

## PERTEMUAN 13

### IMPLEMENTASI PERANCANGAN UML LANJUTAN SEQUENCE DIAGRAM, COLLABORATION DIAGRAM, COMPONENT DIAGRAM, DEPLOYMENT DIAGRAM

#### A. TUJUAN PEMBELAJARAN

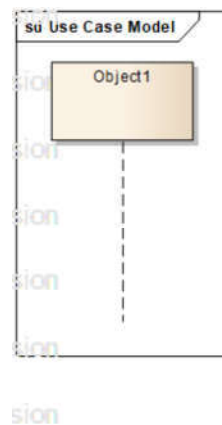
Pada pertemuan ini dijelaskan tentang pengertian dan tata cara merancang sebuah diagram UML lanjutan Sequence Diagram, Collaboration Diagram, Component Diagram, Deployment Diagram.

#### B. URAIAN MATERI

##### 1. Sequence Diagram

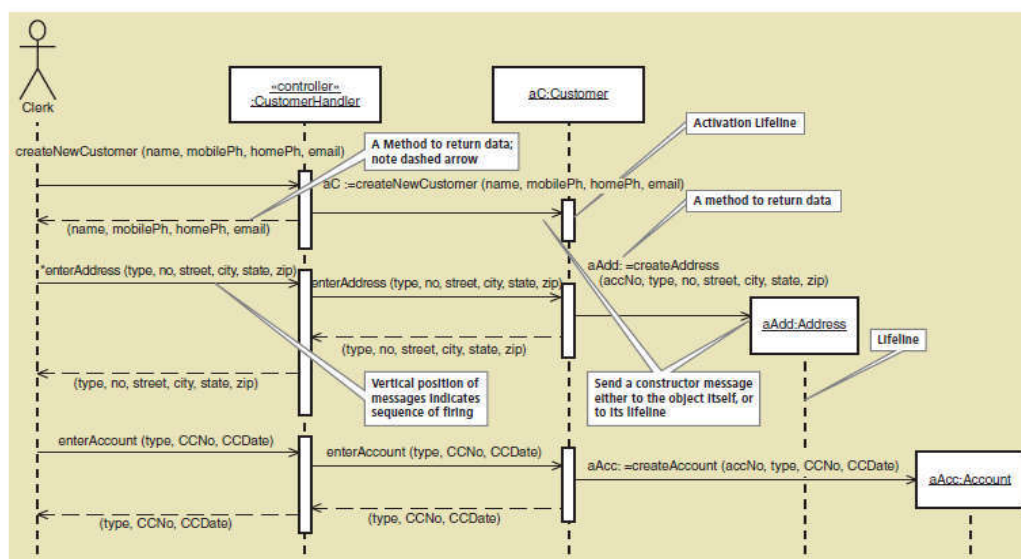
Sequence Diagram digunakan untuk menggambarkan tingkah laku yang berada di dalam skenario. Diagram ini menjelaskan beberapa sampel objek dan pesan yang terletak antara objek yang berada di sebuah use case. Elemen inti sequence diagram biasa dideskripsikan dengan warna kotak persegiempat bernama. Setiap pesan atau *Message* terwakili dengan sebuah garis tanda panah. Objek diletakkan dekat bagian atas diagram mengurut dari kiri ke arah kanan.

Sequence Diagram menunjukkan interaksi objek yang diatur dalam urutan waktu. Sequence Diagram menunjukkan sedikit perbedaan dari yang ada pada Collaboration Diagram disebut diagram Objek orientasi. Dimana disini secara signifikan Sequence Diagram menjelaskan secara abstrak tentang apa saja yang terjadi di balik suatu proses yang terjadi secara berurutan. Sequence diagram mempunyai kemiripan dengan Collaboration Diagram. Mempunyai pandangan yang berbeda tapi mempunyai pemodelan diagram yang sama. Berikut ini sebuah contoh gambar participant yang ada pada Sequence Diagram.



Gambar 33. sebuah participant pada sequence diagram

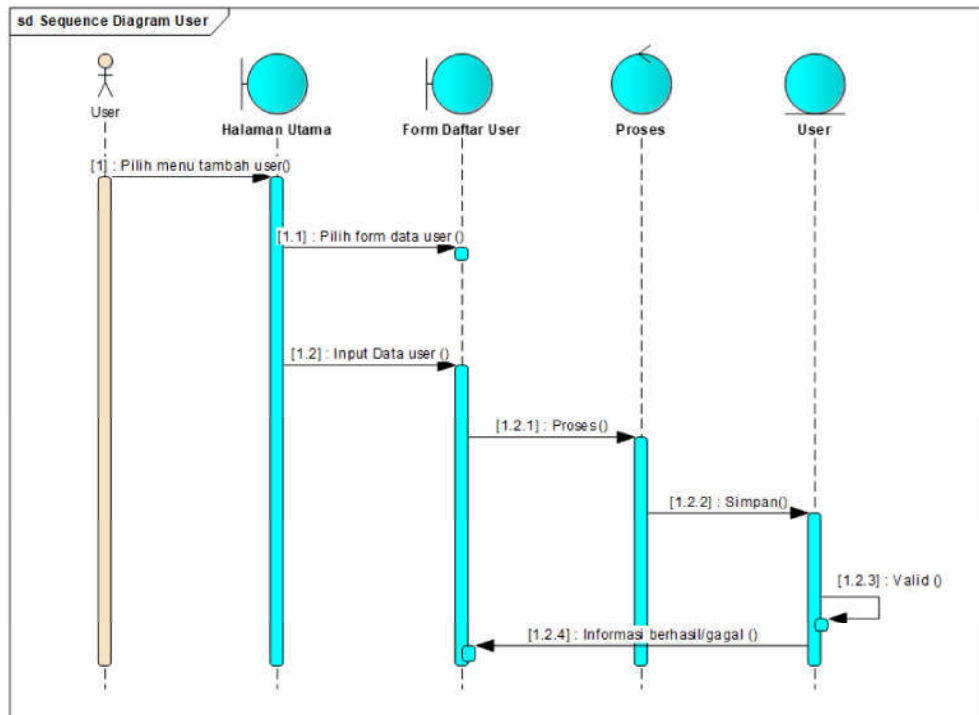
Lalu adanya penandaan aktifitas yang dilambangkan seperti panah, panah tersebut bergerak berurutan dari satu participant ke participant lain dan dari satu lifeline ke lifeline yang lain. Sebuah participant memungkinkan untuk mengirim sebuah *Message* kepada diri sendiri. Penandaan panah tersebut ada 3 macam yaitu, *simple*, *synchronous*, *asynchronous*. Untuk Message simple perpindahan control dari 1 participant ke participant lain. Jika 1 participant mengirim message *synchronous*, maka jawaban ditunggu dan diproses oleh sistem dengan urusan aktivitasnya. Jika yang dikirimkan adalah message *asynchronous*, maka jawaban message tak perlu ditunggu. Sequence Diagram ini bertujuan untuk menunjukkan urutan waktu pesan dari suatu obyek ke obyek lain. Berikut salah 1 contoh sequence diagram untuk pembuatan akun customer.



Gambar 34. Contoh Sequence diagram pembuatan akun costumer

Bisa dilihat ada sebuah aktor, dan 4 partisipan didalamnya, pada sequence diagram dijelaskan detail proses apa saja yang terjadi disetiap participant. Seperti yang ada digambar pembuatan baru akun costumer, penginputan data kelengkapan diri, teridentifikasi sebagai sebuah sebuah aktifitas yang terjadi pada sequence diagram dan berurutan tiap prosesnya.

Selanjutnya akan ada studi kasus yang akan dijelaskan pada gambar dibawah



Gambar 35. Sequence diagram menu registrasi akun login

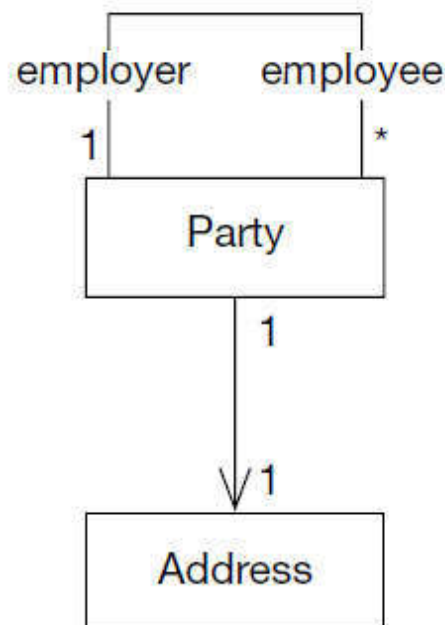
Bisa dilihat participant pada gambar diatas dapat digantikan dengan simbol-simbol lingkaran. Gambar menunjukan tiap-tiap urutan proses participant terdeskripsi dan tiap proses diurutkan melalui panah-panah. Panah tersebut pada akhirnya akan mencapai titik finish yang menandakan berakhirnya sebuah proses yang terjadi pada sistem tersebut.

Disimpulkan bahwa sequence diagram merupakan elemen terpenting dari proses analisa. Dapat mengijinkan perancang diagram untuk melakukan percobaan dan teori-teori serta pendemonstrasian sebagaimana sebuah class dapat berinteraksi dan mengirim secara spesifik perilaku yang berada pada sistem.

## 2. Collaboration Diagram

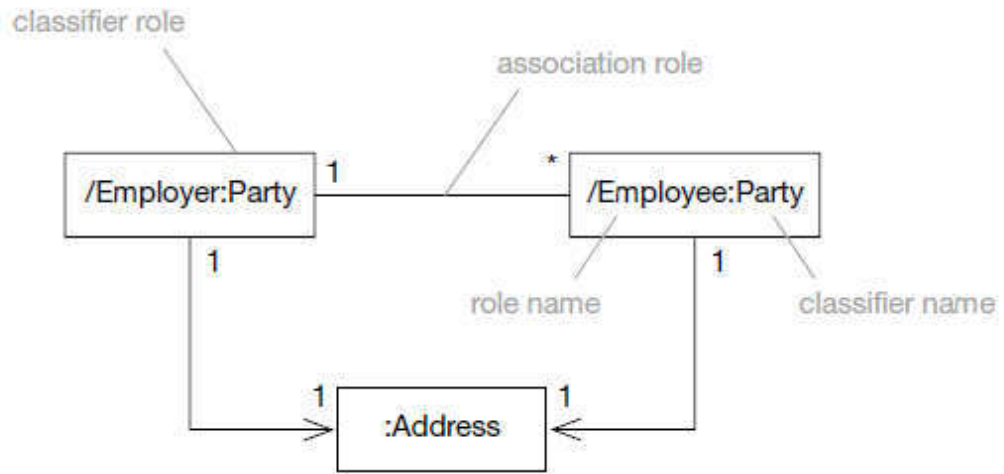
Collaboration diagram digunakan untuk menunjukkan bagaimana suatu obyek berinteraksi untuk menjalankan tingkah laku use case tertentu. Hampir sama pemodelan yang ada pada Sequence diagram tetapi mempunyai perilaku yang berbeda. Pada Collaboration Diagram alur-alur berjalannya sebuah aktifitas digambarkan secara umum atau mudah dimengerti disertai dengan deskripsi dilapangan yang terjadi dalam penggunaan suatu sistem tersebut. Tidak seperti Sequence Diagram yang menjelaskan tiap-tiap aktivitas secara abstrak pada suatu sistem tetapi pada Collaboration Diagram berfungsi menjelaskan kepada user bagaimana alur-alur berjalannya aktivitas yang sedang terjadi pada sistem tersebut. Collaboration Diagram berfokus terhadap aspek struktur yang ada pada interaksi objek. Collaborasi diagram memiliki dua bentuk sebagai *descriptor form*, dan *instance form*. *Descriptor form* memberikan pandangan yang umum terhadap *collaboration* dengan menentukan peran yang dimainkan oleh peristiwa atau bisa disebut sebagai pengklasifikasian sebuah peran.

Berikut ini contoh rupa bentuk *descriptor form* yang ada pada Collaboration Diagram.



Gambar 36. Contoh rupa *descriptor form* pada Collaboration Diagram

Kemudian *descriptor form collaboration diagram* bermodelkan seperti gambar dibawah.

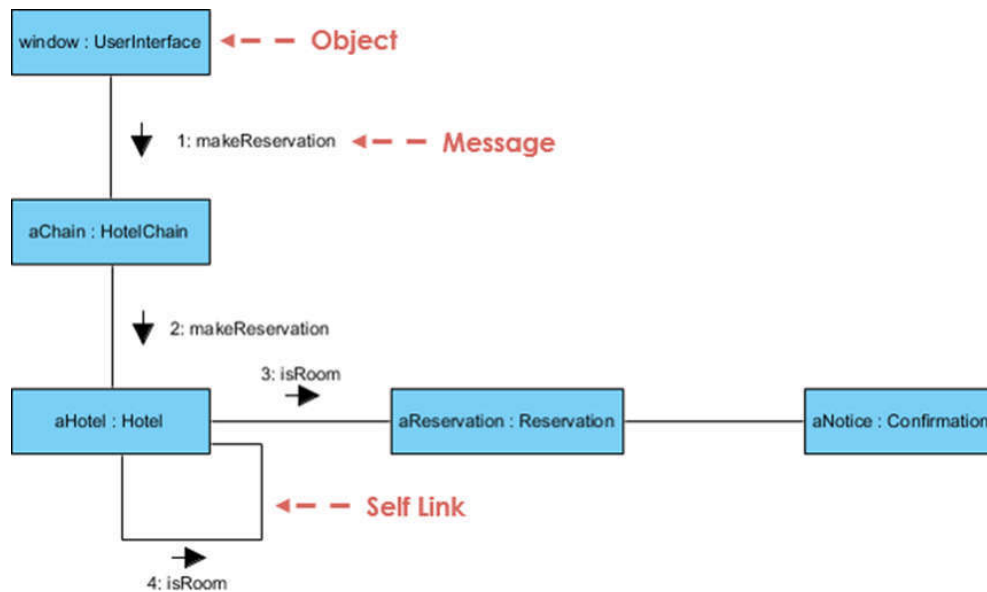


Gambar 37. Descriptor form Collaboration Diagram

Diagram ini menunjukkan peran pengklasifikasi dan peran asosiasi/*association*. Peran pengklasifikasi adalah peran yang dimainkan oleh *instance* dari pengklasifikasi, dan peran asosiasi mengindikasikan peran yang dapat dimainkan oleh *instance* pada asosiasi.

Meskipun diagram ini berisi banyak informasi yang sama dengan Class Diagram, Diagram ini menekankan berbagai peran yang dimainkan oleh *instance* di tiap classnya.

Berikut ini akan dijelaskan studi kasus Collaboration diagram pada sistem reservasi/pemesanan kamar hotel.



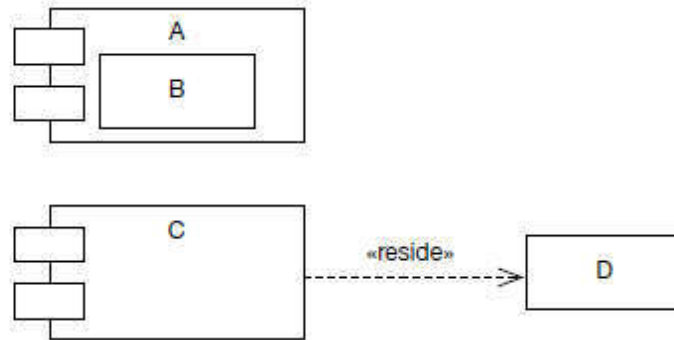
Gambar 38. Contoh Collaboration Diagram sistem pemesanan kamar hotel

Terlihat pada gambar diatas alur-alur pergerakan aktivitas dijelaskan langsung dengan keadaan aslinya, yang sebelumnya pada Sequence Diagram hanya dijelaskan apa yang terjadi dibalik satu aktivitas secara abstrak, kini pada collaboration diagram dijelaskan secara menyeluruh pergerakan aktifitas sampai ke berakhirnya suatu aktifitas pada sebuah diagram.

### 3. Component Diagram

Component diagram digunakan dalam permodelan aspek bentuk dari sistem yang berorientasi objek yang digunakan untuk memvisualisasikan atau menggambarkan, menentukan, dan mendokumentasikan sistem berbasis komponen dan juga membangun sistem yang dapat dijalankan melalui rekayasa. Component Diagram pada dasarnya adalah class diagram yang berfokus pada komponen sistem yang sering digunakan untuk memodelkan tampilan implementasi secara statis pada suatu sistem. Elemen-elemen yang ada pada sebuah sistem seperti contoh, *sourcefile*, *implementation subsystems*, *AactiveX controls*, *JavaBeans*, *Enterprise JavaBeans*, *Java servlets*, *Java Server Pages*, Elemen tersebut dibuatkan sebuah diagram dan juga mendeskripsikan alur-alur aktivitas yang terjadi didalam suatu sistem dari awal hingga akhir.

Pada dasarnya component diagram adalah sebuah diagram yang menjelaskan isi komponen yang ada pada sebuah sistem yang ingin dibuat. Berikut salah satu contoh pemodelan penggambaran sebuah Component Diagram.



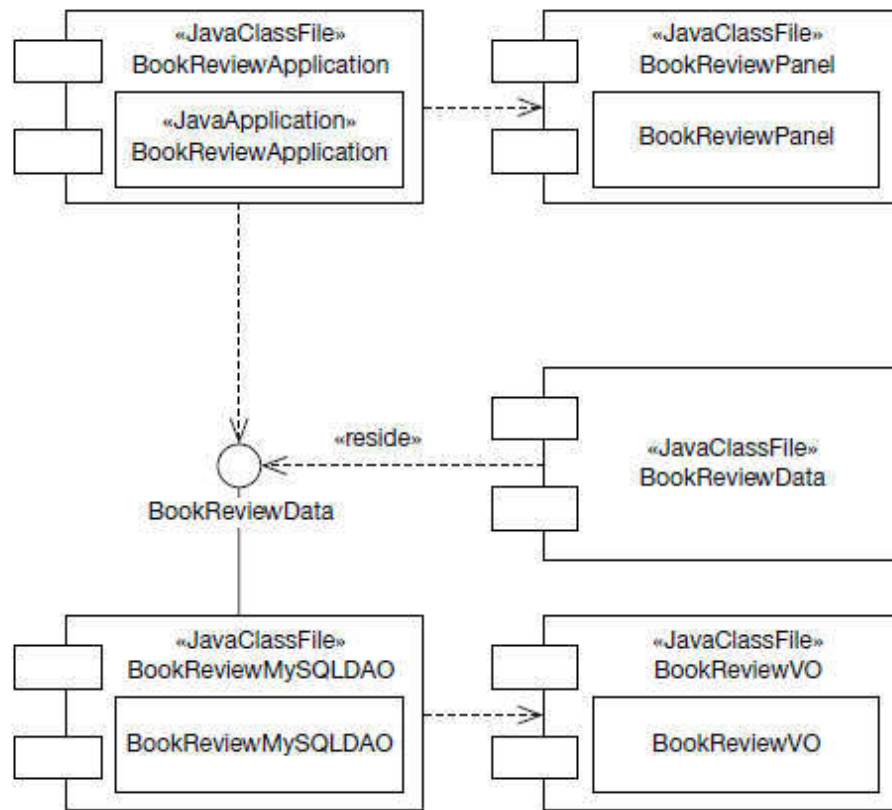
Gambar 39. Contoh pemodelan pada Component Diagram.

Setiap komponen model menunjukkan beberapa class dan interface bertugas sebagai komponen. Fungsi ini dapat menunjukkan penugasan tiap class komponen maupun secara fisik yang digambarkan dengan ikon, ataupun sebagai ketergantungan antara komponen dan class pada gambar diatas tertulis '`<<reside>>`'. Seperti contoh diatas komponen C bergantung dengan komponen D. Pada Component Diagram hanya ada *descriptor form*. Dikarenakan komponen adalah bagian fisik yang ada pada suatu sistem. Hanya komponen hanya dapat diunjukkan dengan sebuah *instance*/contoh yang tergabung kedalam wujud perangkat fisik.

#### a. Contoh pada sistem Java

Disesi kali ini akan mengilustrasikan UML component yang ada pada sistem Java. Akan ditunjukkan pemodelan simple yang ada pada aplikasi Java database. Aplikasi ini menggunakan *Data Access Objec* (DAO). Rencana dibalik pola ini DAO akan membuat sebuah kode akses secara spesifik ketika mengakses kedalam database. Aktivitas DAO terjadi dibelakang layar atau tidak berada dalam interface. Yang nantinya proses pada DAO akan menghasilkan sebuah data visual kedalam *interface*/layarmuka. Dengan membuat proses DAO didalam database, pengguna tidak akan kebingungan ketika mengakses suatu data yang ingin ditampilkan pada *interface*.

Di Java, file kelas biasanya berisi kode untuk sebuah java class dan ini membuat penggambaran hubungan antara class dan komponen menjadi sederhana. Berikut implementasi contoh Component Diagram pada sebuah aplikasi.



Gambar 40. Component Diagram pada aplikasi berbasis Java.

*BookReviewApplication* merupakan komponen yang berisi Class tunggal dan juga disebut *BookReviewApplication*. Dan kelas tersebut berada pada komponen *JavaApplication* atau terindikasi dengan program yang dapat dieksekusi pada sebuah program Java. Komponen ini menggunakan *BookReviewPanel* komponen yang mempunyai class yang sama, dan berada pada GUI (*Graphic User Interface*) pada sebuah aplikasi. Class tersebut memiliki ketergantungan antara dua komponen, daripada yang ditampilkan ditampilkan, yang dimana komponen terkait erat dan berada di *logical layer* yang sama.

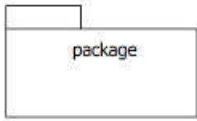
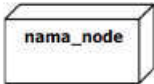


Komponen *BookReviewApplication* menggunakan tampilan dari *BookReviewData* yang berada didalam komponen *BookReviewData*. Tampilan *BookReviewData* diimplementasikan oleh komponen



*BookReviewMySQLDAO*. Komponen ini adalah sebuah DAO yang menyediakan akses ke *book Review data* yang tersimpan didalam database berelasi MySQL. Fungsi DAO lainnya dapat disediakan untuk merelasikan database lainnya asalkan keseluruhannya harus mengimplementasikan kedalam tampilan *BookReviewData*. Semua informasi file dapat dipertukarkan selama aplikasi masih bersangkutan.

b. Deployment Diagram

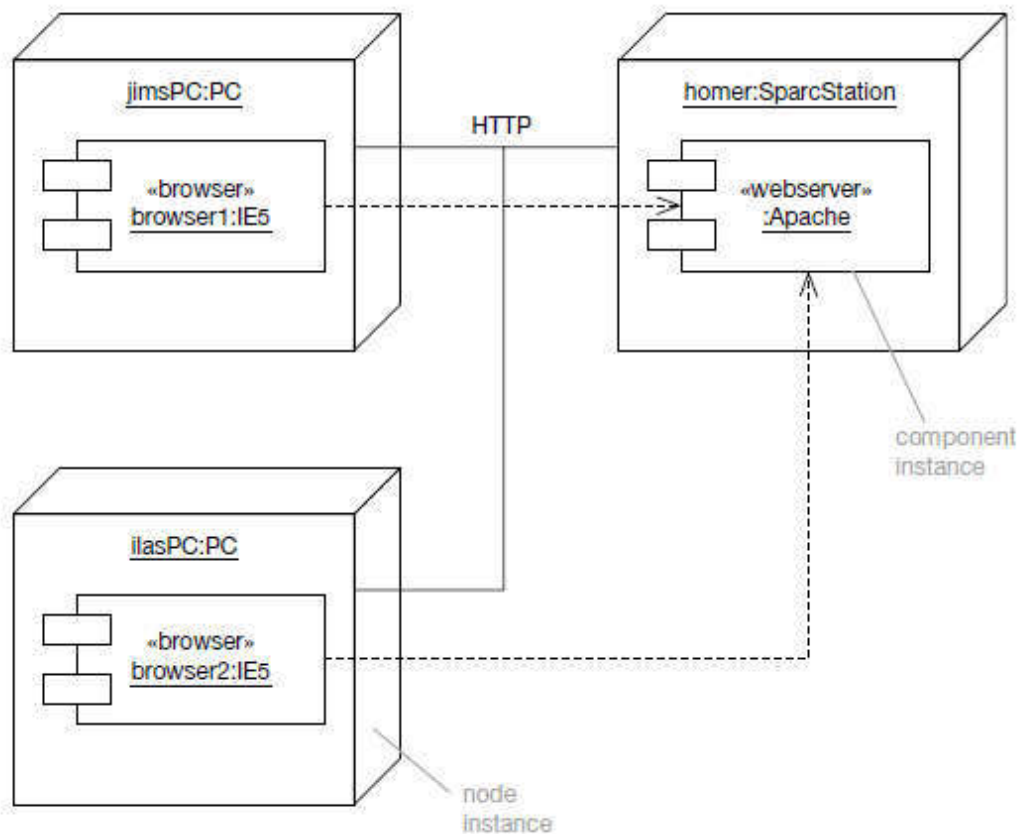
Deployment diagram adalah diagram uml yang dipakai untuk menjelaskan dan menggambarkan, menspesifikasikan serta menyimpan suatu process yang terjadi disebuah aplikasi berbasis OOP (*Object Oriented Programming*) yang akan dibangun. Dapat diartikan juga deployment diagram adalah sebuah pendeskripsian proses pada aplikasi yang sedang berlangsung terjadi serta menjelaskan bagaimana hubungan relasi di dalam sistem tersebut. Seperti yang diterangkan sebuah fungsi deployment diagram yaitu mendeskripsikan dan menggambarkan dan menspesifikasi proses yang terjadi pada sistem yang ingin dirancang. Salah satu contohnya ketika ingin menspesifikasikan sebuah situs web, maka yang perlu diperhatikan yaitu perangkat keras yang digunakan atau sebut saja dengan node pada kasus ini. Contohnya seperti Database server, Server Aplikasi, Web Server. Akan dijelaskan simbol-simbol yang ada dideployment diagram.

Simbol	Deskripsi
<b>Package</b> 	package merupakan sebuah bungkusan dari satu atau lebih <i>node</i>
<b>Node</b> 	biasanya mengacu pada perangkat keras ( <i>hardware</i> ), perangkat lunak yang tidak dibuat sendiri ( <i>software</i> ), jika di dalam <i>node</i> disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen
<b>Kebergantungan / dependency</b> 	Kebergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai
<b>Link</b> 	relasi antar <i>node</i>

**Tabel 2.** Simbol pada deployment Diagram

Didalam Package terdapat node-node yang biasanya mengacu pada perangkat keras, perangkat lunak. Komponen tersebut nantinya akan saling bergantung dengan node lain, serta dapat berelasi dengan node lain.

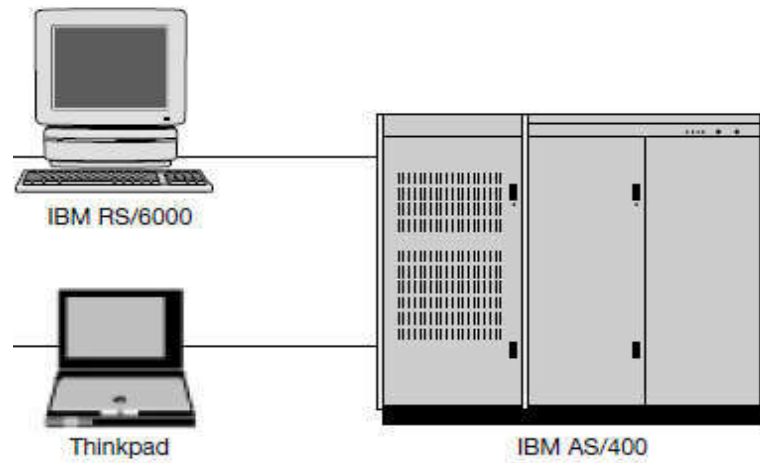
Akan dijelaskan juga contoh sederhana pengimplementasian sederhana deployment diagram yang ada pada gambar berikut.



Gambar 41. Contoh deployment diagram sederhana.

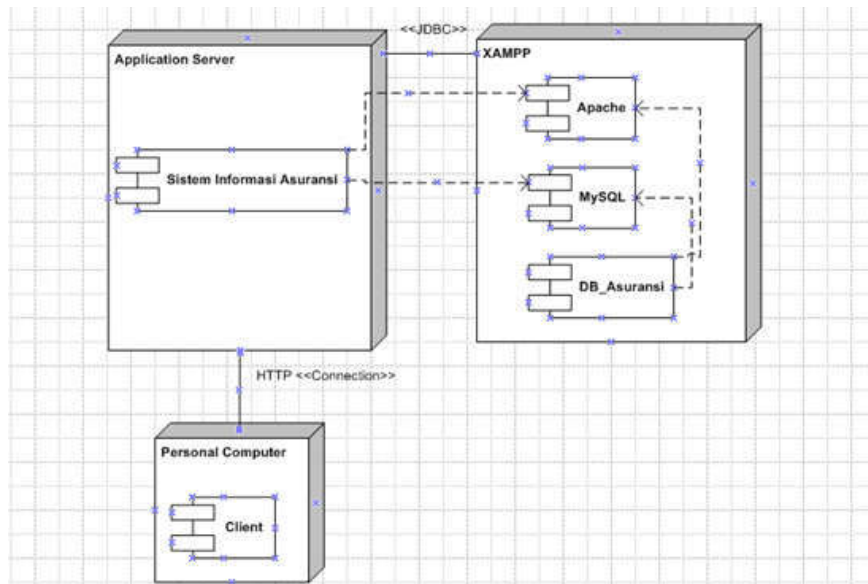
Contoh sederhana ketika seseorang ingin mengakses sebuah website sparcstation, 'jimsPC:PC' kita anggap sebuah node atau perannya disini sebagai perangkat keras, 'homer:SparcStation' sebagai alamat website yang ingin diakses oleh 'jimsPC' lewat browser, website 'homer:SparcStation' tersebut dihosting dengan super computer. Kemudian ada ilasPC:PC yang berperan sebagai admin yang mempunyai akses secara keseluruhan terhadap website homer:SparcStation dan dapat mengedit semua elemen yang ada pada website homer:SparcStation melalui device yang ia punya baik itu komputer ataupun laptop.

Jika digambarkan secara fisik maka akan terlihat seperti gambar dibawah berikut ini.



Gambar 42. Gambaran perangkat keras yang digunakan pada gambar sebelumnya

Selanjutnya akan ada studi kasus penggunaan Deployment diagram pada Sistem Informasi Asuransi.



Gambar 43. Deployment Diagram pada sistem asuransi

Bisa dilihat yang digambarkan pada Deployment diagram yaitu relasi koneksi hubungan antara, aplikasi user interface yang digunakan pengguna dengan server yang akan diakses. Kemudian dari aplikasi server, sistem akan mengambil informasi menuju database langsung

### C. SOAL LATIHAN/TUGAS

1. Carilah Definisi dari Sequence Diagram, Collaboration Diagram, Component Diagram, Deployment Diagram selain yang disebutkan diatas!
2. Jelaskan perbedaan antara Sequence Diagram & Collaboration Diagram menurut kalian!
3. Jelaskan perbedaan antara Component Diagram & Deployment Diagram menurut kalian!
4. Buatlah Studi kasus contoh untuk pembuatan Sequence Diagram!
5. Buatlah Studi kasus contoh untuk pembuatan Collaboration Diagram!
6. Buatlah Studi kasus contoh untuk pembuatan Component Diagram!
7. Buatlah Studi kasus contoh untuk pembuatan Deployment Diagram!
8. Buatlah Relasi Deployment Diagram yang tergambarkan secara fisik!

### D. REFERENSI

- John W. Satzinger, Robert B. Jackson, Stephen D.Burd (2016). *Systems Analysis and Design In A Changing World*. 7th edition. Boston: Cengage Learning.
- Jim Arlow, Ila Neustadt (2002). *UML And The Unified Process Practical Object-Oriented Analysis & Design*. Great Britain: Pantek Arts,Ltd, Maidstone, Kent

## GLOSARIUM

**Sequence** merupakan sebuah antrian yang harus dilalui ketika sistem berjalan

**Message** merupakan sebuah pesan atau pendeskripsian hal-hal apa saja yang terjadi pada sebuah sequence diagram.

**Participant** adalah sebuah elemen pada sequence diagram, yang tiap elemennya mempunyai tugas masing-masing.

**Source file** adalah sebuah sumber data yang ada pada database atau penyimpanan data.

**JavaBeans** merupakan sebuah elemen suatu program yang menggunakan bahasa Java.

**Java** merupakan bahasa pemrograman dasar pada JDK (Java Development Kit)

**Object oriented programming** merupakan bahasa pemrograman yang berfokus pada orientasi serta objek.

**Database Server** merupakan server penyimpanan data yang dipakai untuk kebutuhan sebuah sistem ataupun website.

**Web Server** merupakan server yang berfungsi untuk menghosting suatu website agar dapat dipublikasikan kedalam jaringan internasional atau *Internet*.

**Pc** merupakan istilah lain dari personal computer, atau perangkat komputer milik pribadi.

**Xampp** adalah sebuah software yang dapat menghosting suatu database, ketika seseorang membuat aplikasi berbasis *CRUD* (*create, read, update, delete*).

**MySql** merupakan salah satu elemen aplikasi yang berada pada Xampp, berfungsi sebagai penyimpanan data.