

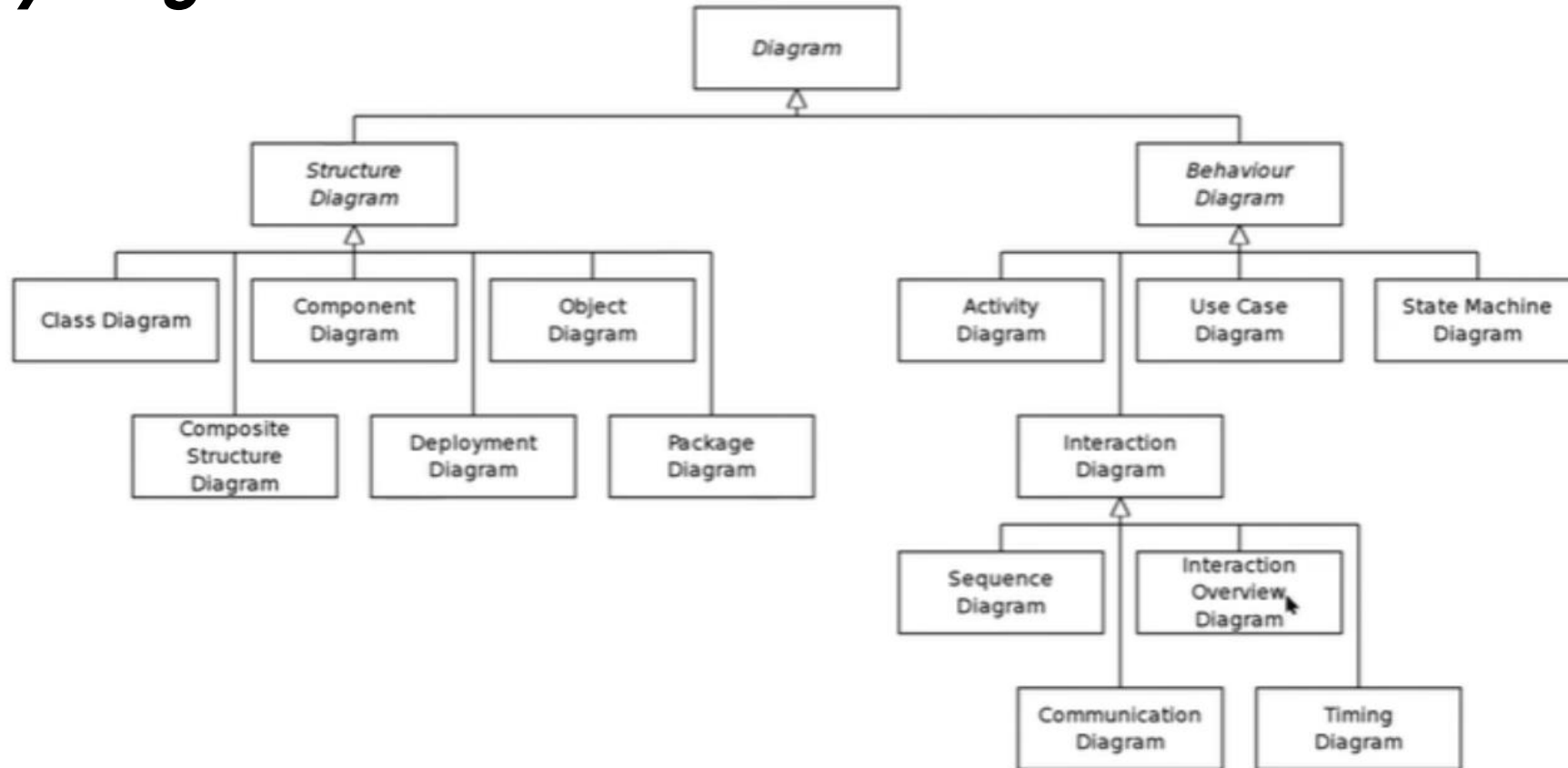


UNIVERSITAS IPWIJA

Pemodelan Analisa Dengan Pendekatan Berorientasi Objek (2)

#1 Use Case Diagram

#2 Activity Diagram



#3 Class Diagram

Class diagram adalah jenis diagram struktur statis dalam UML yang menggambarkan struktur sistem dengan menunjukkan sistem class, atributnya, metode, dan hubungan antar objek.

Sukamto dan Shalahuddin (2013:141)

Class diagram adalah penggambaran dari struktur sistem serta mendefinisikan kelas yang hendak dibangun system.

Satzinger (2011: 28)

Diagram kelas ialah alat untuk membantu menggambarkan serta memvisualisasikan struktur kelas yang terdapat dalam suatu sistem atau program.

Mengapa demikian? Sebab class diagram ini mendeskripsikan kelompok objek bersama properti, operasi dan relasi yang sama.

Class Diagram

Class diagram disebut jenis diagram struktur karena menggambarkan apa yang harus ada dalam sistem yang dimodelkan dengan berbagai komponen.

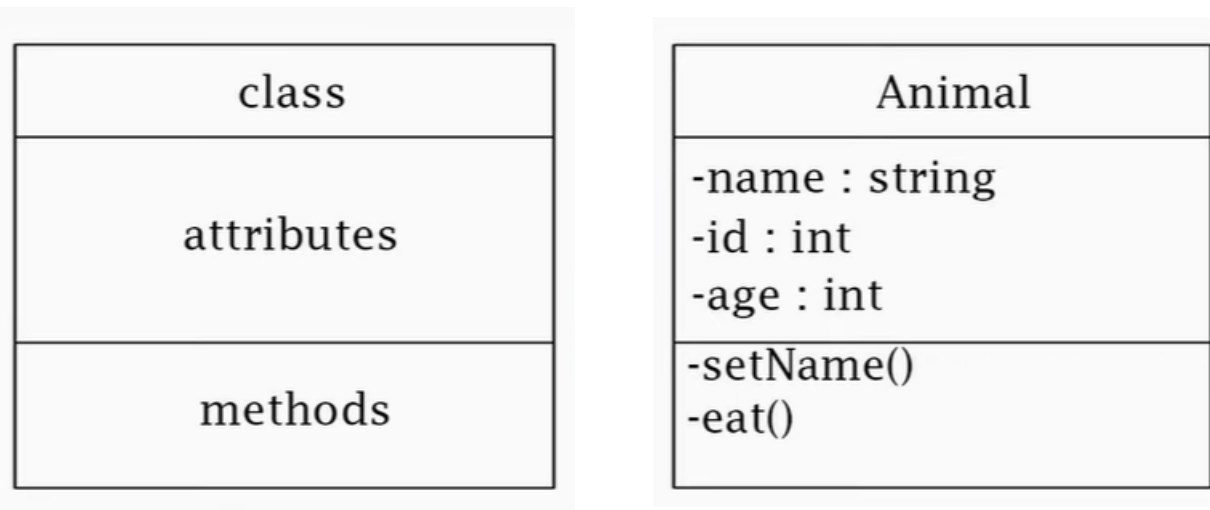
Berbagai komponen tersebut dapat mewakili class yang akan diprogram, objek utama, atau interaksi antara class dan objek.

Class sendiri merupakan istilah yang mendeskripsikan sekelompok objek yang semuanya memiliki peran serupa dalam sistem.

Sekelompok objek ini terdiri atas **fitur struktural** yang mendefinisikan apa yang diketahui class dan **fitur operasional** yang mendefinisikan apa yang bisa dilakukan oleh class.

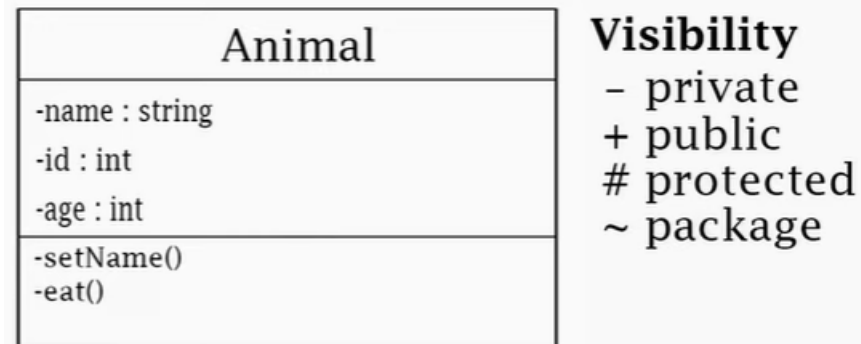
Komponen Class Diagram

- Terdapat **3 komponen** dalam sebuah class diagram, sebagai berikut:
 - **Class name**, merupakan sesuatu yang mewakili dari nama kelas.
 - **Atribut**, merupakan properti dari sebuah kelas, atribut melambangkan batas nilai kelas yang mungkin terdapat dalam objek kelas.
 - **Proses atau method**, sesuatu yang dapat dilakukan atau diproses oleh sebuah kelas.



Komponen Class Diagram

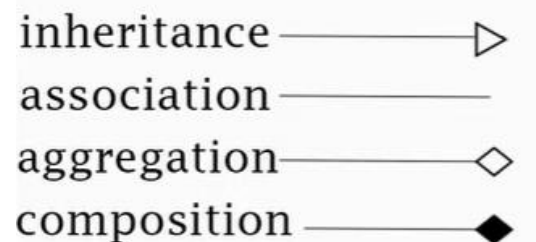
- Selain itu, **atribut** dan **proses** dapat memiliki salah satu **sifat/visibility** berikut ini:
- **Private**, tidak dapat dipanggil atau digunakan oleh kelas lain, itu artinya hanya dapat digunakan oleh kelas sendiri.
 - **Public**, dapat dipanggil dan digunakan oleh siapa saja yang memerlukannya.
 - **Protected**, hanya dapat dipanggil dan digunakan oleh kelas-kelas bersangkutan serta turunannya dari kelas itu sendiri.
 - **Package**, hanya dapat dipanggil dan digunakan oleh instance sebuah kelas di paket yang sama.



Hubungan Antar Class

Pada class diagram terdapat **4 hubungan antar class** sebagai berikut:

- **Pewarisan**, yaitu suatu hubungan hirarki antar class, yang mana dapat diturunkan dari class lain dan mewarisi semua atribut. Hal ini disebut juga sebagai class child atau anak class.
- **Asosiasi**, yaitu hubungan yang sifatnya statis di antara class. Biasanya memperlihatkan class yang mempunyai atribut lain. Bisa juga sebuah class yang harus mengetahui ekstensi class lain, panah navigasi yang ada sebagai penunjuk arah query.
- **Agresi**, yaitu suatu hubungan yang menyatakan bagian (terdiri atas)
- **Hubungan dinamis**, yaitu serangkaian pesan atau message yang dipasing ke class lain. Hubungan ini dapat dilihat dan dijelaskan dalam sebuah sequence diagram.



Fungsi dan Keunggulan Class Diagram

Fungsi

Fungsi utama dari class diagram adalah menggambarkan struktur sebuah sistem pemrograman.

Fungsi lainnya dari class diagram:

- Menunjukkan struktur statis pengklasifikasi dalam suatu sistem.
- Memberikan notasi dasar untuk diagram struktur lain yang ditentukan oleh UML.
- Dapat digunakan business analyst untuk membuat model sistem dari perspektif bisnis.

Fungsi dan Keunggulan Class Diagram

Keunggulan






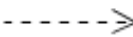

Menurut **LucidChart**, class diagram memberikan beberapa keunggulan bagi pemrograman, terutama dalam bisnis.

Berikut adalah beberapa keunggulan class diagram:

- Mampu mengilustrasikan model data untuk sistem informasi, terlepas dari apakah model data tersebut rumit atau sederhana.
- Memberikan gambaran umum mengenai skema aplikasi dengan lebih baik.
- Membantumu menyampaikan secara visual kebutuhan spesifik apa pun dari suatu sistem dan menyebarkan informasi tersebut ke bisnis.
- Terdapat bagan terperinci yang menyoroti kode spesifik yang perlu diprogram dan diterapkan ke struktur yang sesuai.
- Menyediakan deskripsi implementasi independen dari tipe yang digunakan dalam sistem untuk kemudian diteruskan di antara komponen-komponennya.

Simbol Class Diagram

SIMBOL CLASS DIAGRAM

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Kardinalitas dari class diagram

Nilai Kardinalitas	Arti	Contoh	
 0..1	Nol atau satu	karyawan	0..1 istri
1	Hanya satu	negara	1 presiden
0..*	Nol atau lebih	karyawan	0..* anak
1..*	Satu atau lebih	bos	1..* bawahan
n	Hanya n (dengan n > 1)	karyawan	n cek up
0..n	Nol sampai n (dengan n > 1)	karyawan	0..n sim
1..n	Satu sampai n (dengan n > 1)	kereta api	1..n gerbong

Multiplicity

0..1 zero to one
 n specific number
 0..* zero to many
 1..* one to many

Berikut adalah komponen-komponen yang menyusun class diagram

1. Komponen Atas

Bagian ini berisikan nama class yang selalu diperlukan baik itu dalam pengklasifikasi atau objek.

2. Komponen Tengah

Komponen ini berisikan atribut class yang digunakan untuk mendeskripsikan kualitas kelas. Ini hanya diperlukan saat mendeskripsikan instance tertentu dari class.

3. Komponen Bawah

Bagian ini adalah komponen class diagram yang menyertakan operasi class yang ditampilkan dalam format daftar. Sehingga, setiap operasi mengambil barisnya sendiri. Komponen ini juga menggambarkan bagaimana class berinteraksi dengan data.

Berikut adalah komponen-komponen yang menyusun class diagram

4. Komponen Tambahan (#1)

- **Class**, yang merepresentasikan obyek atau sekumpulan obyek yang memiliki persamaan struktur.
- **Signals**, yaitu simbol yang mewakili komunikasi asinkron satu arah antara objek aktif.
- **Tipe data** yang terbentuk dengan melakukan pengklasifikasi yang menentukan nilai data. Tipe data ini dapat menghasilkan tipe primitif dan pencacahan.
- **Packages** yang dirancang untuk mengatur pengklasifikasi terkait dalam diagram. Komponen ini dilambangkan dengan bentuk persegi panjang.
- **Interface**, yaitu sekumpulan atribut yang mendefinisikan sekumpulan perilaku yang kohesif. Komponen ini mirip dengan class, namun harus memiliki setidaknya satu class untuk mengimplementasikannya.

Berikut adalah komponen-komponen yang menyusun class diagram

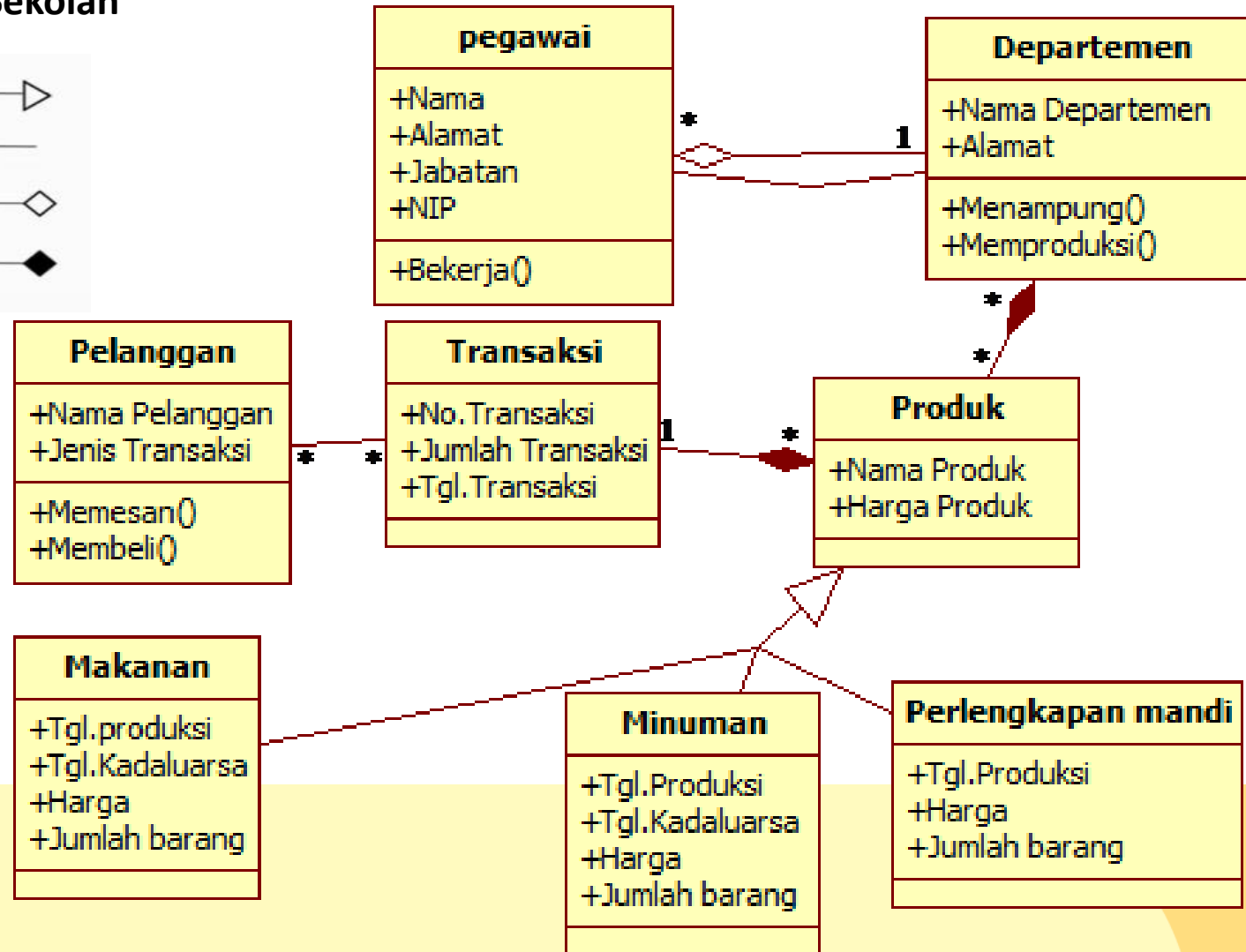
4. Komponen Tambahan (#2)

- **Enumerations**, yang menggambarkan tipe data yang ditentukan pengguna. Komponen enumerations mencakup kelompok pengenal yang mewakili nilai-nilai pencacahan.
- **Objek**, adalah item yang dapat ditambahkan ke dalam class diagram untuk mewakili contoh konkret atau prototipe.
- **Artifacts**, yaitu elemen model yang mewakili entitas konkret dalam sistem perangkat lunak. Misalnya, dokumen, database, file yang dapat dieksekusi, komponen perangkat lunak, dan sebagainya.

Contoh Class Diagram

Sistem Manajemen Sekolah

inheritance —> ▷
association ———
aggregation —◇—
composition —◆—



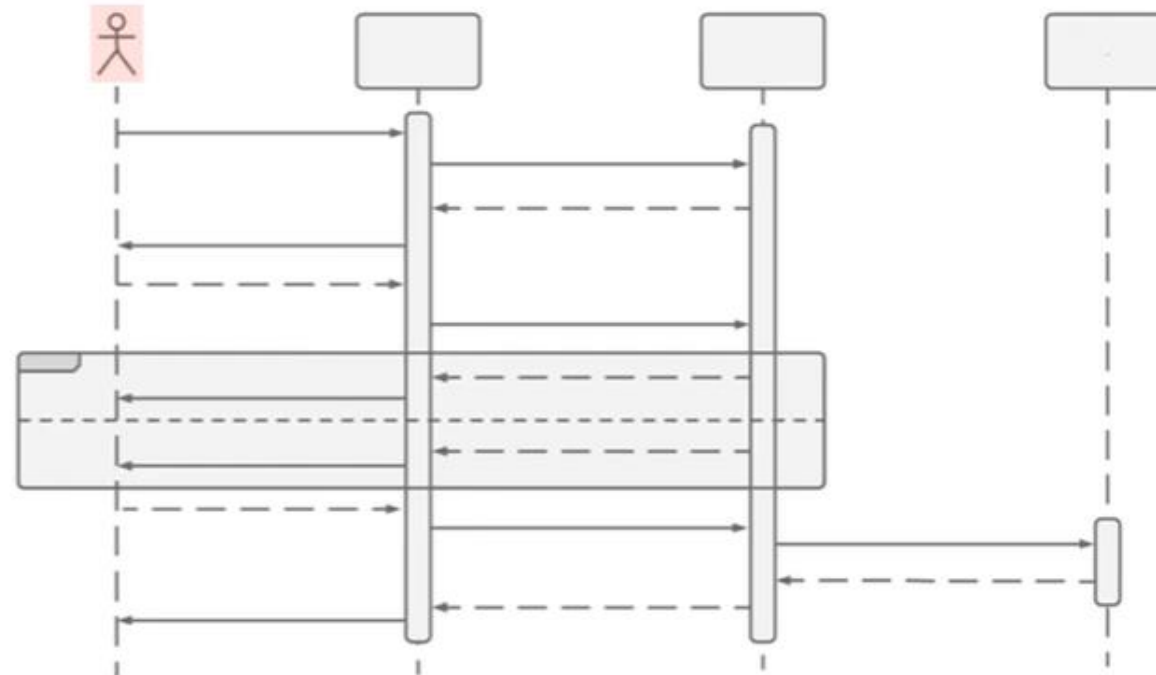
#4 Sequence Diagram

Sequence diagram menggambarkan kelakuan/perilaku objek pada use case dengan mendeskripsikan waktu hidup (lifeline) objek dan message yang dikirimkan dan diterima antar objek.

untuk menggambar sequence diagram harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

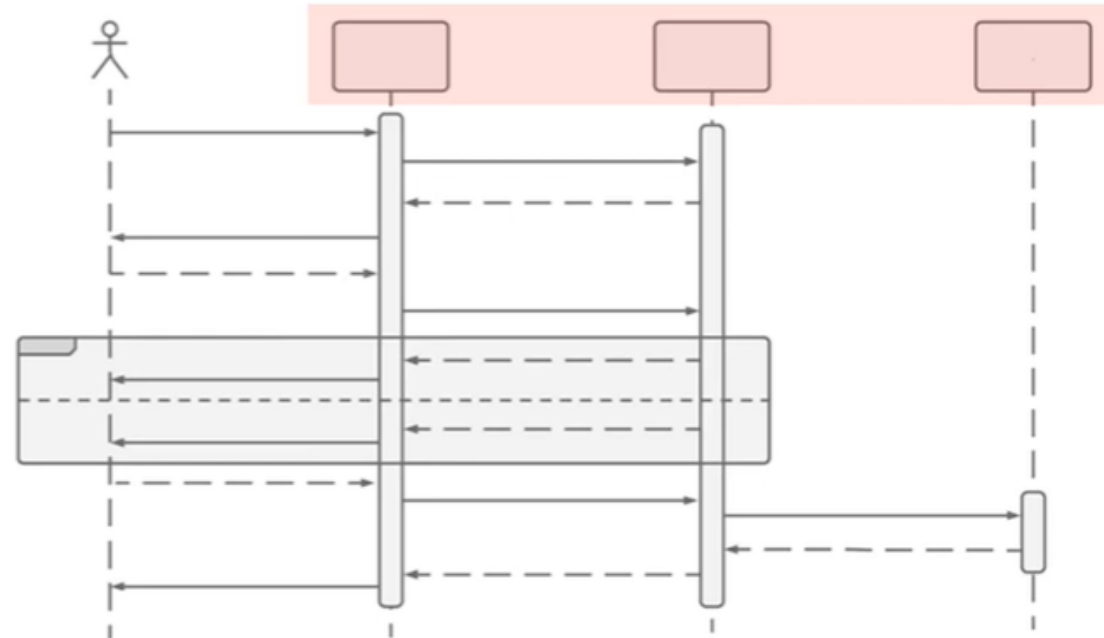
Banyaknya sequence diagram yang harus digambar adalah sebanyak pendefinisian use case yang ada.

Simbol Sequence Diagram



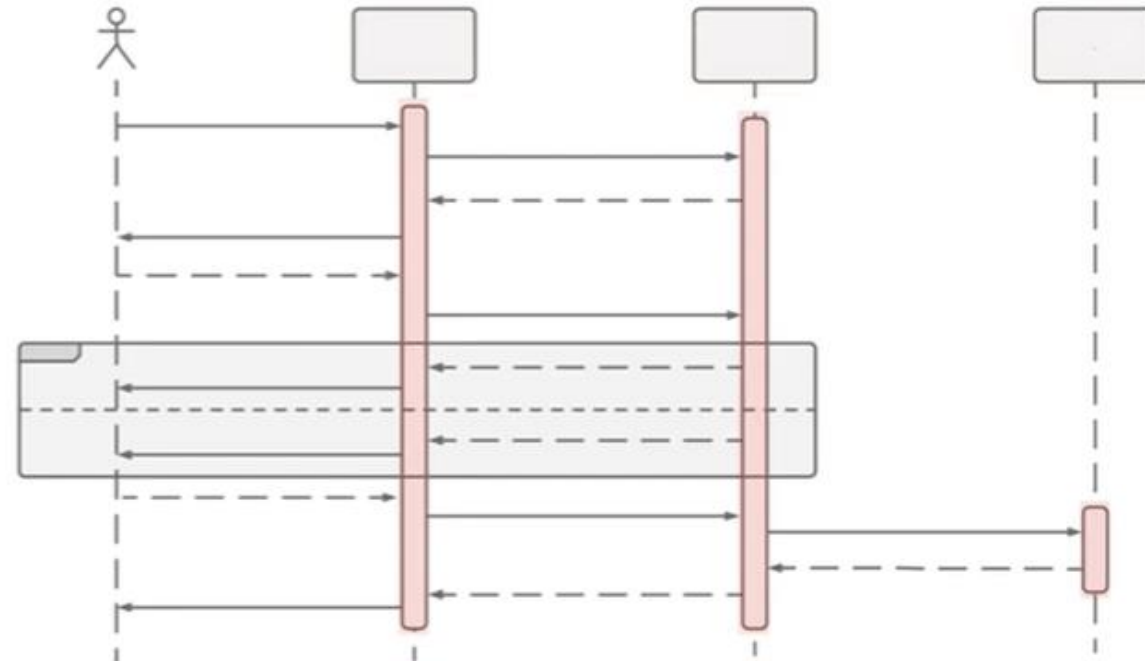
Actor : Komponen ini mewakili seorang pengguna yang sedang berinteraksi dengan system.

Simbol Sequence Diagram



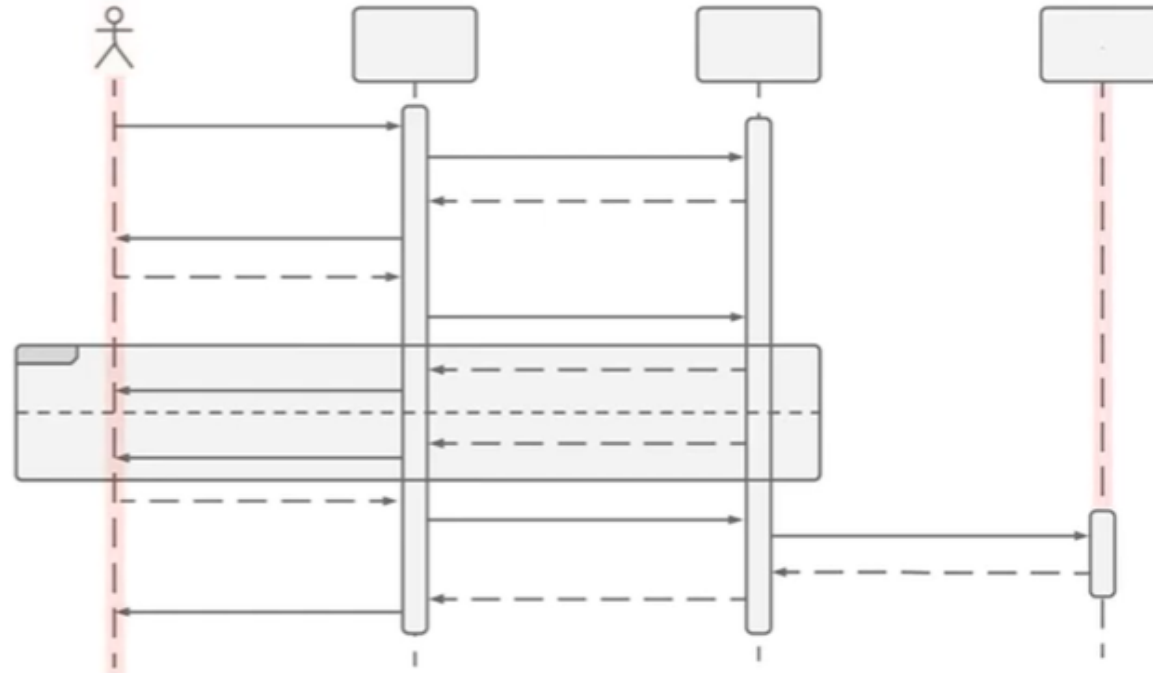
Object : kumpulan berbentuk kotak yang mewakili sebuah class atau object. Kumpulan tersebut berfungsi untuk mendokumentasikan bagaimana sebuah object berperilaku sebagai sebuah system.

Symbol Sequence Diagram



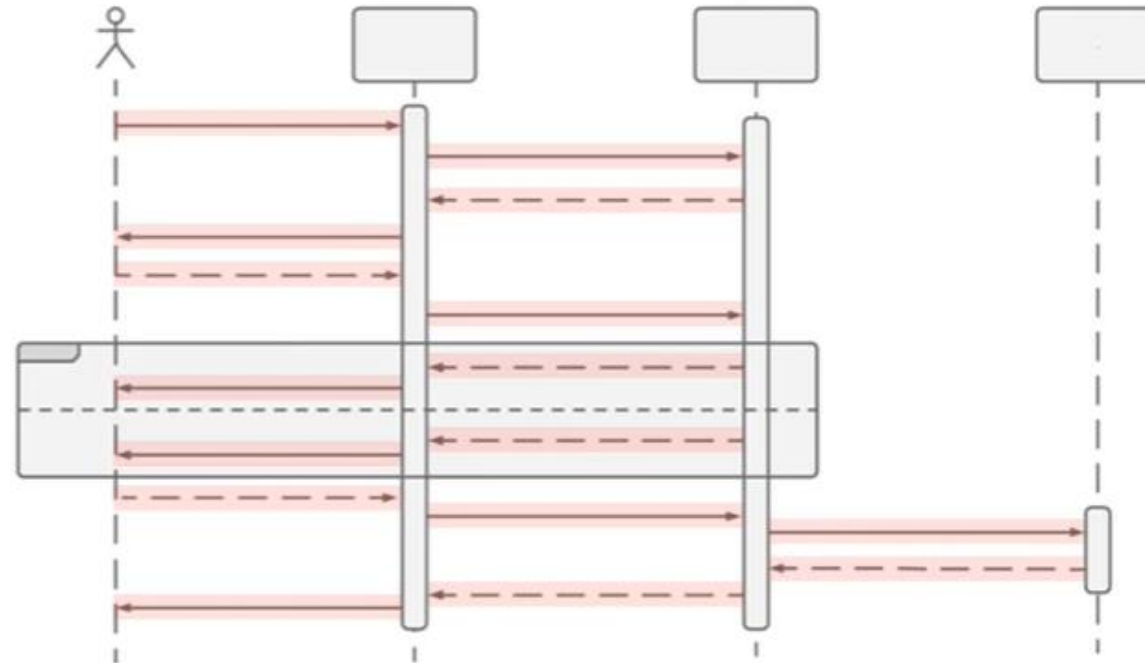
Activation boxers : komponen yang berbentuk persegi panjang yang menggambarkan waktu yang diperlukan sebuah object untuk menyelesaikan tugas.

Simbol Sequence Diagram



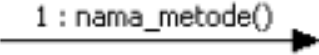
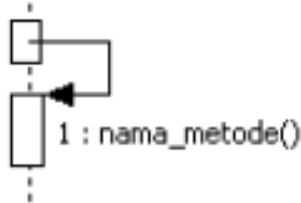

Lifeline : komponen yang berbentuk garis putus-putus. Komponen ini biasanya memuat kotak yang berisi sebuah object yang berfungsi untuk menggambarkan aktifitas dari object.

Symbol Sequence Diagram



Messege : sebuah pesan yang menggambarkan interaksi yang terjadi antara actor dengan object dan sebaliknya atau object dengan object.

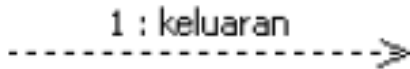

Simbol Sequence Diagram

Pesan tipe call	Menyatakan suatu objek memanggil
	<p>operasi/metode yang ada pada objek lain atau dirinya sendiri,</p>  <p>arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi</p>
Pesan tipe send	
	<p>menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>

Simbol Sequence Diagram

Pesan tipe call

Menyatakan suatu objek memanggil

<p>Pesan tipe return</p> 	<p>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</p>
<p>Pesan tipe destroy</p> 	<p>menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy</p>

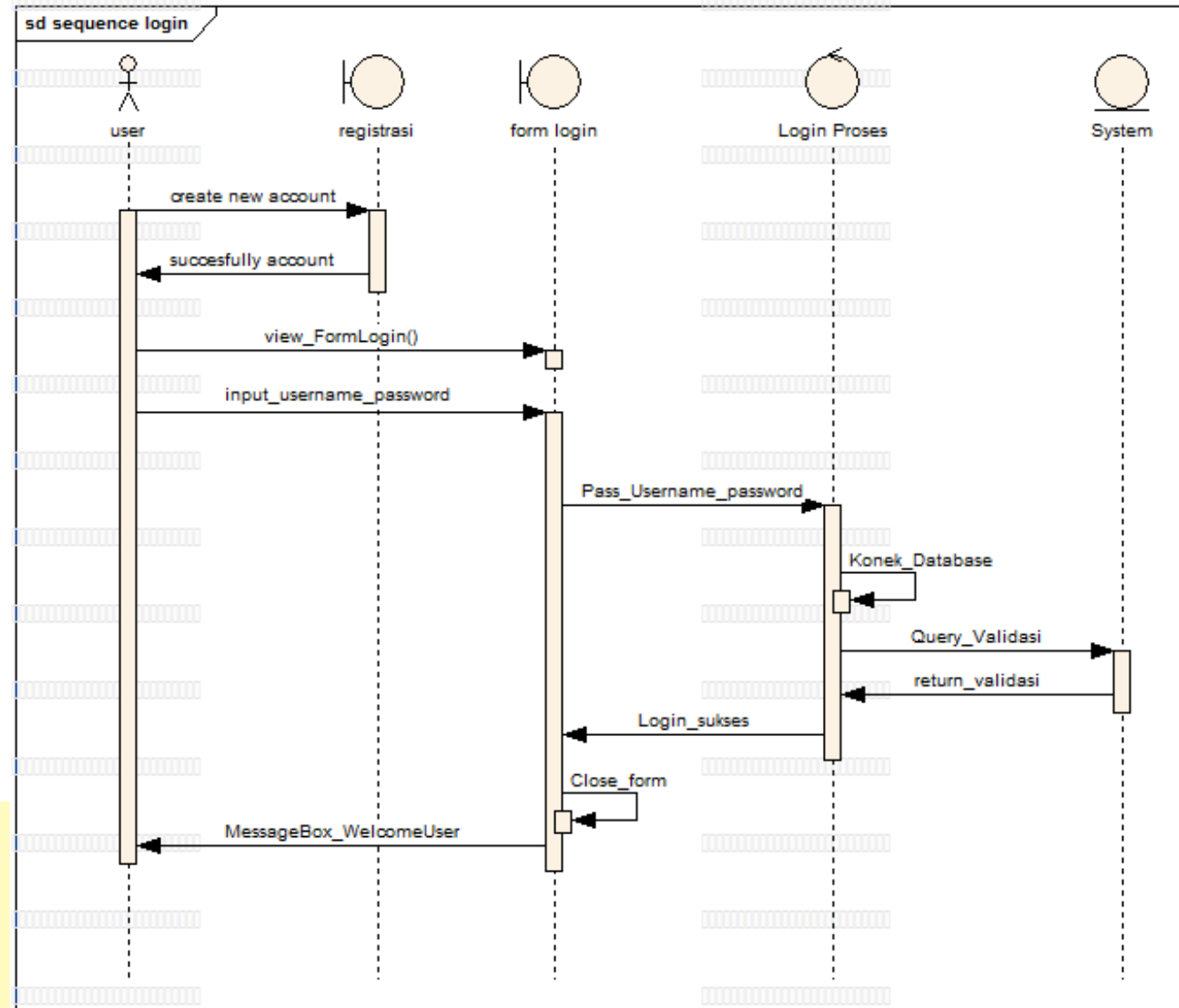
Aturan Sequence diagram #1

- Penomoran pesan berdasarkan urutan interaksi pesan.
- Penggambaran letak pesan harus berurutan, pesan yang lebih atas dari lainnya adalah pesan yang berjalan terlebih dahulu
- Pada sequence diagram terdapat garis hidup objek (life line).
- Life line adalah garis tegas vertikal yang mencerminkan eksistensi sebuah objek sepanjang periode waktu.
- Sebagian besar objek-objek yang tercakup dalam diagram interaksi akan eksis sepanjang durasi tertentu dari interaksi, sehingga objek-objek itu diletakkan di bagian atas diagram dengan garis hidup tergambar dari atas hingga bagian bawah diagram.
- Suatu objek lain dapat saja diciptakan, dalam hal ini garis hidup dimulai saat pesan Create diterima suatu objek.
- Selain itu suatu objek juga dapat dimusnahkan dengan pesan Destroy, jika kasus ini terjadi, maka life line juga berakhir.

Aturan Sequence diagram #2

- Terdapat fokus kendali (Focus of Control).
- berupa empat persegi panjang ramping dan tinggi yang menampilkan aksi suatu objek secara langsung atau sepanjang sub ordinat.
- Puncak dari empat persegi panjang adalah permulaan aksi, bagian dasar adalah akhir dari suatu aksi (dan dapat ditandai dengan pesan Return).
- Pada diagram ini mungkin juga memperlihatkan penyarangan (nesting) dan fokus kendali yang disebabkan oleh proses rekursif dengan menumpuk fokus kendali yang lain pada induknya

Contoh Sequence Diagram



#5 Package Diagram

Package diagram atau diagram paket merupakan salah satu jenis UML yang dipakai untuk mengelompokkan elemen-elemen seperti use case maupun class diagram.

Package diagram sendiri sangat jarang digunakan dalam pemuatan sistem dengan skala kecil, sebab memang tidak terlalu dibutuhkan.

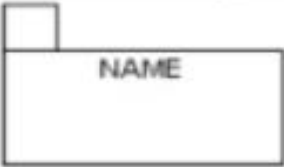
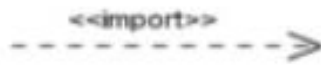
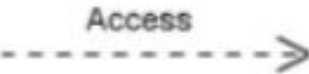
Artinya package diagram digunakan hanya dalam pengembangan sistem dengan skala besar. Tentu tujuannya sendiri agar lebih mudah dalam melakukan penyusunan.

- Package diagram dapat kita asusmsikan seperti halnya folder yang terdapat dalam operating system seperti windows maupun linux.
- Package adalah sekumpulan class atau element dengan sifat yang sama.
- Package digambarkan seperti sebuah direktori folders.

Fungsi Package Diagram

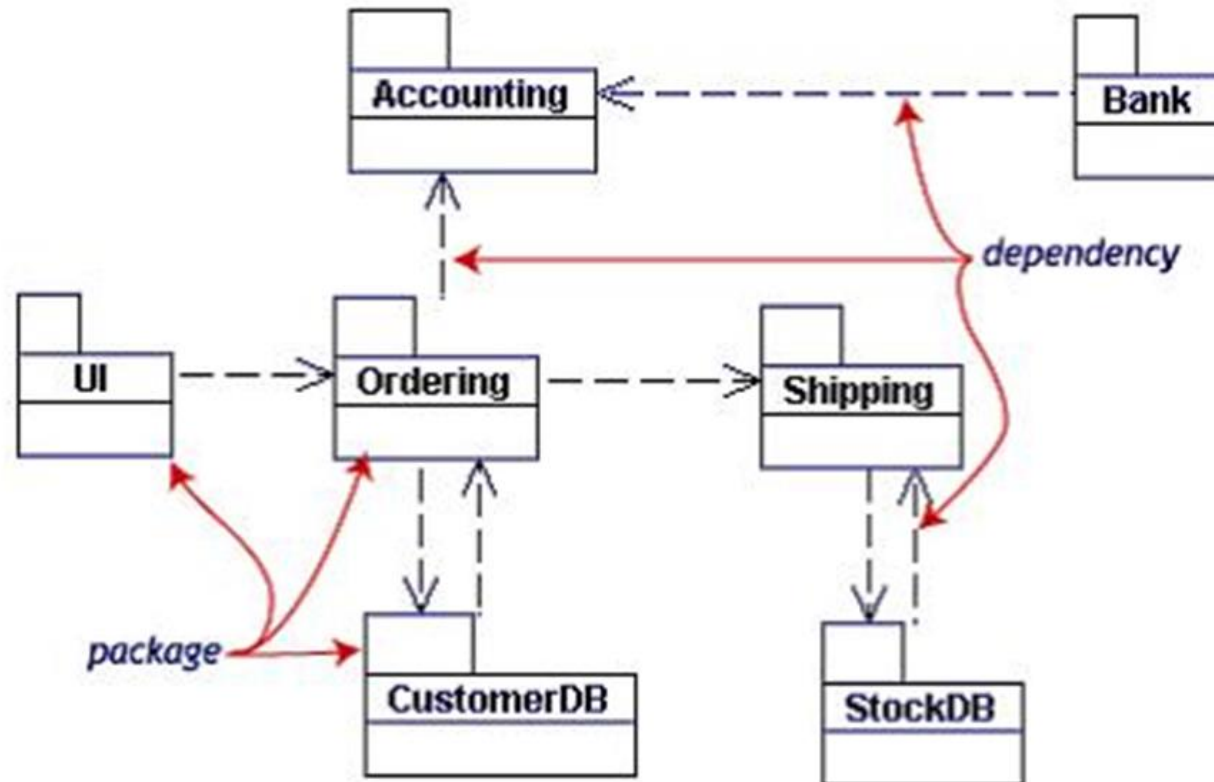
- Digunakan untuk mengelompokkan use case diagram atau class diagram.
- Untuk menunjukan sebuah elemen dikelompokkan atau diorganisasikan.
- Package diagram bisa diterapkan dalam UML apa saja.
- Menyederhanakan diagram UML, sebab dapat dikelompokkan berdasarkan elemen yang digunakan.
- Menjadikan pekerjaan lebih tertata rapih dan tidak membingungkan.
- Diagram yang dibangun lebih sederhana.

Simbol Package Diagram

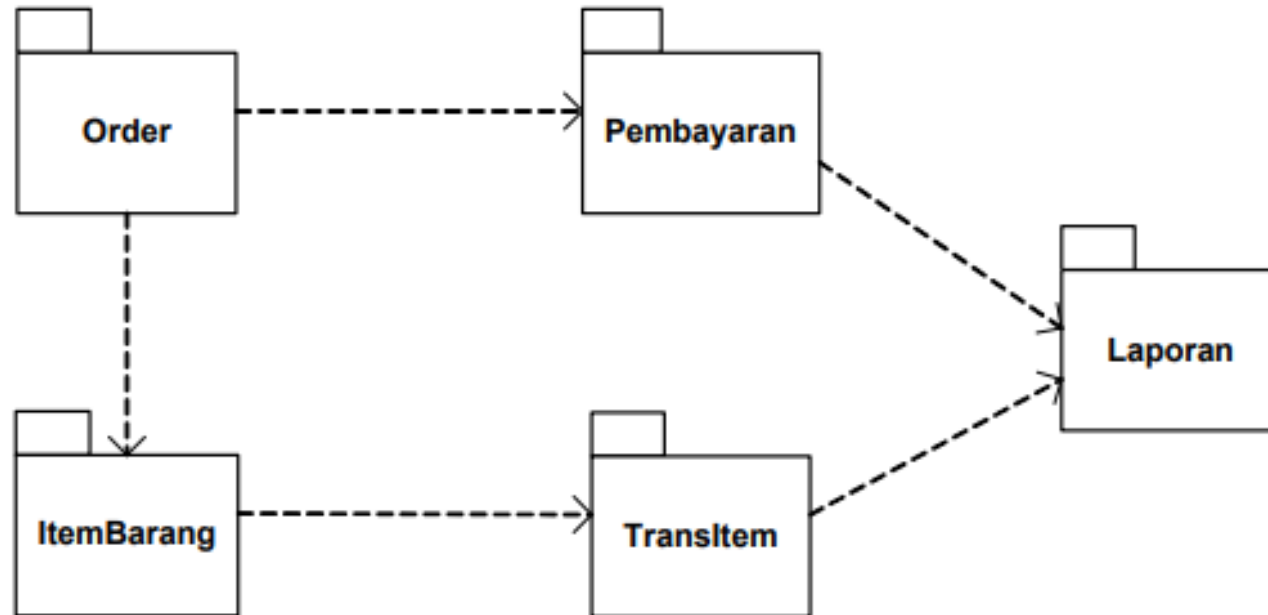
Contract	Deskripsi	Lambang
Package	Sekelompok elemen-elemen model	
Import	Suatu dependency yang mengindikasikan isi tujuan paket secara umum yang ditambahkan kedalam sumber paket	
Access	Suatu dependency yang mengindikasikan isi tujuan paket secara umum yang bisa digunakan pada nama sumber paket	

Contoh Package Diagram

Package diagram penjualan



Contoh Package Diagram



Contoh Package Diagram Sistem Galeri

Cara Membuat Package Diagram

1. ***Membaca data*** : Baca dan pahami data apa saja yang akan dijadikan package diagram
2. ***Memahami alur data*** : Pahami alur data yang hendak dijadikan package diagram
3. ***Gambar package diagram*** : Gambarlah package diagram sesuai dengan alur data

#6 Deployment Diagram

Deployment diagram adalah jenis diagram yang statis, artinya tidak akan mengalami perubahan, ketika kita merancang diagram tersebut, maka akan bertahan sampai kapanpun.

Menurut **winberllo**, deployment diagram adalah diagram yang menjelaskan secara detail bagaimana komponen disebar atau di deploy kedalam infrastruktur sebuah sistem.

Selain itu deployment diagram juga mempermudah user dalam menggunakan sistem yang telah dibangun. Seperti yang telah disebutkan di atas, salah satu dari fungsi deployment diagram adalah menggambarkan dan memvisualisasikan dan menspesifikasikan proses.

Sebagai contoh, ketika menspesifikasikan sebuah situs web, maka deployment diagram akan memperlihatkan perangkat keras yang digunakan atau dalam hal ini disebut dengan node, misalkan;

- Server aplikasi
- Web server
- Database server, dll.

Manfaat & Kegunaan Deployment Diagram

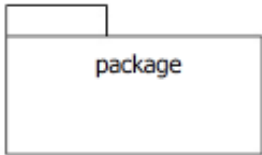
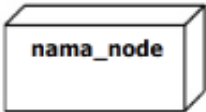
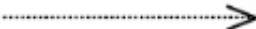

Manfaat dari deployment diagram:

1. Memberikan kemudahan dalam menjelaskan atau menerangkan sebuah sistem dibandingkan jika hanya dengan kata-kata.
2. Mempermudah untuk mengingat data-data kuantitatif.
3. Lebih efektif dan proaktif sebagai penjelasan dibandingkan dengan menggunakan kalimat.

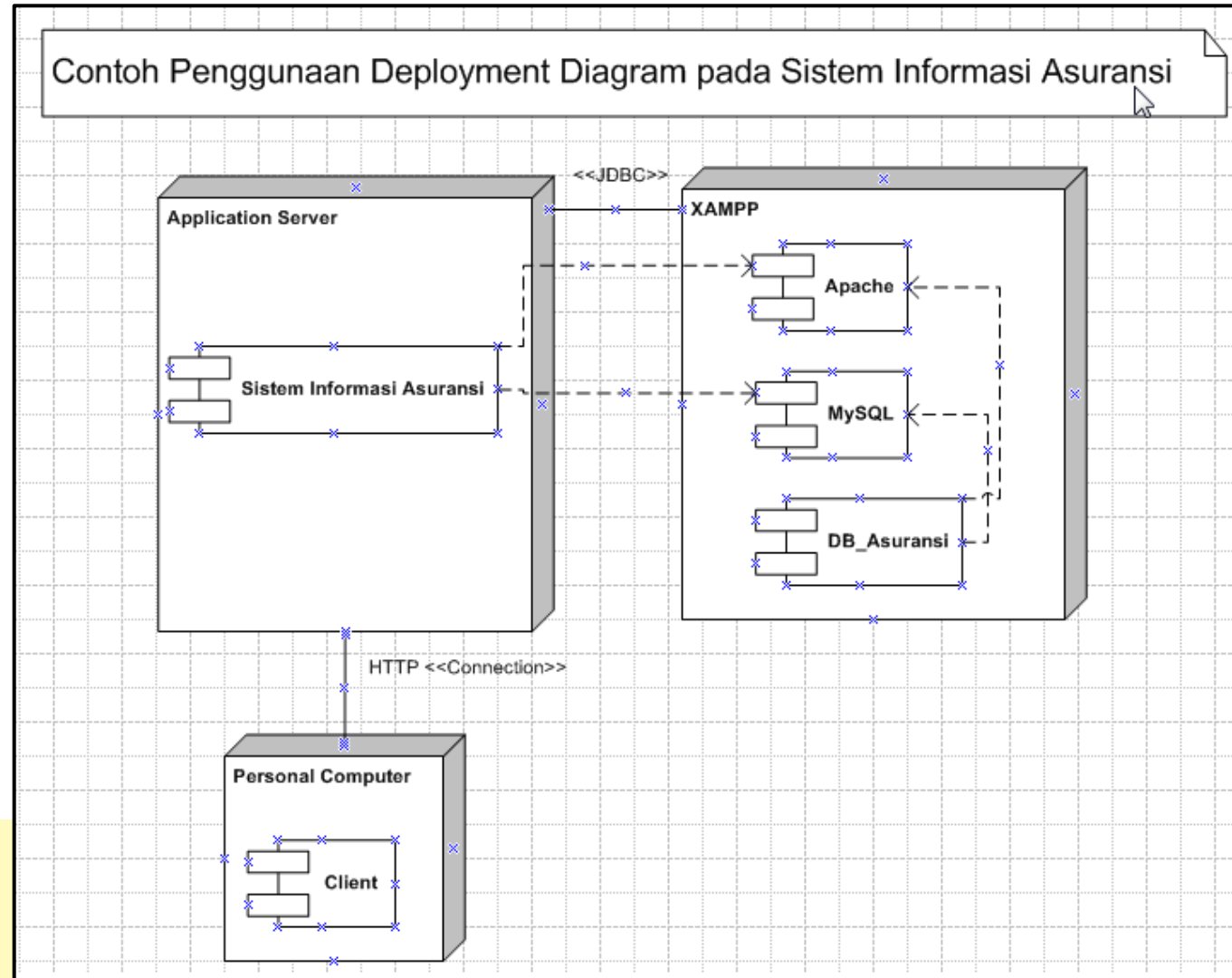
Kegunaan atau fungsi dari deployment diagram:

1. Menunjuk struktur dari sistem run-time.
2. Memberikan gambaran bagaimana perangkat keras memiliki keterkaitan antara satu dengan yang lainnya.
3. Menunjukkan perangkat keras dan perangkat lunak di install.

Simbol Deployment Diagram

Simbol	Deskripsi
<p>Package</p> 	package merupakan sebuah bungkusan dari satu atau lebih <i>node</i>
<p>Node</p> 	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
<p>Kebergantungan / <i>dependency</i></p> 	Kebergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai
<p>Link</p> 	relasi antar <i>node</i>

Contoh Deployment Diagram



#7 Component Diagram

Diagram component atau komponen diagram ialah sesuatu yang digunakan untuk memperlihatkan sebuah organisasi serta keterkaitan dan ketergantungan diantara kumpulan komponen dalam sebuah sistem.

Sedangkan pengertian component diagram berdasarkan UML adalah sebagai berikut.

“Hal fisik dari sistem yang dimodelkan saat sistem dieksekusi.”

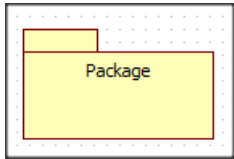
Kegunaan Component Diagram

- Sebagai source code perangkat lunak (software).
- Komponen executable yang dilepas dan digunakan oleh user.
- Bagian dari basis data secara fisik.
- Sistem yang harus peradaptasi dengan sistem yang lainnya.
- Sebagai framework sistem, dimana dalam hal ini memiliki tujuan untuk mempermudah perawatan dan pengelolaan dari sistem.

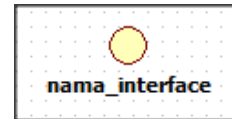
Contoh Component Diagram

- Komponen data yang digunakan untuk memanipulasi data.
- Komponen bussiness processing yang digunakan untuk menangani proses bisnis.
- Komponen security yang digunakan dalam menangani keamanan.
- Komponen user interface yang digunakan dalam menangani tampilan sebuah sistem.

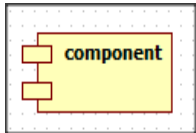
Simbol Component Diagram



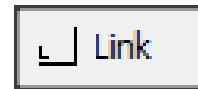
Package adalah sebuah simbol yang digunakan untuk mewadahi komponen.



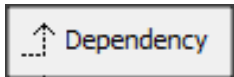
Hal ini mirip dengan pemrograman berorientasi objek, dimana simbol ini digunakan untuk antarmuka dengan tujuan supaya tidak langsung mengakses objek.



Component system merupakan simbol yang menggambarkan hardware atau objek dalam sistem tersebut.

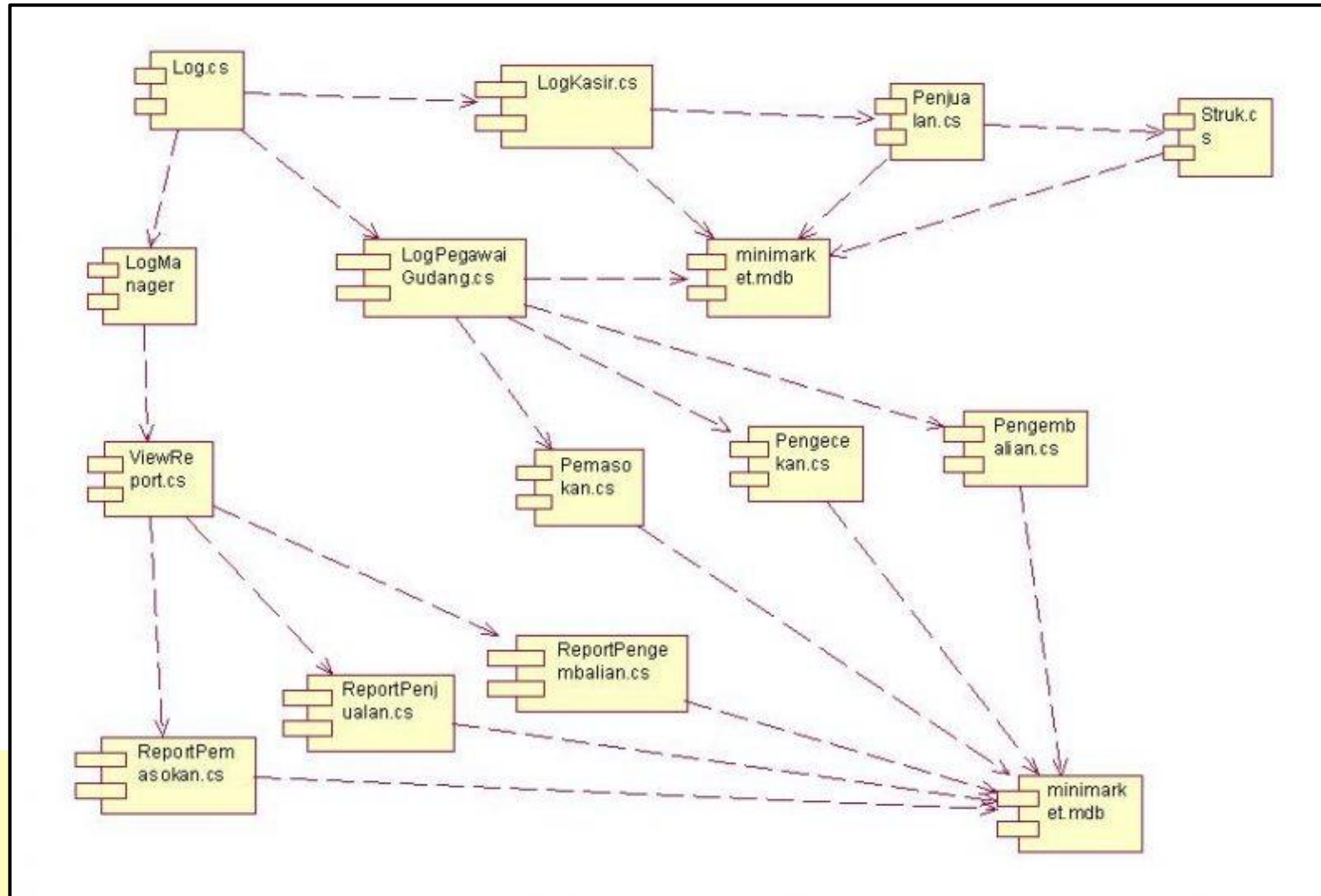


Simbol link ini digunakan untuk menunjukan relasi antar komponen, apabila suatu komponen memiliki relasi atau keterhubungan dengan komponen lainnya maka digunakan simbol link ini.



Simbol yang menggambarkan sebuah ketergantungan antar komponen, satu komponen dengan yang lainnya. Arah panah dalam simbol tersebut diarahkan pada komponen yang digunakan.

Contoh Component Diagram



Perbedaan Component Diagram & Deployment Diagram

- **Component Diagram** adalah diagram UML yang menampilkan komponen dalam system dan hubungan antara mereka. Saat berurusan dengan dokumentasi sistem yang kompleks, component diagram dapat membantu memecah sistem menjadi komponen yang lebih kecil.
- **Deployment Diagram** diagram yang digunakan untuk memvisualisasikan hubungan antara *software* dan *hardware*. Secara spesifik deployment diagram dapat membuat *physical model* tentang bagaimana komponen perangkat lunak (artefak) digunakan pada komponen perangkat keras, yang dikenal sebagai node.

Kesimpulan:

Component diagram menunjukkan bagaimana berbagai elemen sistem telah dikelompokkan bersama (menjadi rakitan) dan hubungan antara komponen-komponennya.

Deployment diagram lebih detail dimana diagram tersebut menjelaskan elemen perangkat keras mana yang berada.