Object Interaction
Oleh : Kundang K Juman

Object Interaction

Sequence Diagram

- Sequence Diagram digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/even untuk menghasilkan output tertentu
- Sequence Diagram diawali dari apa yang me-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan

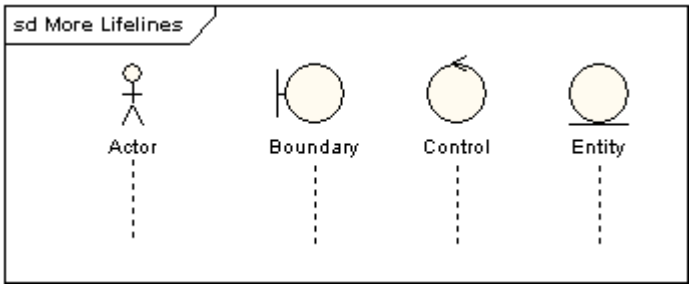
Komponen Sequence diagram :

•Actor

• Interface (Boundary)

• Proses pembacaan (Control)

•Nama table (Entity)

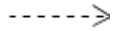
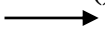


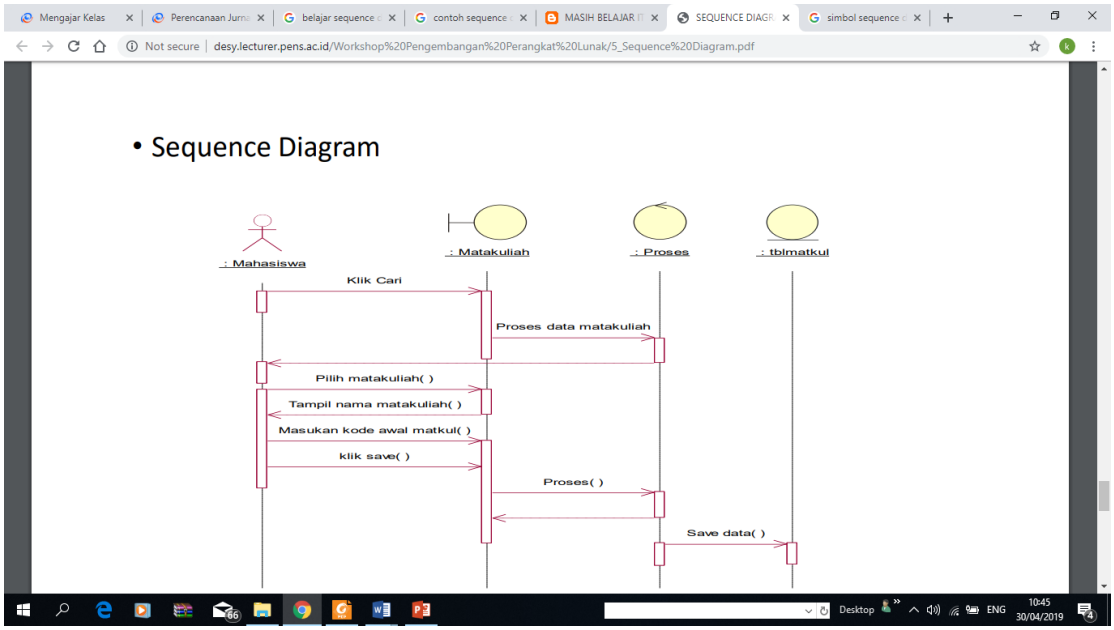
Symbol sequce diagram

Sequence Diagram

Sequence diagram merupakan gambaran interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display* dan sebagainya) berupa *message* yang digambarkan terhadap waktu (Verdi Yasin, 2012).

NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi itu sendiri.
2		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.
3		*Objek*	Menyatakan objek yang berinteraksi oleh pesan.
4		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

5		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
6	 I:keluaran	*Pesan tipe return*	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
7	 I:masukan	*Pesan tipe send*	Menyatakan bahwa suatu objek mengirim data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8	I:nama_ metode() 	*Pesan tipe call*	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.



Communication & Sequence

Diagrams

• An Interaction Diagram is a generalization of

two specialized UML diagram types

– Communication Diagrams: Illustrate object interactions

organized around the objects and their links to each

other

– Sequence Diagrams: Illustrate object interactions

arranged in time sequence

Communication & Sequence

Diagrams (2)

• Both diagram types are semantically equivalent, however,

they may not show the same information

– Communication Diagrams emphasize the structural

organization of objects, while Sequence Diagrams emphasize

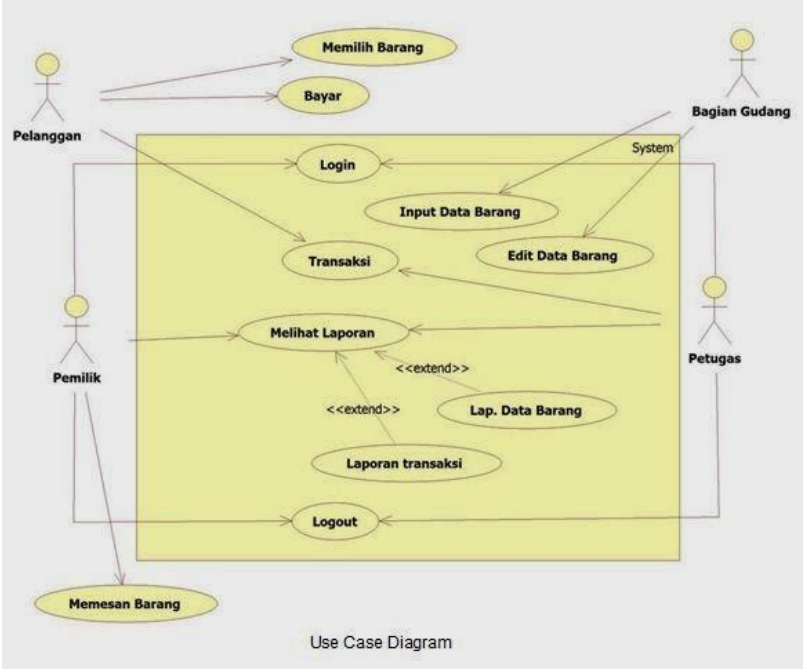
the time ordering of messages

– Communication Diagrams explicitly show object linkages,

while links are implied in Sequence Diagrams

- Sequence Diagram digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/even untuk menghasilkan output tertentu
- Sequence Diagram diawali dari apa yang me-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan
- Untuk lebih jelas berikut gambaran **Diagram Use Case** dari aplikasi penjualan barang dibawah ini :
-

Berikut adalah contoh Aplikasi Penjualan Barang :

- 

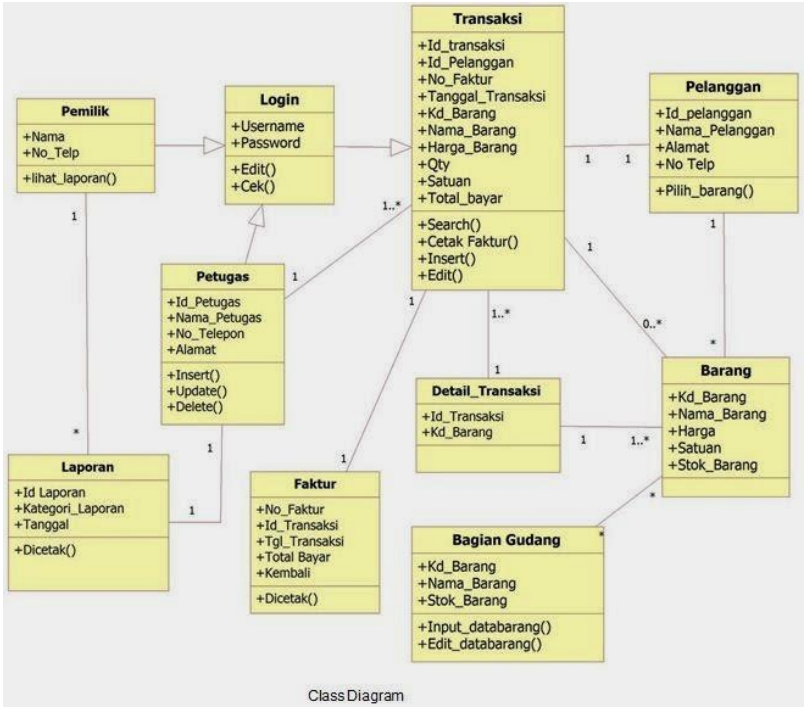
.

Dari gambar Use Case diatas usecase yang ada didalam sistem ditandai dengan

Boundary System artinya terlibat langsung dengan sistem.

Selanjutnya untuk mengklasifikasikan atau menggambarkan dari kelas-kelas

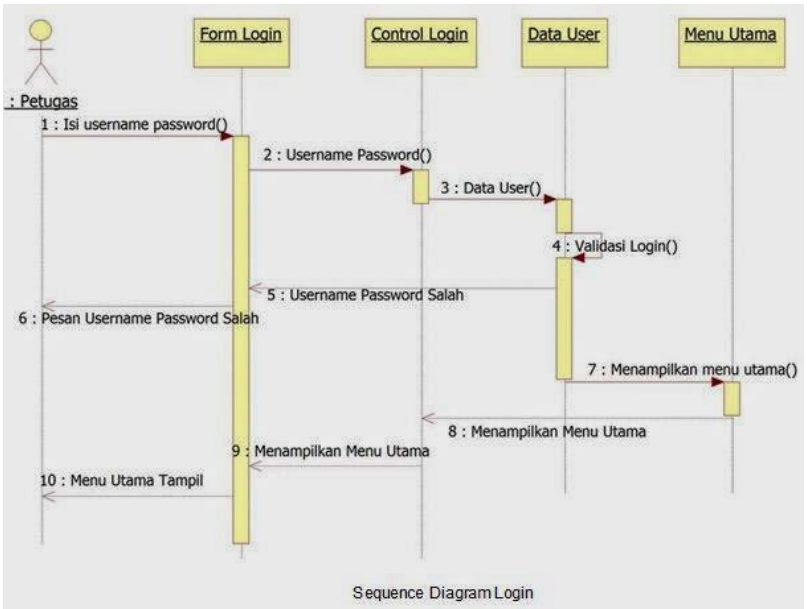
tersebut dapat dilihat pada **Diagram Class** berikut ini :

- 

Class Diagram

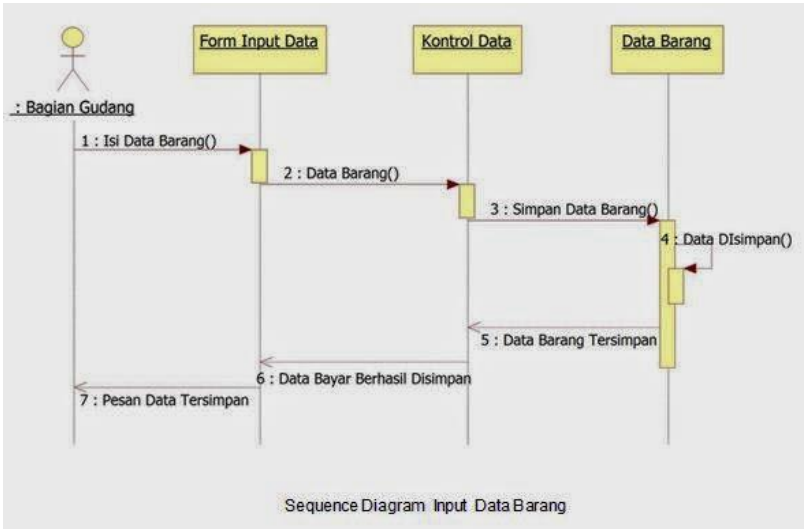
Kemudian untuk menggambarkan interaksi antar objek menunjukkan rangkaian pesan yang dikirim
antara object juga interaksi antara objek kita bisa melihatnya pada **Sequence Diagram** berikut ini :

1. Sequence Diagram Login

- 

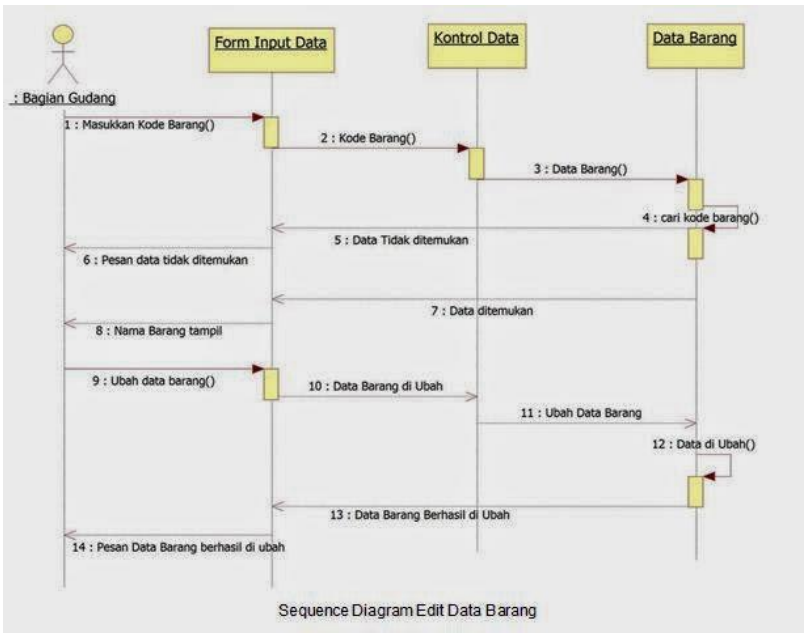
Sequence Diagram Login
-

- 2. Sequence Diagram Input Data Barang



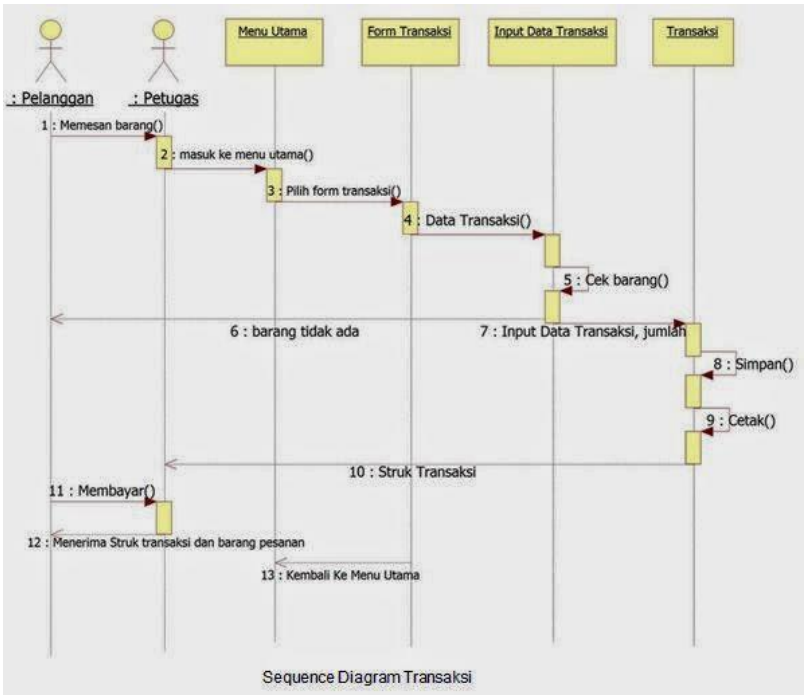
Sequence Diagram Input Data Barang

- 3. Sequence Dagram Edit Data Barang



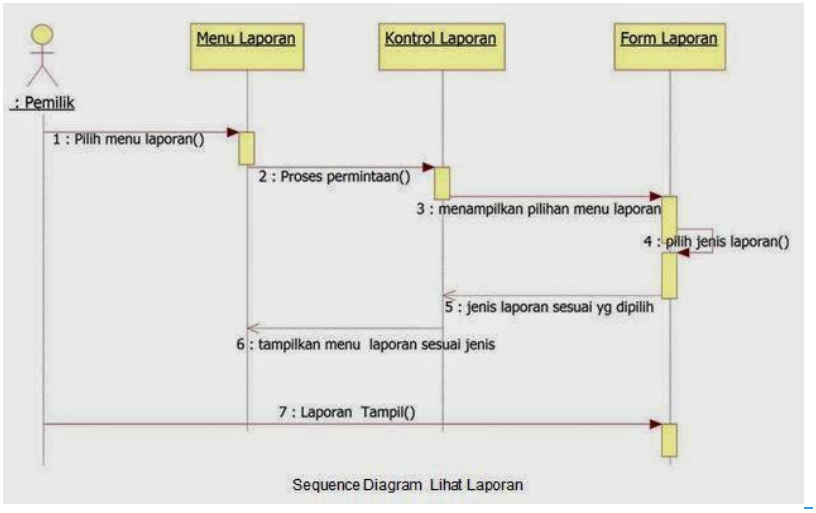
Sequence Diagram Edit Data Barang

- 4. Sequence Diagram Transaksi

- 

  Sequence Diagram Transaksi

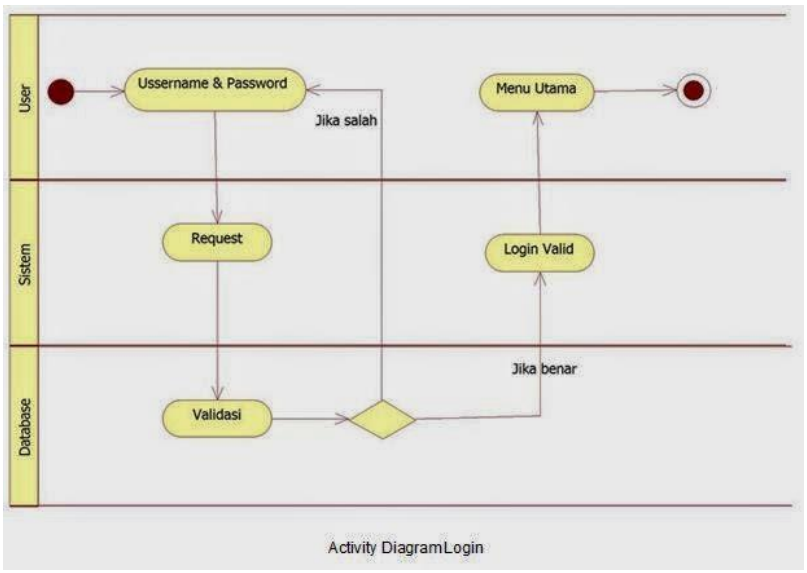
- 1. 5. Sequence Diagram Lihat Laporan

- 

  Sequence Diagram Lihat Laporan

- Setelah kita menggambarkan dengan sequence diagram selanjutnya kita buat Activity diagram (diagram aktivitas) dari sistem ini. Adapun Activity diagram itu adalah diagram yang menggambarkan aliran fungsionalitas dari sistem. Pada tahap pemodelan bisnis, diagram aktivitas dapat digunakan untuk menunjukkan aliran kerja bisnis (business work flow). Dapat juga digunakan untuk menggambarkan

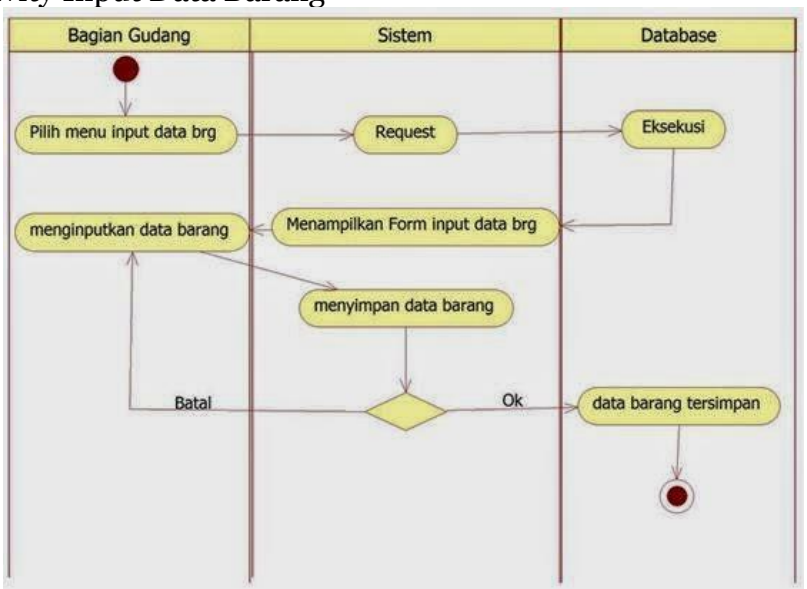
aliran kejadian (flow of events). Dan diagram aktivitas ini lebih memfokuskan diri pada eksekusi dan alur sistem dari pada bagaimana sistem itu dirakit.

-
- Adapun **Activity diagram** untuk penjualan barang adalah sebagai berikut:
-
- 1 Activity Login
-
- 

Activity DiagramLogin
-
-

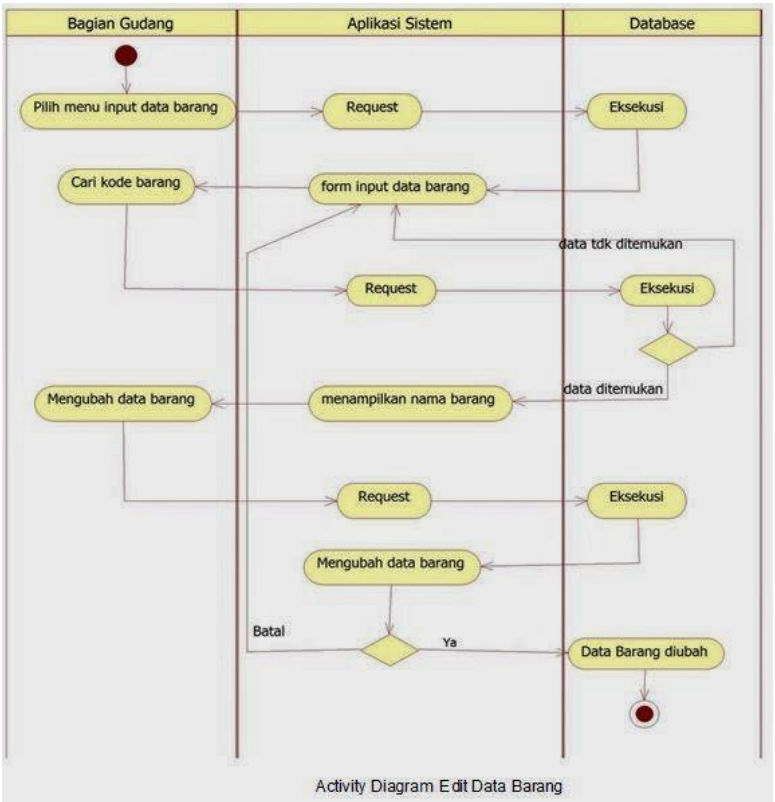
Dari aktivitas login diatas menggambarkan alur untuk sebuah user dapat login ke sistem. Dengan langkah yang pertama user memasukkan username dan password terlebih dahulu kemudian sistem memintanya dan username password akan di validasi apakah sudah sesuai atau belum. Selanjutnya activity diagram untuk proses input data barang adalah :

2. Activity Input Data Barang



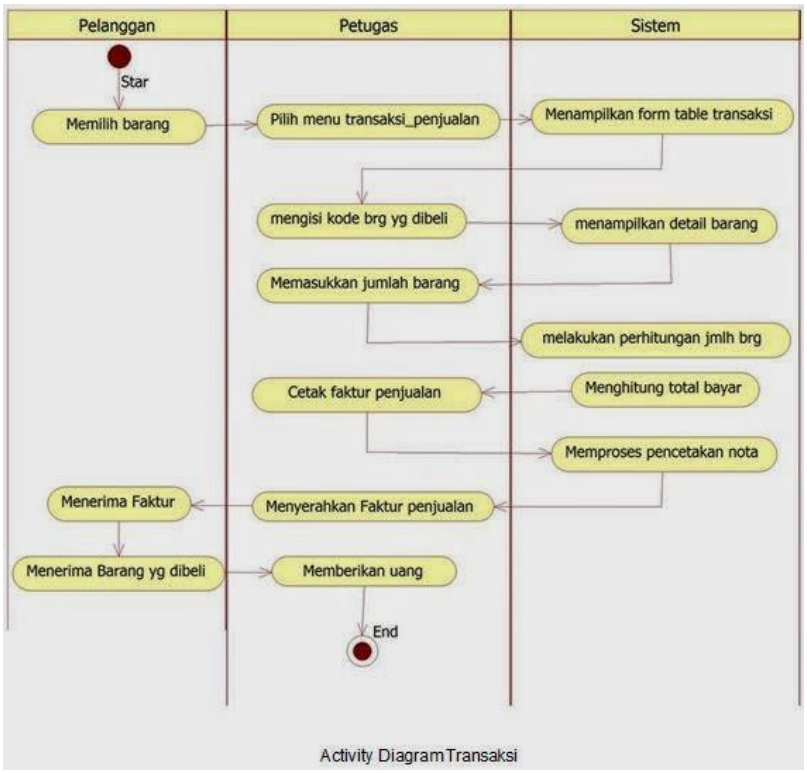
- Activity Diagram Input Data Barang

- 3. Activity Edit Data Barang



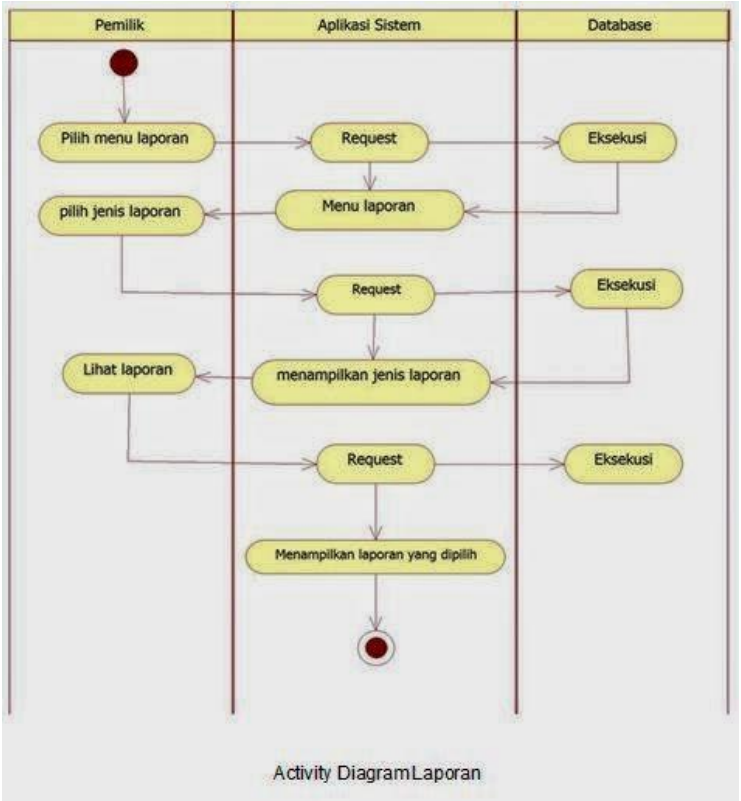
- Activity Diagram Edit Data Barang

- 4. Activity Diagram Transaksi



Activity DiagramTransaksi

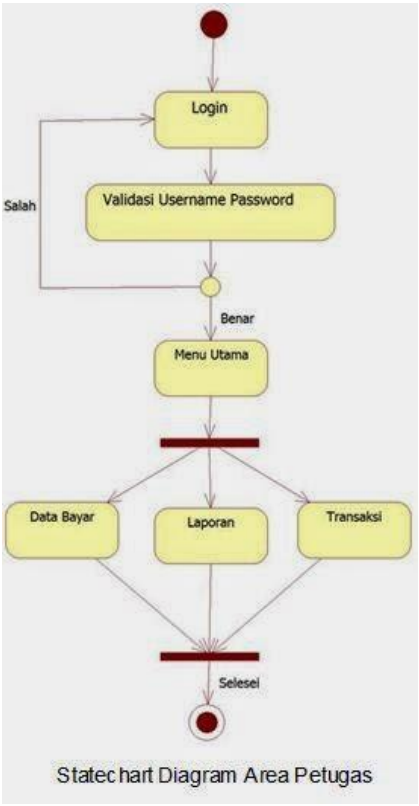
- 5. Activity Diagram Lihat Laporan



Activity DiagramLaporan

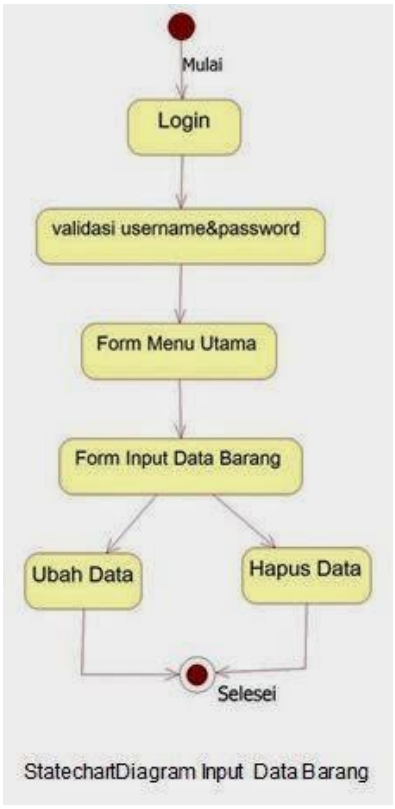
- Setelah menggambarkan dengan poses aktivity diagram digambarkan pula dengan **tatechart diagram**. Statechart diagram ini memperlihatkan diagram status yang memperlihatkan keadaan-keadaan pada sistem, memuat status (state), transisi, kejadian serta aktivitas. Dan diagram statechart juga memperlihatkan sifat dinamis dari interface, kelas, dan kolaborasi. Adapun Statechart Diagram untuk aplikasi ini yaitu sebagai berikut:

1. Statechart Area Petugas



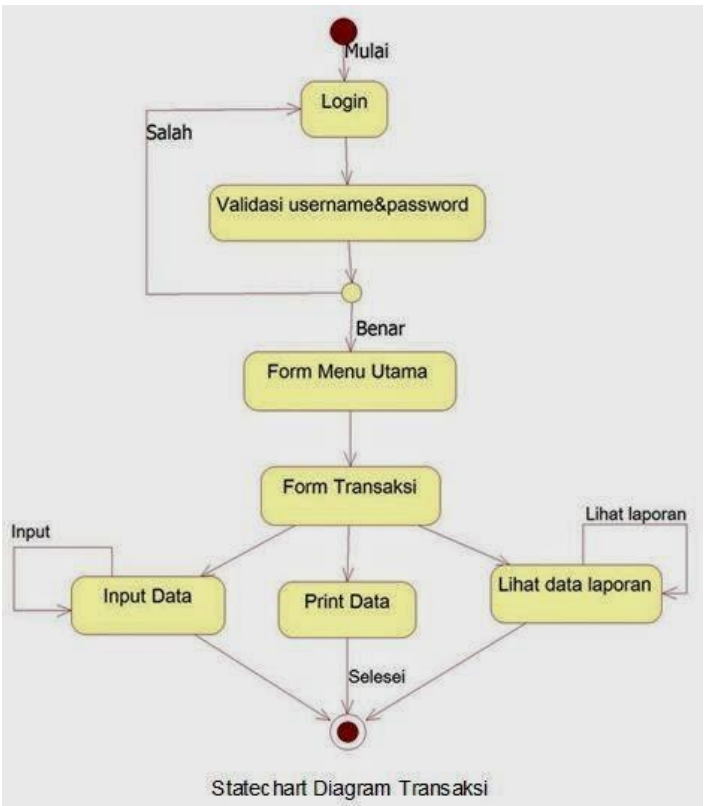
Statechart Diagram Area Petugas

2. Statechart Input Data Barang

- 

StatechartDiagram Input Data Barang

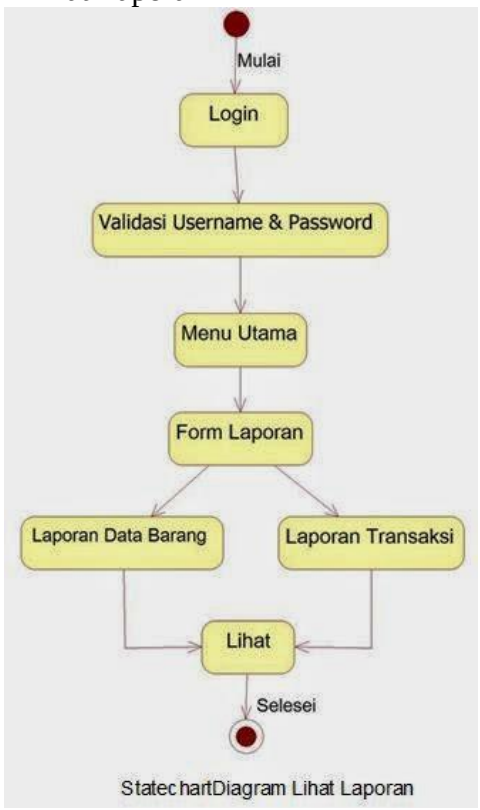
- 3. Statechart Diagram Transaksi

- 

Statechart Diagram Transaksi

-

- 4. Statechart Diagram Lihat Laporan



Statec hartDiagram Lihat Laporan

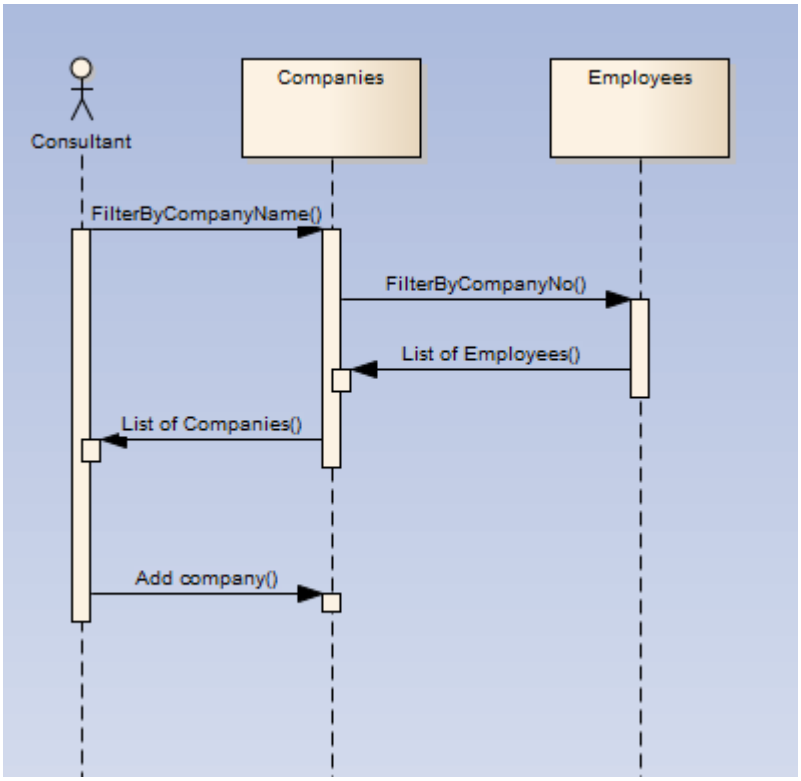
- Demikian rangkaian dari beberapa diagram mengenai aplikasi penjualan barang dari Toko DIKKA Sumedang, semoga bermanfaat dan menjadi nilai ibadah bagi yang membaca.

And draw between the actor and the object (not the life line!)

Extending the Sequence Diagram

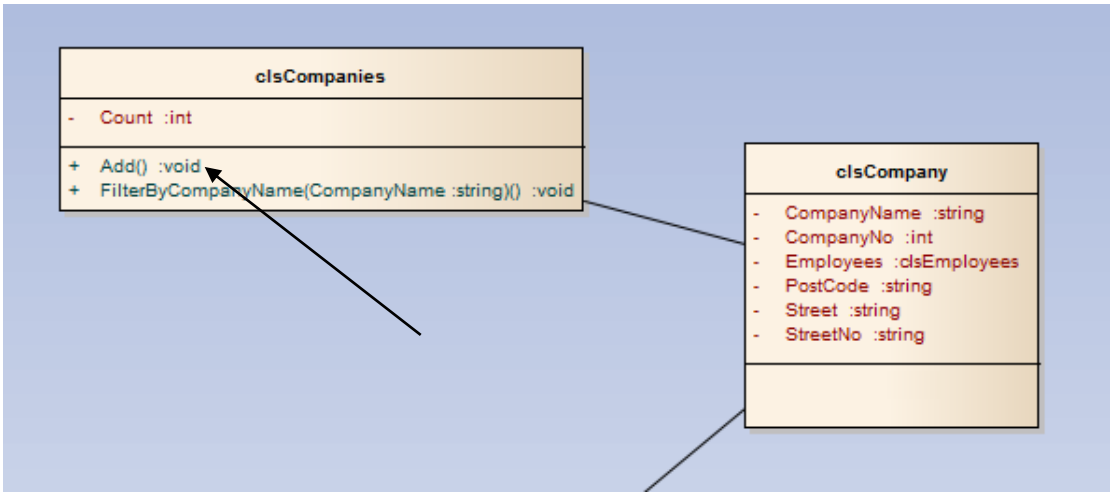
Once we have entered the company name and established that it doesn't exist on the system (let's ignore the possibility of employees switching companies for the moment!) we are in a position to add a new company / employee.

Let's add the Add Company message to the diagram…

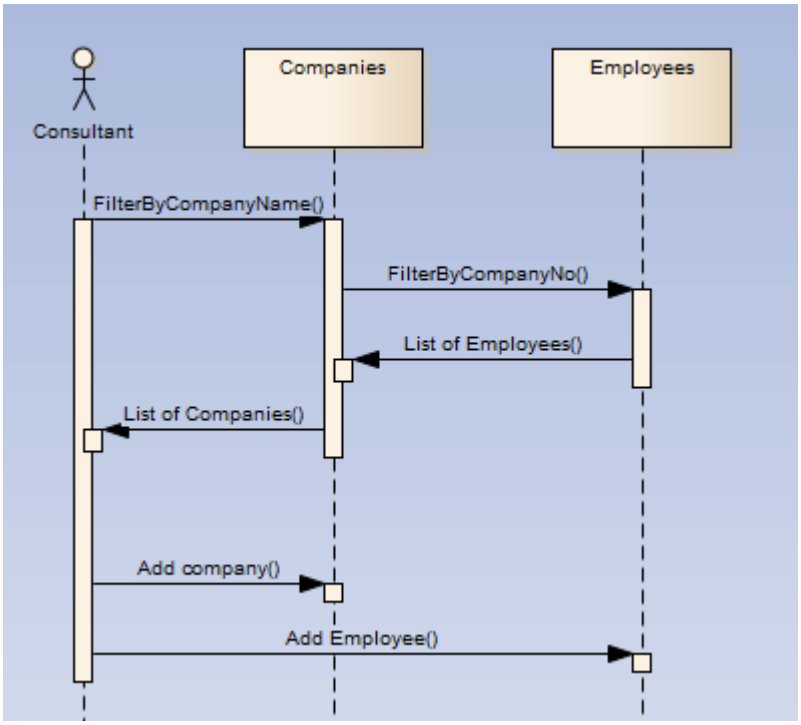


Do we have an operation to support this message? No!

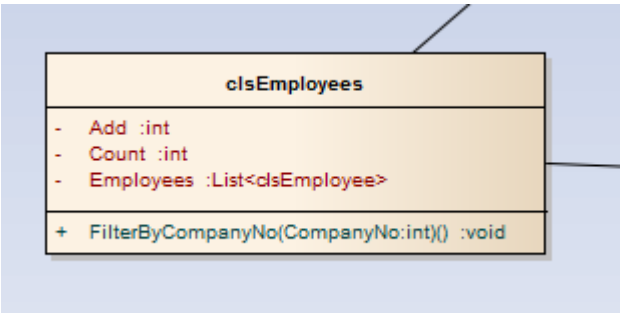
So let's modify the class diagram accordingly…



Now we need to add the employee data…



Where we find the same flaw in the class diagram as before…



Note that we are adding the Add functionality to the collection classes.

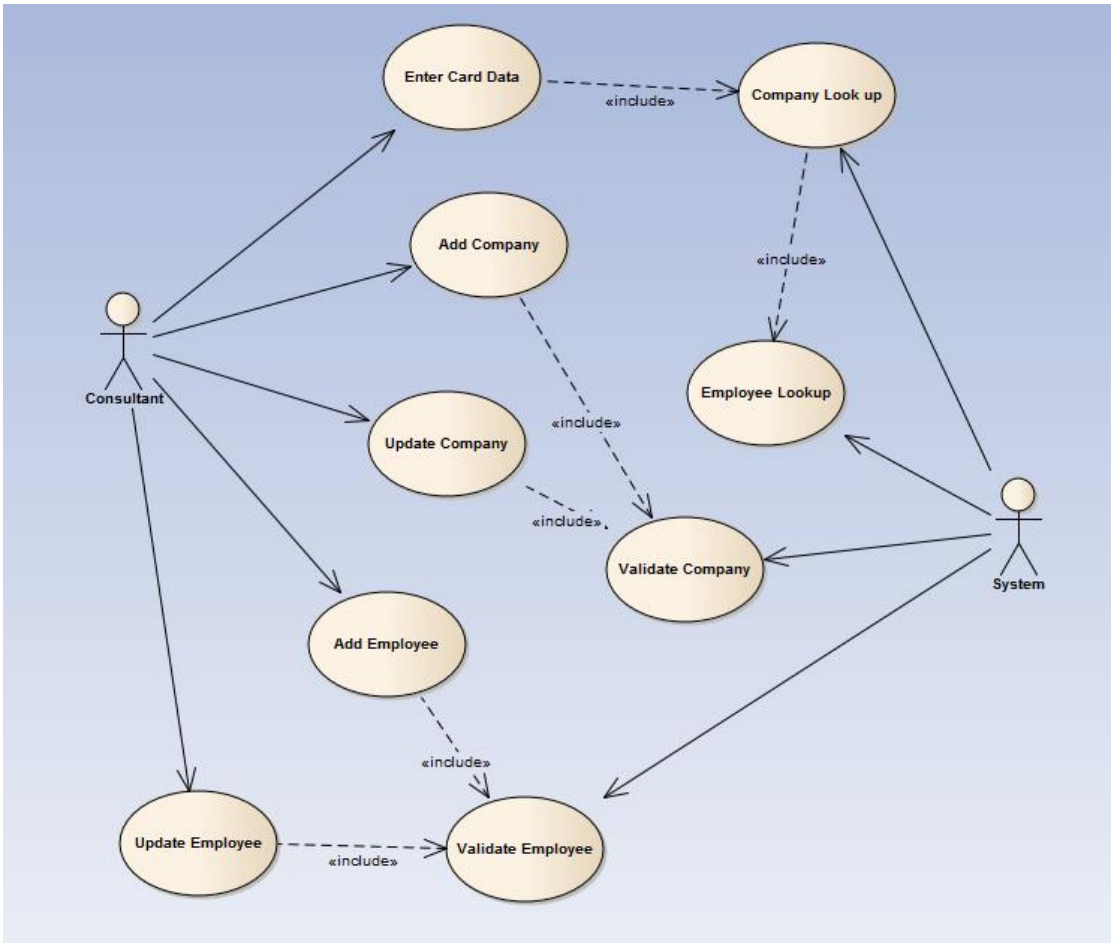
We now hit another problem.

Is it a good idea to be adding data without any suitable validation?

In creating the sequence diagram we have now spotted problems with both the uses case and the class diagram.

We need to add some sort of validation use case and we need to decide where to place the validation.

We will modify the use case as follows…



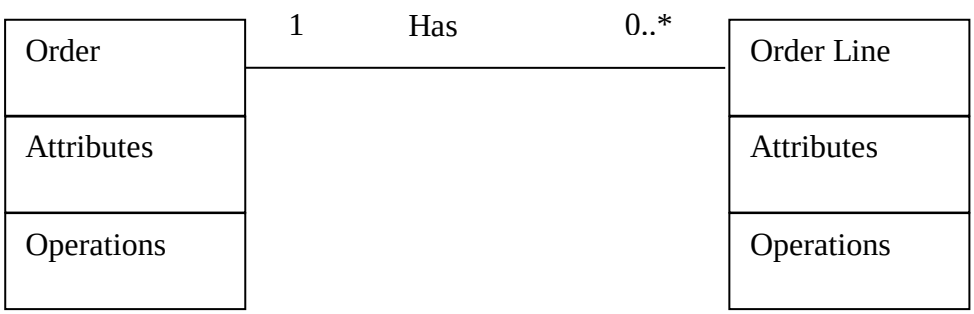
We need to decide where the validation needs to go.

Composition and Aggregation Revisited

Composition

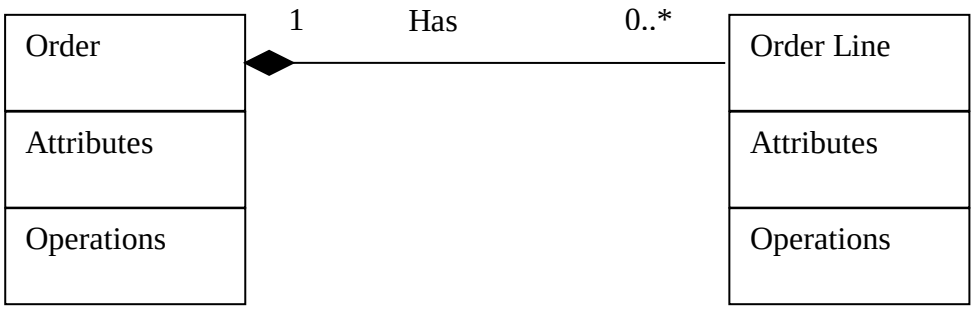
Two classes are related via composition when you cannot have an instance of one class without the parent class.

For example if we were placing an order in a system could we ever have an order line without an associated order?



The answer is “no”.

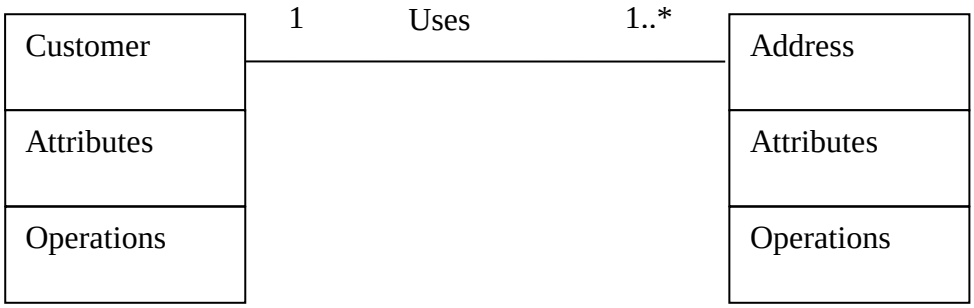
To indicate this strong relationship between the two classes we use a solid diamond like so…



Aggregation

In this case we are asking the question “can this class exist without the associated class even though they have a relationship?”

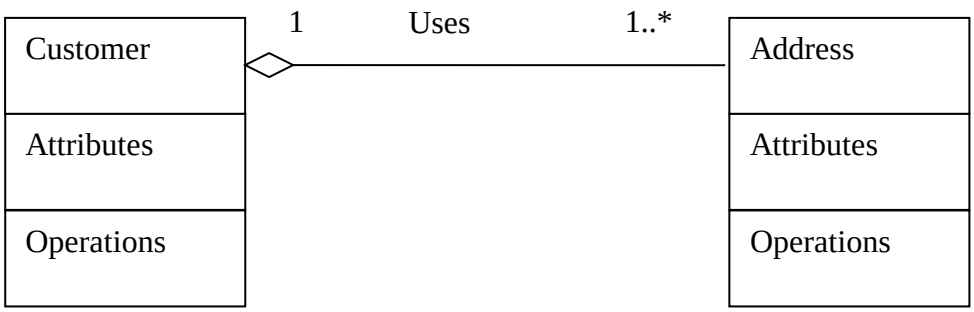
A good example of this is customer and address…



If all of the customers were to vanish from our store would addresses cease to exist?

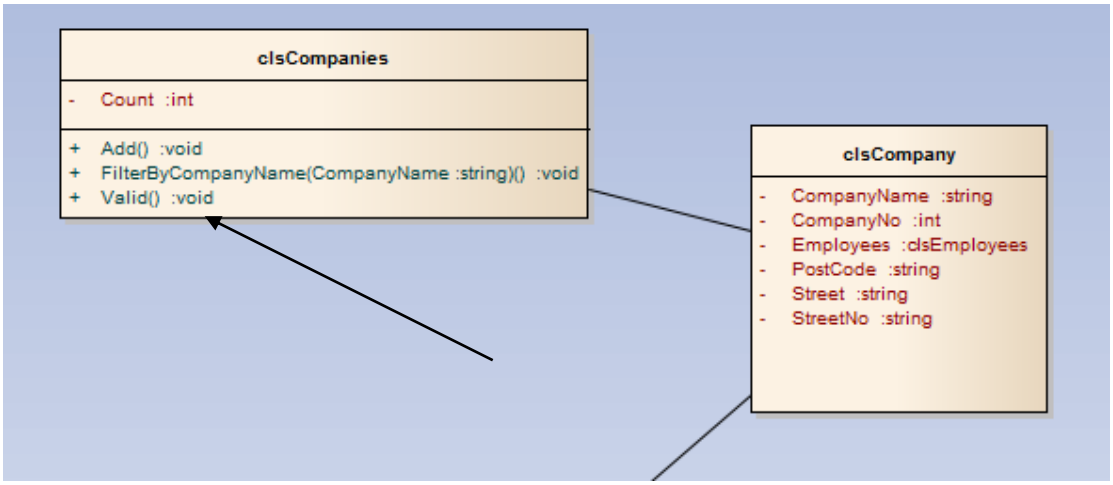
Obviously the answer is they would not!

To indicate this looser relationship a white triangle is used like so…



Composition and Aggregation give us important clues as to which class gets ownership of which attributes / operations.

We could design the classes like so…

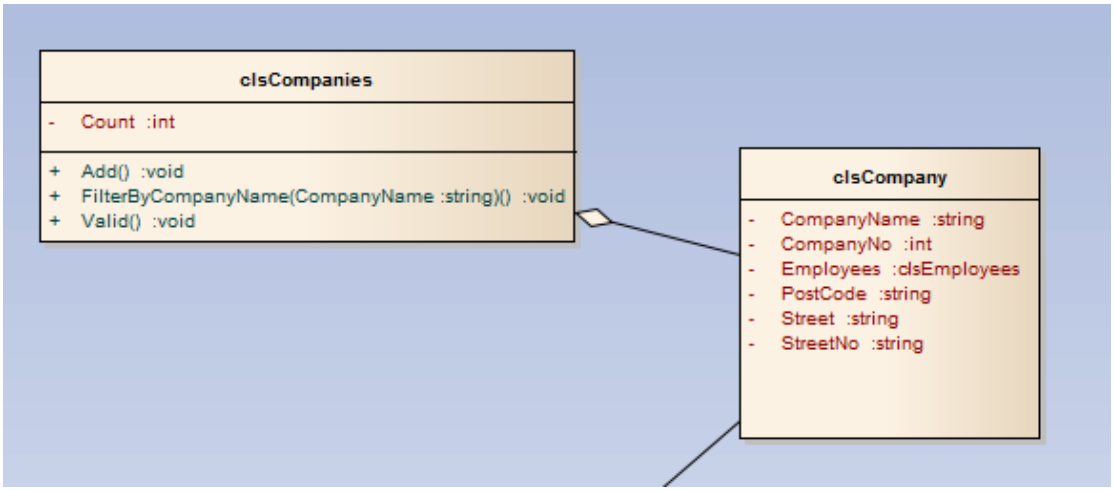


We need to ask, are we dealing with an aggregation or a composition?

Can we have an instance of a company on its own without the collection class?

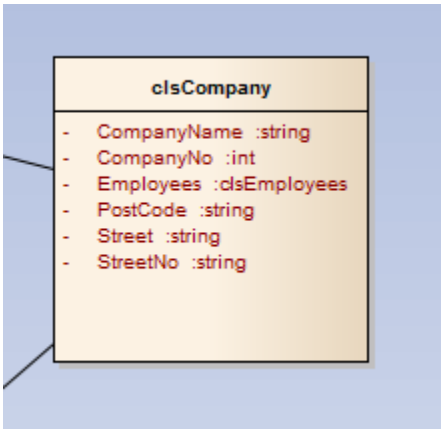
The answer is “yes” we may at times need to deal with a single company on its own.

This means that the diagram needs updating like so to indicate an aggregation…



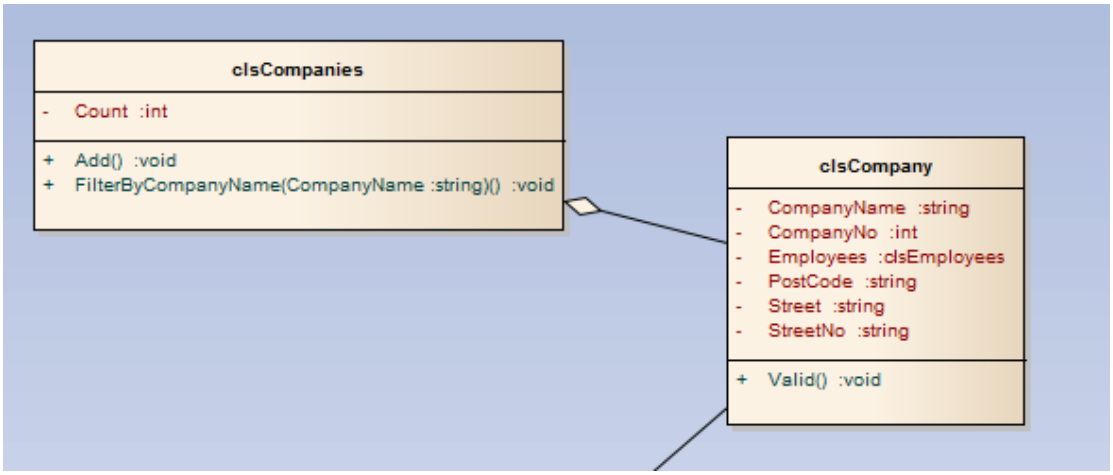
clsCompanies uses clsCompany.

OK – so what happens if we want to deal with one company and we don’t need an instance of clsCompanies, how do we validate the company details?

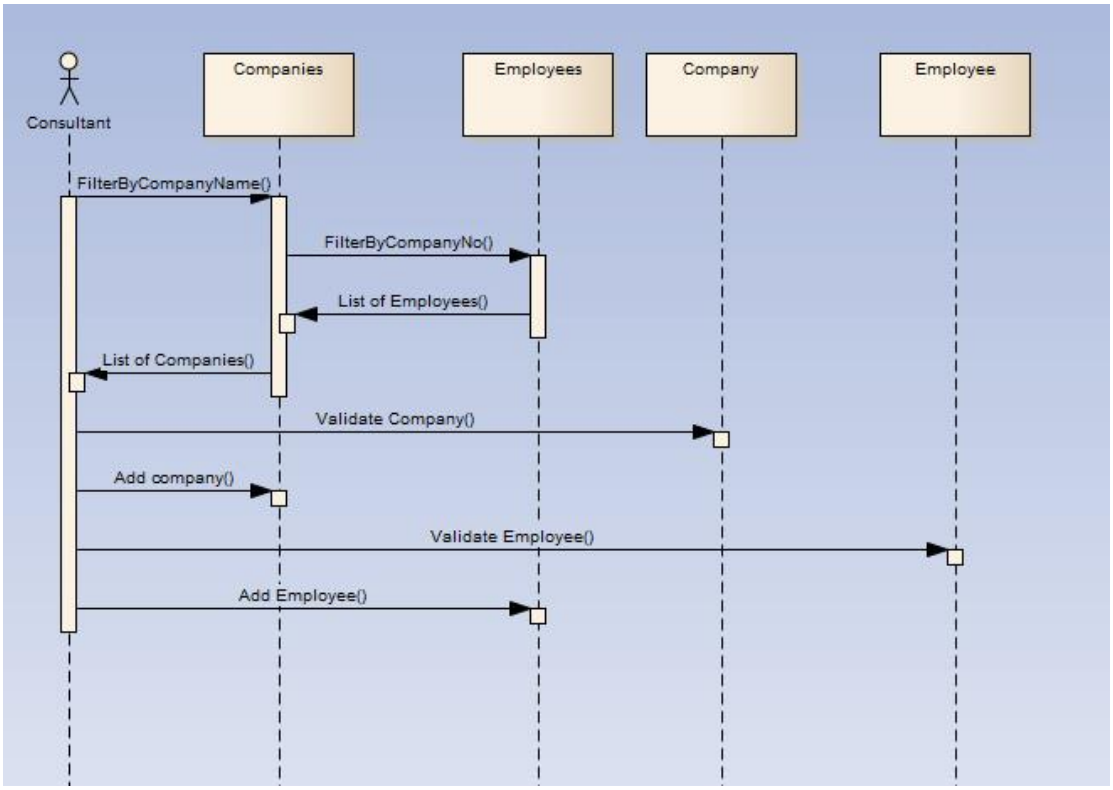


The answer is that we can't – we have placed the validation inside the wrong class!

The diagram needs to be modified like so…



The sequence diagram starts to look something like this…



It is still important to appreciate that there is a lot of work still to do. This is a voyage of discovery and we are still a long way from producing a final map of the problem domain.

It is also important to appreciate the role the sequence diagram plays in helping us to cross check and refine the use case diagram and class diagram.

Refrensi :

1. https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/
2. http://www.tutorialkampus.com/2014/06/aplikasi-penjualan-barang-di-toko-dikka.html
3. https://www.lucidchart.com/pages/uml-sequence-diagram