity*) biasanya pasif dan tetap/tidak berubah-ubah. Tanggungjawabnya adalah menyimpan dan mengatur informasi dalam sistem.

Gambar 4. 4 Simbol Class Entity

## 6.4 Class Diagram

### 6.4.1 Class dan Instance

*Class diagram* berisi kumpulan kelas yang berassosiasi antara satu dengan yang lain. Sebuah *Class diagram* digambarkan dengan kotak yang terdiri atas tiga bagian. Masing-masing bagian berisi tentang nama kelas, atribut serta operation class tersebut. Dibawah ini adalah notasi *class*.

Gambar 4. 5 Notasi Class

*Class Instance* adalah representasi dari *class*. Pada *class instance* menunjukkan masing-masing objek dari setiap *class. Class instance* tidak memiliki operation.

Gambar 4. 6 Notasi Instance

### 6.4.2 Atribut

Atribut merupakan bagian paling penting dari pendefinisian *class*. Atribut merupakan struktur umum yang dimiliki oleh kelas. Setiap objek memiliki nilai atribut.

### 6.4.3 Link

Link merupakan koneksi logic antara dua buah objek

Gambar 4. 7 Link

### 6.4.4 Assosiasi

***Asosiasi*** menunjukkan Kemungkinan hubungan logis atau hubungan antara objek dari satu kelas dengan objek lain. Jika dua Objek bisa dihubungkan maka ada assosiasi antar kelas.

Gambar 4. 8 Assosiasi

### 6.4.5 Multiplicity

Merupakan *range* kardinalitas dari suatu assosiasi. Sebagai contoh: Setiap nasabah bank mungkin memiliki satu atau lebih rekening. Setiap *account* adalah untuk satu, dan hanya satu, pelanggan

Gambar 4. 9 Multiplicity

### 6.4.6 Operation

***Operation*** merupakan bagian yang penting dari class. Operation biasanya merupakan *behavior* (perilaku) dari objek sebuah *class*. *Operation* menggambarkan apa yang bisa dilakukan oleh *instance* dari sebuah class.

Gambar 4. 10 Operation

## 6.5 Langkah-langkah menggabarkan diagram Class

Untuk menggambarkan diagram class dapat dilakukan hal sebagai berikut:

Mulai dari satu buah *Use Case*

Identifikasi semua *Class* yang terlibat (Gunakan *use case collaboration*)

Gambar *collaboration diagram* yang memenuhi kebutuhan *use case*

Terjemahkan collaboration ke class diagram

Ulangi untu *use case* lainnya

## 6.6 LATIHAN

PT Procon Indah (PI), bergerak dalam bidang usaha property terdiri dari penyewaan ruang gedung kantor, apartemen, flat, dan rumah tinggal. PI mempunyai wilayah pemasaran di Jabodetabek dan beberapa kota propinsi di Indonesia. Berdiri sejak tahun 1987 yang lalu, kantor PI berpusat di Jakarta dengan kantor cabang di setiap wilayah dan propinsi.

Organisasi perusahaan.

PI dipimpin oleh seorang Direktur dibantu beberapa Wakil, diantaranya Wakil Direktur Pemasaran, Keuangan, Umum dan Administrasi. Tiap cabang dipimpin oleh seorang Kepala Cabang dibantu Supervisor dan Asisten Supervisor. Kegiatan utama di setiap cabang adalah melakukan pemasaran kepada para pelanggan untuk mencapai hasil kesepakatan sewa properti yang paling memuaskan.

Visi Pimpinan

Visi perusahaan adalah menyediakan beragam properti kepada pelanggan yang umumnya terdapat di cabang-cabang. Visi yang utama adalah kemampuan perusahaan menyediakan properti yang berkualitas untuk mencapai kepuasan pelanggan dan didukung oleh peranan para pemilik (owner). Perusahaan menyediakan jasa profesional untuk para owner sehingga mereka akan memperoleh benefit yang optimal dari propertinya

Manajer Cabang.

Manajer Cabang memberi layanan ke pelanggan dan owner dibantu oleh para staf. Sejumlah staf melakukan monitor proyek baru dan menghubungi para pelanggan. Tujuannya meyakinkan bahwa semua kebutuhan pelanggan dapat terpenuhi, juga melakukan negosiasi kontrak. Untuk

melakukan tugasnya manajer memerlukan data cabang, owner, pelanggan dan data kontrak. Selain itu jika terdapat beberapa property yang tidak laku-laku, Manajer cabang berhak memutuskan memasang advertise misalnya di koran, dsb. Secara periodik cabang membuat laporan tentang staf, property owner, kontrak sewa ke kantor Pusat.

Gambaran umum bisnis.

Properti yang disewakan, tidak selalu milik perusahaan. Tetapi dapat saja milik perorangan atau perusahaan yang disebut *Owner* (pemilik). Sudah tentu mereka ini menginginkan hasil penjualan sewa yang memuaskan. Setiap pemilik akan menyerahkan rincian data properti miliknya yang akan disewakan ke Cabang PI dan juga menyebutkan tarip-sewa yang diharapkannya untuk ditawarkan kepada para pelanggan.

Prosedur penyewaan property

Penyewa adalah perorangan atau perusahaan, akan dilayani oleh staf PI setelah mereka melakukan registrasi, dan selanjutnya disebut Pelanggan. Para pelanggan ini diantar oleh staf yang bertanggungjawab pada properti tertentu, akan melakukan peninjauan (viewing) untuk melihat kondisi properti yang akan disewa. Apabila masih belum cocok, pelanggan akan diantar ke properti lainnya sampai diperoleh unit yang sesuai dengan harapan mereka oleh staf itu. Pelanggan biasanya mempertimbangkan luas ruang, jumlah kamar, lokasi dan harga sewa ruang.

Staf PI akan menyiapkan bahan-bahan kontrak sewa dan diserahkan ke Manajer Cabang untuk diproses lebih lanjut Jika pelanggan telah memperoleh unit yang sesuai dengan kebutuhan mereka, , jika diperlukan negosiasi pelanggan akan dilayani langsung oleh Manajer Cabang.

Kontrak sewa dibuat, karena pembayaran dapat dilakukan secara bertahap sesuai ketentuan perusahaan. Selain itu lamanya sewa disesuaikan dengan kebutuhan biasanya enam bulan sampai satu tahun. Sedangkan pembayarannya dapat dilakukan setiap bulan.

Berdasarkan gambaran kasus tersebut, Tentukan object, class dan gambarkanlah *Class Diagram* dengan menggunakan rational Rose

---------- SELAMAT MENGERJAKAN ----------