



Requirements Analysis

Mengapa Melakukan Analisa Requirement

- Model usecase saja tidak cukup
 - Banyak terjadi perulangan
 - Beberapa bagian bisa saja sudah tersedia sebagai komponen standar
- Analisa bertujuan untuk mengidentifikasi :
 - Common elements
 - Pre-existing elements
 - Interaksi antar requirements

Tujuan Requirement

Requirement diperlukan dengan tujuan:

- Konfirmasi apa yang user inginkan dari sistem
 - Harus di pahami user
 - Harus benar dan lengkap
- Lebih spesifik apa yang perancang akan rancang
 - Tidak boleh ambigu

Tujuan Requirement

- Menjelaskan apa yang harus dilakukan Menjelaskan apa yang harus dilakukan oleh *software* yang akan di bangun
- Mewakili orang, benda atau konsep untuk memahami apa yang sedang terjadi
- Menunjukkan hubungan serta interaksi antara orang, benda atau konsep tersebut
- Menunjukkan situasi bisnis secara jelas untuk mengevaluasi kemungkinan rancangan yang akan di buat
- *Requirement* disusun kembali sehingga berguna untuk pengembangan *software* berikutnya

Bagaimana Menganalisa Model

- Alat utama yang digunakan untuk analisis Requirement adalah diagram kelas
- Dua cara utama untuk menghasilkan:
 - Langsung diperoleh berdasarkan pengetahuan pada *application domain*
 - Membuat *class diagram* dari setiap *use case* secara terpisah kemudian setiap *class diagram* yang diperoleh disusun menjadi satu class diagram.

Diagram Kelas: Stereotypes

- 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

Diagram Kelas : Stereotypes

Alternative notations for boundary class:

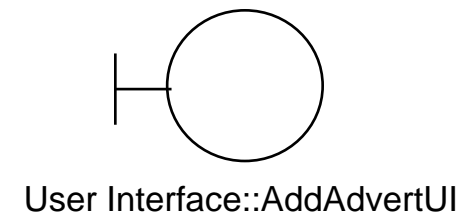
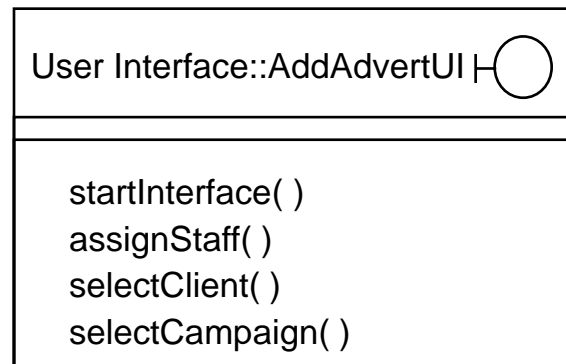
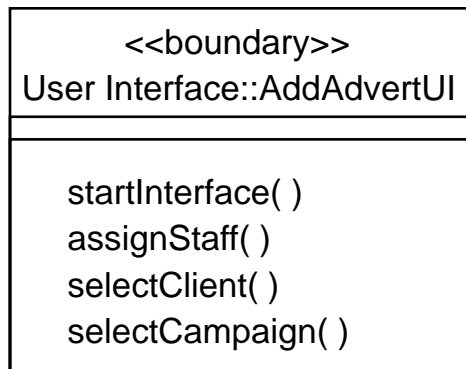


Diagram Kelas : Stereotypes

Alternative notations for entity class:

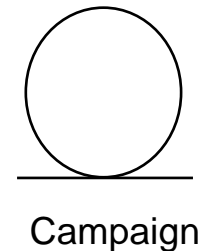
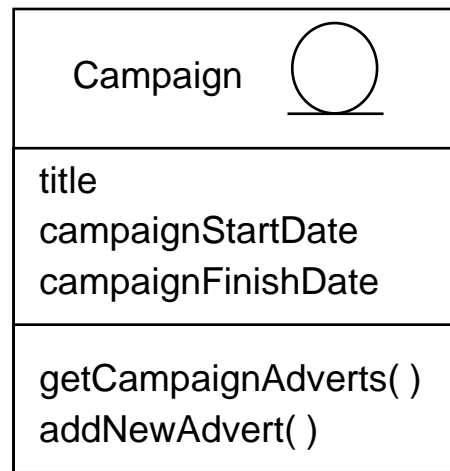
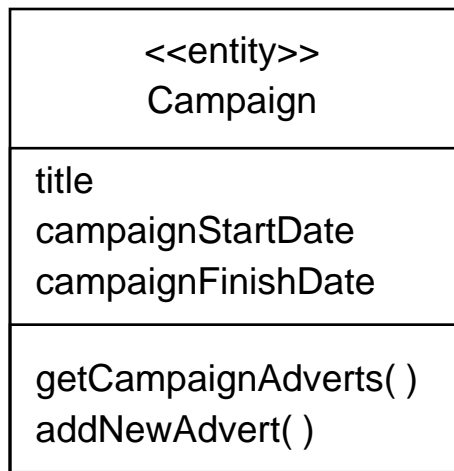


Diagram Kelas : Stereotypes

Alternative notations for control class:

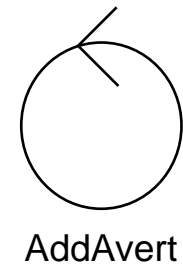
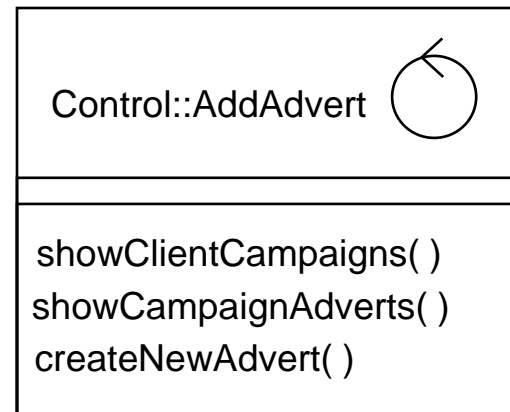
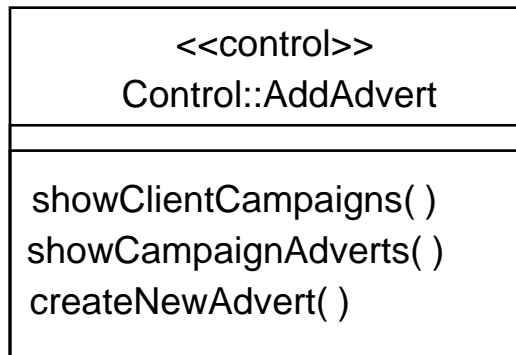


Diagram Kelas : Simbol Kelas

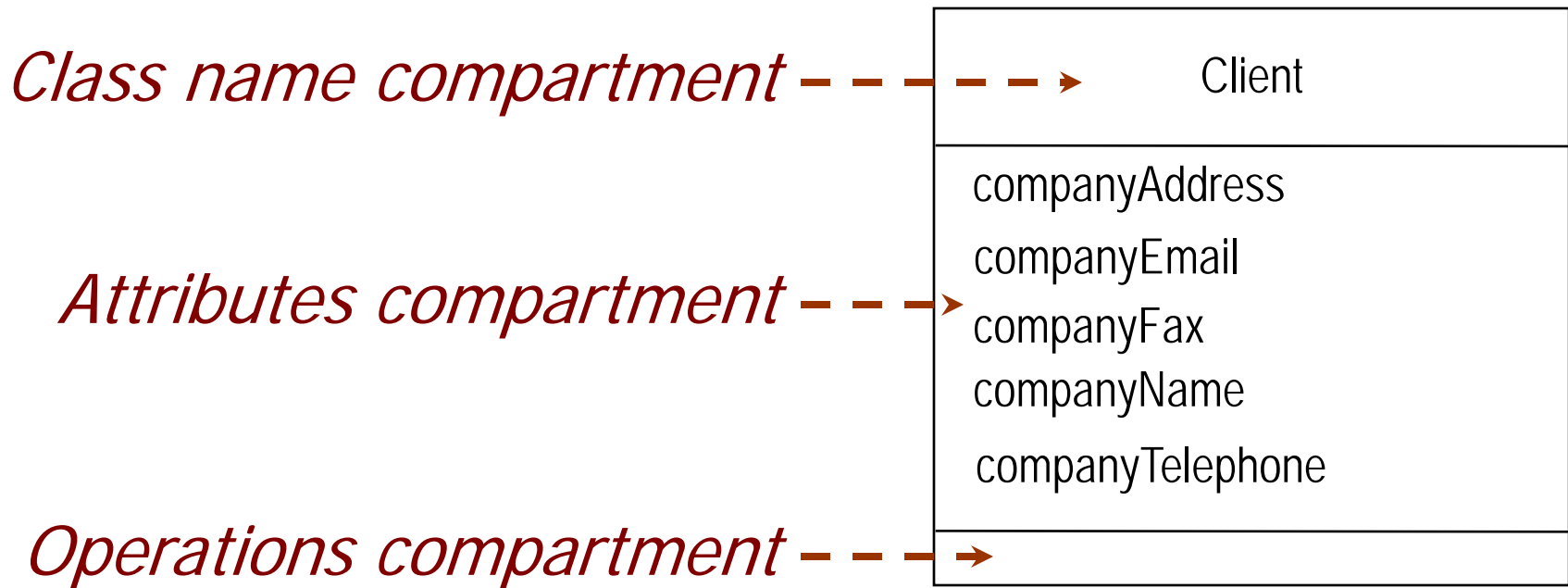


Diagram Kelas : Simbol Instan

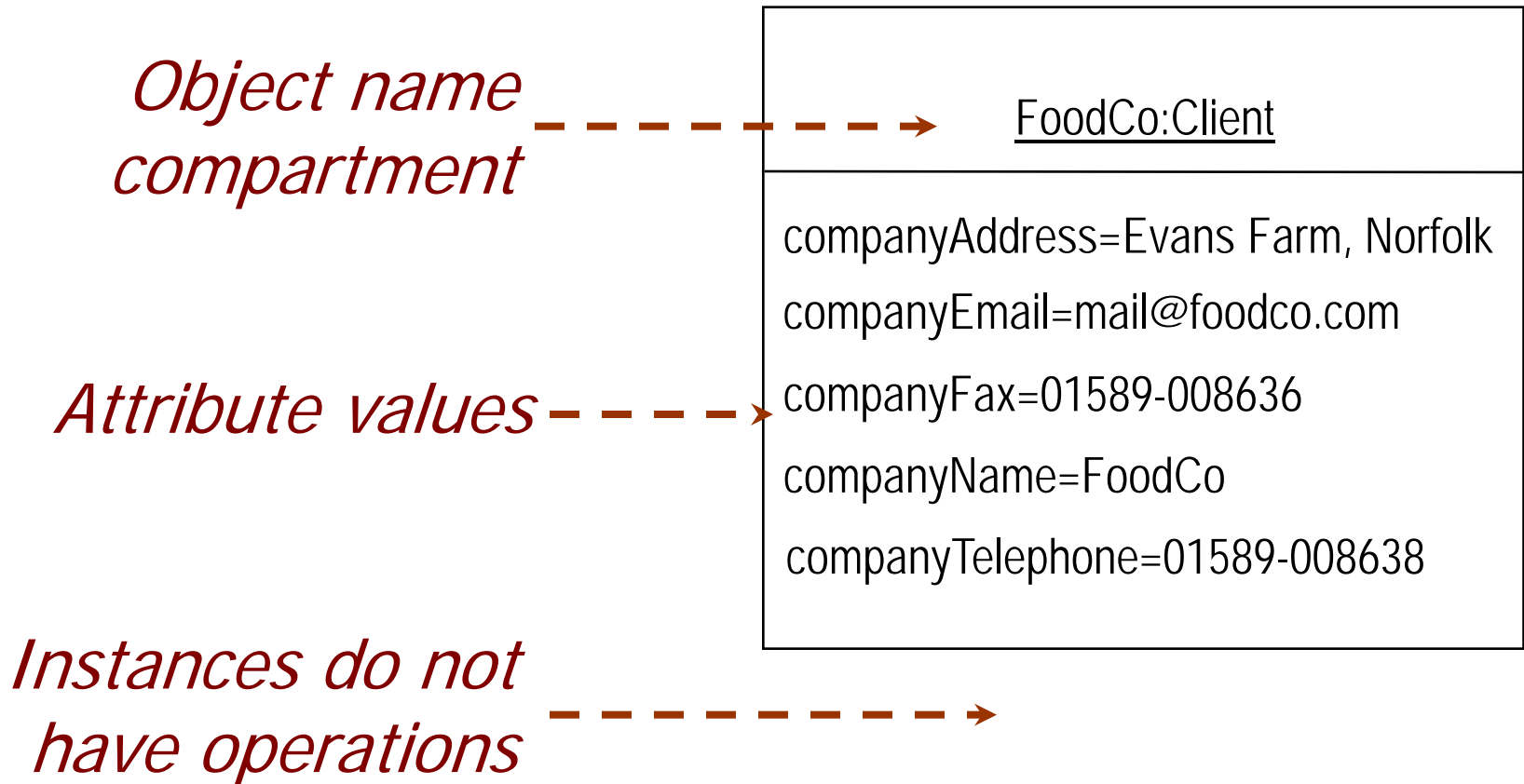


Diagram Kelas : Atribut

Attributes adalah:

- Bagian yang paling penting untuk menggambarkan kelas
- Struktur umum dari class
- setiap objek memiliki nilai atribut dari class

Diagram Kelas : Link

A link is a logical connection between two objects

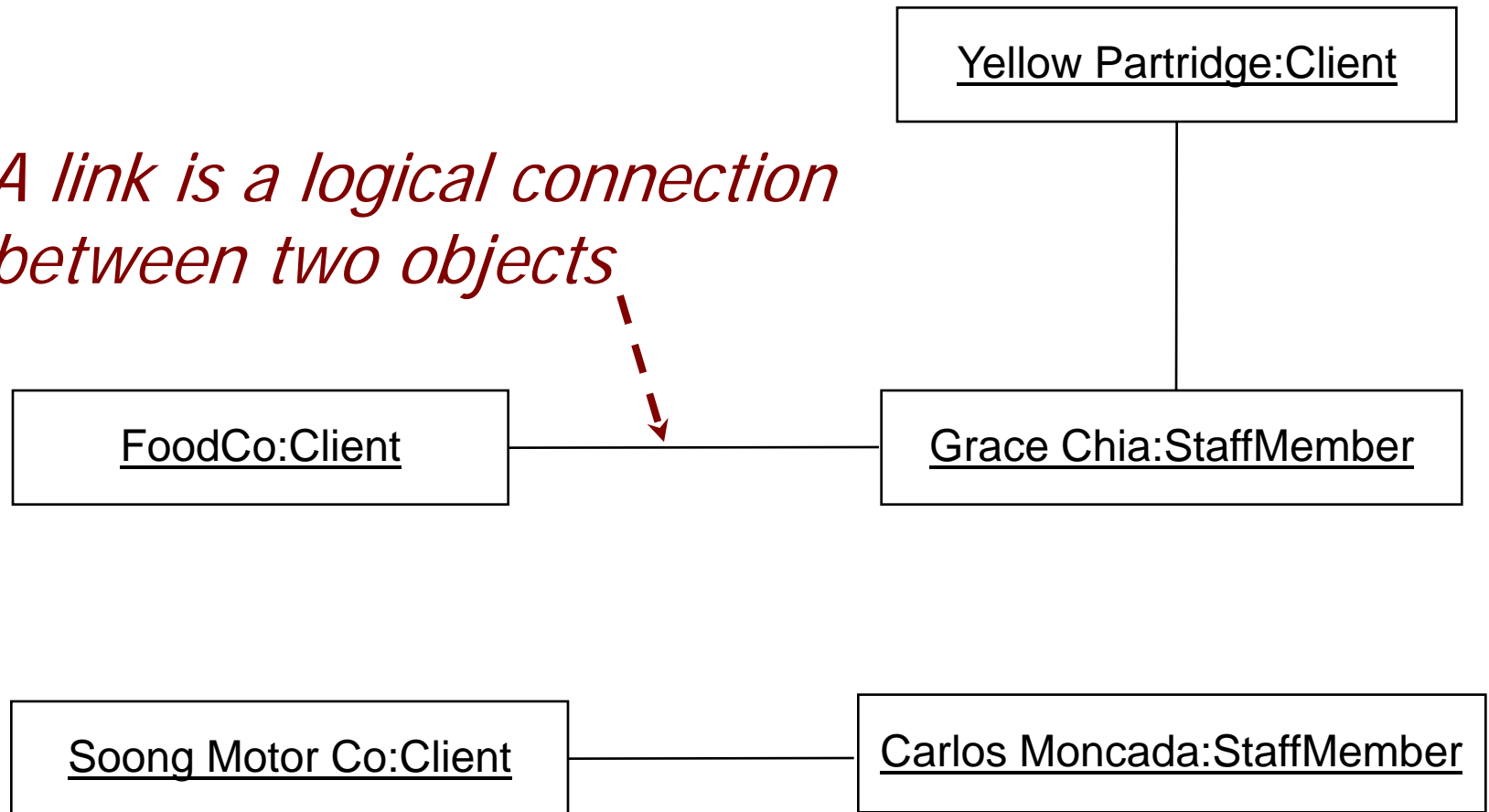


Diagram Kelas : Asosiasi

- ***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.

Diagram Kelas : Asosiasi

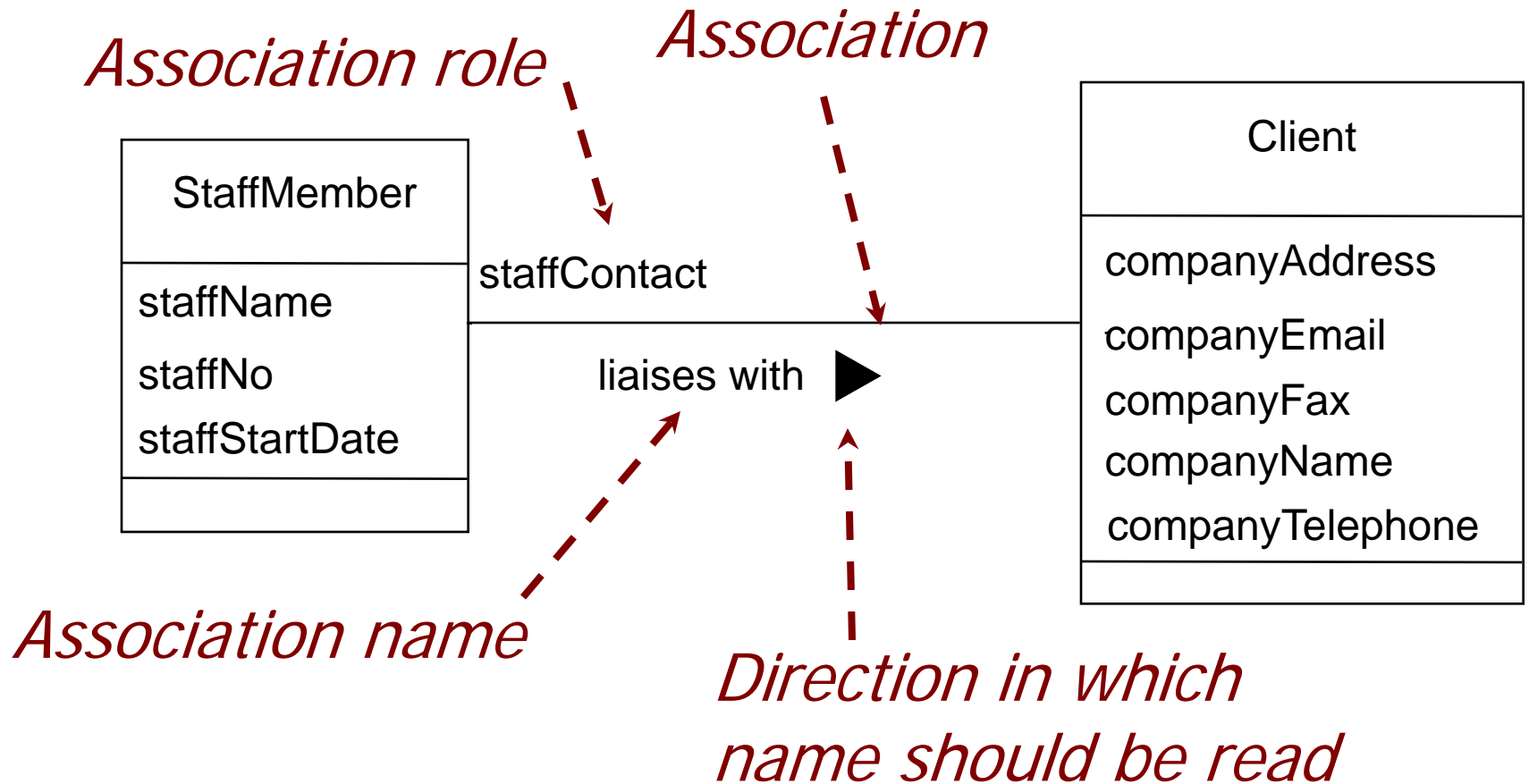
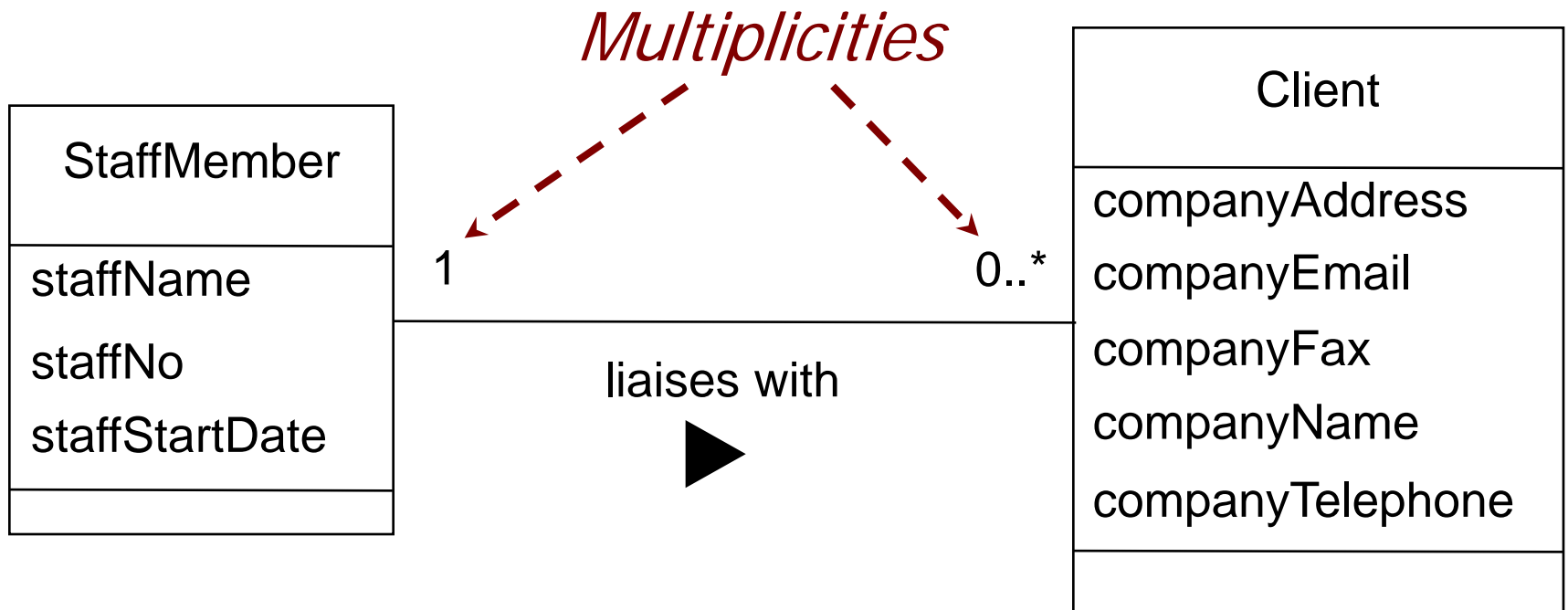


Diagram Kelas : Multiplicity

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

Diagram Kelas : Multiplicity



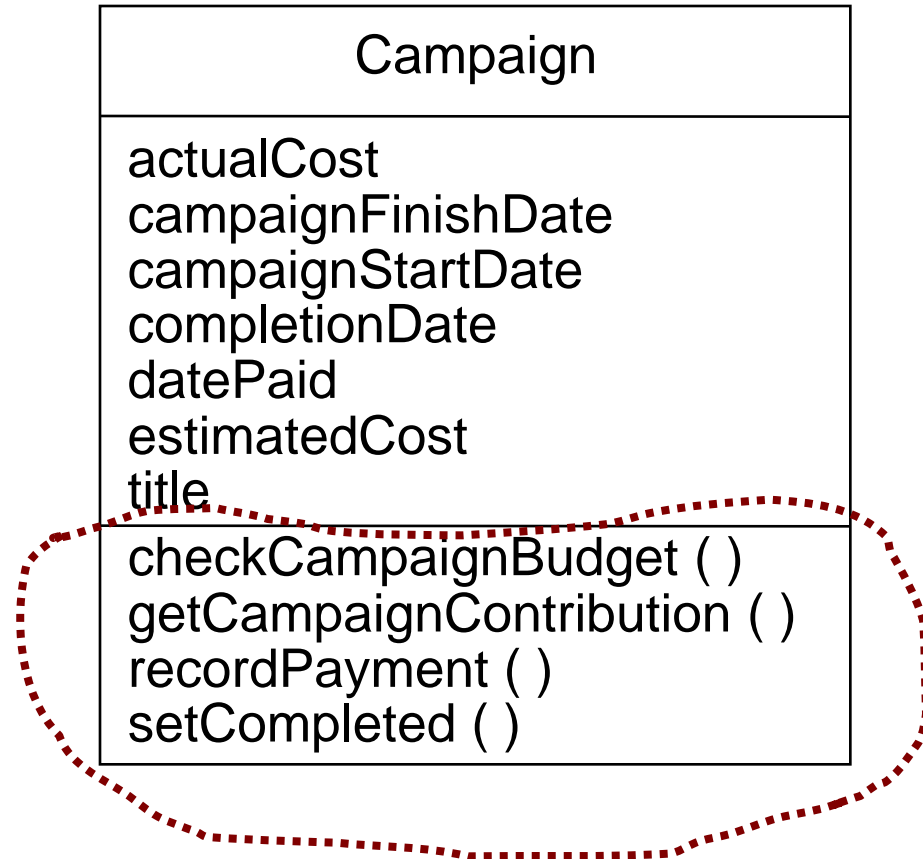
- Exactly one staff member liaises with each client
- A staff member may liaise with zero, one or more clients

Diagram Kelas : Operasi

- ***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.

Diagram Kelas : Operasi

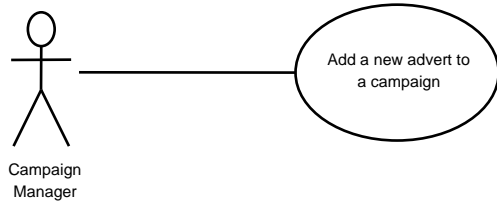
- Operations menggambarkan yang instance atau class kerjakan



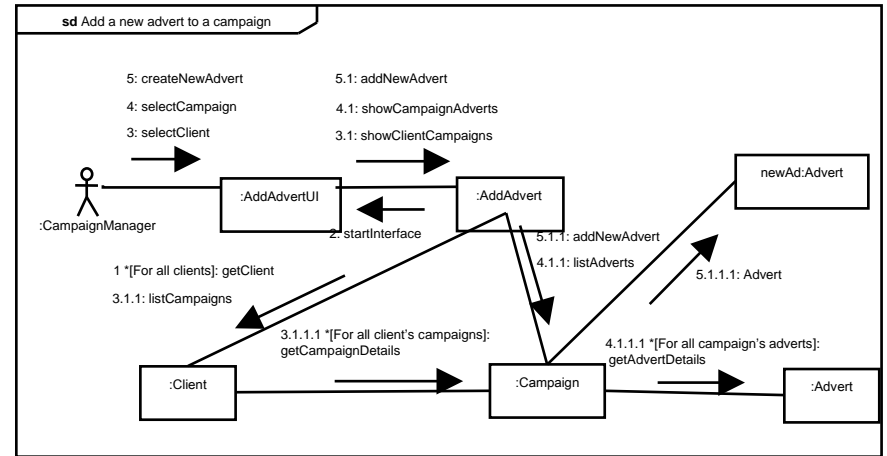
Dari Requirements ke Kelas

- 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

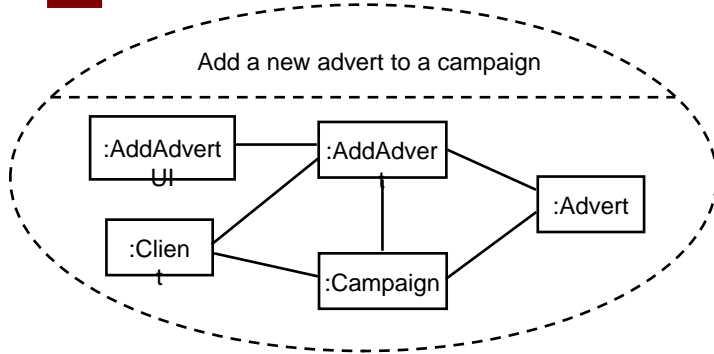
Dari Requirements ke Kelas



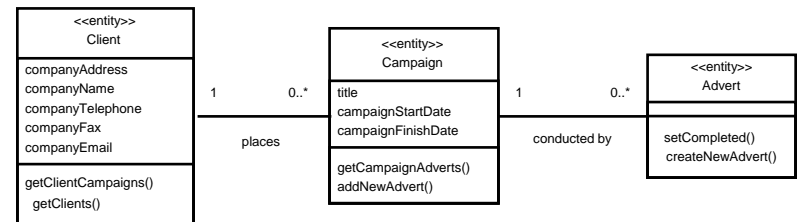
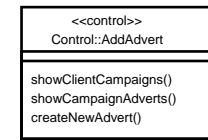
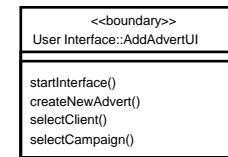
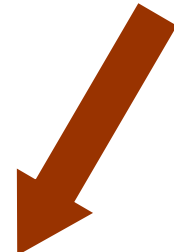
3



2



4



sd Add a new advert to a campaign

5: createNewAdvert

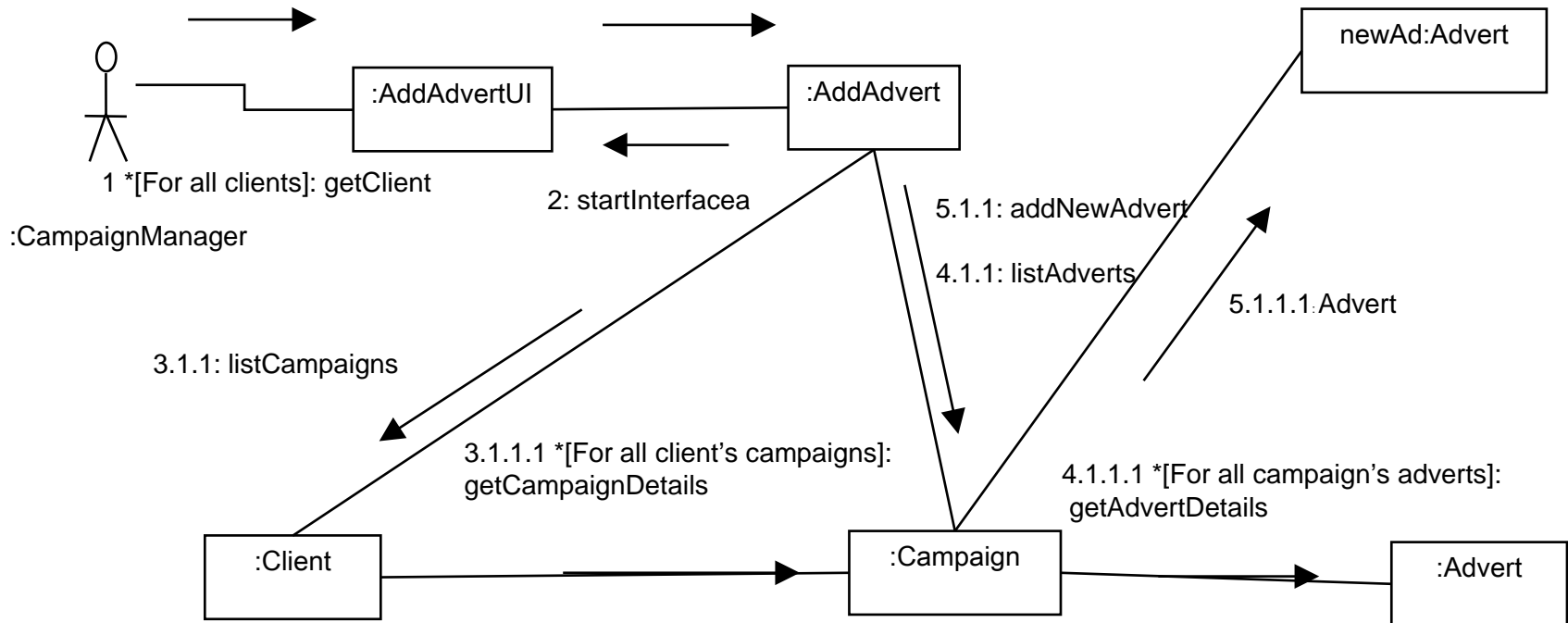
5.1: addNewAdvert

4: selectCampaign

4.1: showCampaignAdverts

3: selectClient

3.1: showClientCampaigns



Mengidentifikasi Kandidat Kelas

- Sejumlah tes membantu untuk memeriksa apakah calon kelas wajar
 - Apakah itu di luar lingkup sistem?
 - Apakah itu mengacu pada sistem secara keseluruhan?
 - Apakah itu duplikat kelas lain?
 - Apakah itu terlalu samar?

Mengidentifikasi Kandidat Kelas (lanjut)

- Apakah terlalu terikat dengan input dan output fisik?
- Apakah itu benar-benar atribut?
- Apakah itu benar-benar operasi?
- Apakah itu benar-benar asosiasi?
- Jika jawaban adalah 'Ya', pertimbangkan model kelas potensial dalam beberapa cara lain

CRC Cards

- Class–Responsibility–Collaboration cards membantu untuk model interaksi antara obyek
- Untuk skenario yang diberikan (atau use case):
 - Brainstorm obyek
 - Mengalokasikan anggota tim
 - Peran dalam bermain interaksi

CRC Cards

Class Name:	
Responsibilities	Collaborations
<i>Responsibilities of a class are listed in this section.</i>	<i>Collaborations with other classes are listed here, together with a brief description of the purpose of the collaboration.</i>

Class Name <i>Client</i>	
Responsibilities	Collaborations
<i>Provide client information.</i> <i>Provide list of campaigns.</i>	<i>Campaign provides campaign details.</i>

Class Name <i>Campaign</i>	
Responsibilities	Collaborations
<i>Provide campaign information.</i> <i>Provide list of adverts.</i> <i>Add a new advert.</i>	<i>Advert provides advert details.</i> <i>Advert constructs new object.</i>

Class Name <i>Advert</i>	
Responsibilities	Collaborations
<i>Provide advert details.</i> <i>Construct adverts.</i>	

Referensi

- Wirfs-Brock (1990) gives a good exposition of CRC cards
(For full bibliographic details, see Bennett, McRobb and Farmer)