

ARCHITECTURAL VIEW

3

- Architectural view adalah menentukan cara mengerti sebuah sistem.
 - Misalnya, gambar pada suatu diorganisasi dalam beberapa cara, misalnya untuk menandakan fungsional, struktural, perilaku dan lainnya dalam sebuah sistem manajemen proyek
- Unsur-unsur pembentuk dinamakan sebagai elemen
 - Contoh, semua elemen dalam gambar menandakan elemen dalam bentuk diagram tertentu
- UML adalah bahasa dan bukan metodologi, tidak menetapkan arsitektur secara eksplisit, namun diagram UML dapat diorganisir untuk view arsitektur

COMPANY

ARSITEKTUR LOGIKA # 1

4

- Fokus pada bentuk logika dari sistem menggunakan diagram class, dependency, visibilitas class dan sub sistem untuk mengkomunikasikan eksistensi dari class dan relasi antar class
- **Dependency**
 - Dependency dari satu elemen asal (disebut client) ke elemen target (disebut supplier) menandakan bahwa elemen asal menggunakan atau tergantung pada elemen target; jika elemen target berubah, elemen asal mungkin membutuhkan perubahan.
 - Dependency ditunjukkan sebagai garis putus-putus dari elemen asal ke elemen target
 - Dependency bisa juga ditandai menggunakan kata kunci (keyword); tetapi kata kunci biasanya dihilangkan karena berarti menjelaskan bagaimana dependency digunakan.
 - Juga dependency tidak mempunyai arah panah, tetapi berupa panah terbuka

COMPANY

ARSITEKTUR LOGIKA #2

5

- **Visibilitas Class**

- 1) Lingkup akses yang diijinkan ke anggota dari sebuah class
- 2) Diaplikasikan ke atribut dan operasi
- Visibilitas UML memetakan ke visibilitas OO:
 - 1) Lingkup privat : dalam sebuah class (-)
 - 2) Lingkup paket : dalam sebuah paket (~)
 - 3) Lingkup publik : dalam sebuah sistem (+)
 - 4) Lingkup Protected : dalam sebuah pohon pewarisan (#)

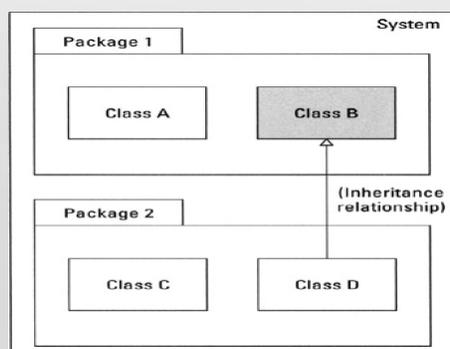
COMPANY

ARSITEKTUR LOGIKA #3

6

- **Visibilitas Privat**

- 1) Elemen privat hanya muncul dalam class tertentu
- 2) Notasinya "-"
- 3) Berguna dalam enkapsulasi



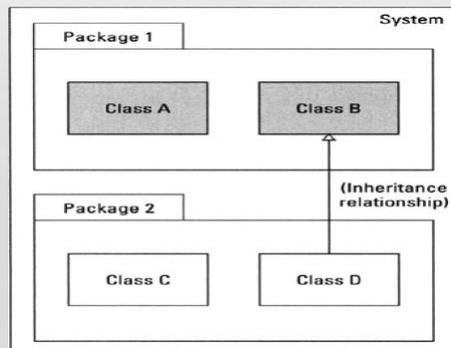
COMPANY

ARSITEKTUR LOGIKA #4

7

- **Visibilitas Paket (Package)**

- 1) Elemen paket visibel untuk elemen yang berada pada paket yang sama dengan nama tertentu.
- 2) Notasi : "~"



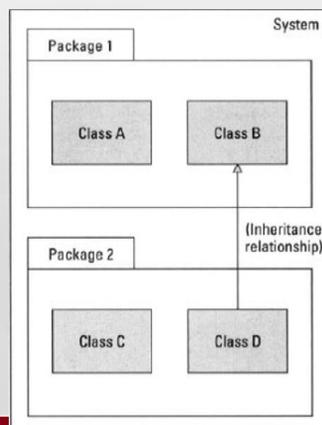
COMPANY

ARSITEKTUR LOGIKA #5

8

- **Visibilitas publik**

- 1) Elemen publik visibel untuk semua elemen yang dapat mengakses isinya
- 2) Notasi : "+"



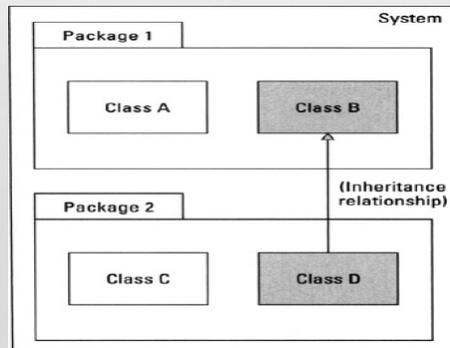
COMPANY

ARSITEKTUR LOGIKA #6

9

- **Visibilitas Protected**

- 1) Elemen protected visibel untuk elemen yang mempunyai hubungan generalisasi
- 2) Notasi : "#"



COMPANY

SUB SISTEM # 1

10

- Sistem adalah sebuah koleksi terorganisir dari elemen-elemen yang dapat didekomposisi menjadi subsistem yang lebih kecil secara rekursif dan akhirnya menjadi elemen-elemen primitif tidak dapat didekomposisi, misalnya, sistem manajemen proyek dapat didekomposisi menjadi :
 - Subsistem antar muka user bertanggung jawab untuk menyediakan sebuah antar muka dimana user dapat berinteraksi dengan sistem
 - Subsistem proses bisnis bertanggung jawab untuk implementasi fungsional bisnis
 - Subsistem data bertanggung jawab untuk implementasi fungsi penyimpanan data

COMPANY

SUB SISTEM #2

11

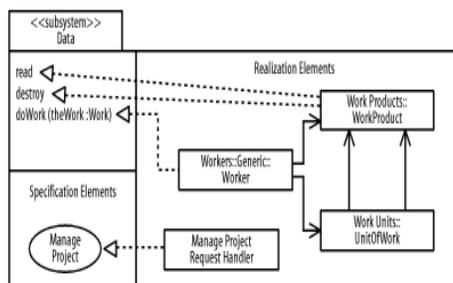
- Elemen primitif dapat berupa berbagai class yang digunakan dalam subsistem dan akhirnya di seluruh sistem.
- Paket (package) menyederhanakan pengelompokan elemen, subsistem mengelompokkan elemen yang menyediakan layanan sama dimana elemen lain boleh mengakses hanya layanan tersebut yang tidak ada pada elemen itu sendiri.
- Paket memungkinkan untuk membagi sistem ke dalam kelompok logika dan berhubungan dengan kelompok logika tersebut, subsistem memungkinkan untuk mempertimbangkan layanan dalam kelompok

COMPANY

SUB SISTEM #3

12

- Sebuah subsistem ditunjukkan sebagai paket (package) yang ditandai dengan kata kunci subsistem.
- Kotak persegi besar (package) yang mempunyai 3 bagian dengan membaginya menggunakan garis vertikal dan kemudian membagi area kiri menjadi 2 bagian dengan garis horisontal, contoh subsistem data pada sistem manajemen proyek, operasi subsistem, spesifikasi elemen dan antar muka menggambarkan layanan dari subsistem dan hanya layanan yang dapat diakses oleh elemen lain di luar subsistem



Bagian kiri atas adalah daftar operasi yang direalisasikan sistem. Bagian kiri bawah berlabel "Specification Elements" menunjukkan elemen yang direalisasikan subsistem. Bagian kanan berlabel "Realization Elements" menunjukkan elemen dalam subsistem yang merealisasikan operasi subsistem dan elemen spesifikasi seperti antarmuka user yang disediakan. Notasi umum tsb dapat dimodifikasi dengan mengubah bagian, mengkombinasikan bagian atau menekankan pada satu atau lebih bagian. Semua elemen dapat digunakan sebagai elemen spesifikasi atau realisasi, karena realisasi menandakan bahwa elemen realisasi mendukung semua operasi dari elemen spesifikasi tanpa perlu mendukung atribut atau asosiasi dari elemen spesifikasi

ARSITEKTUR HARDWARE #1

13

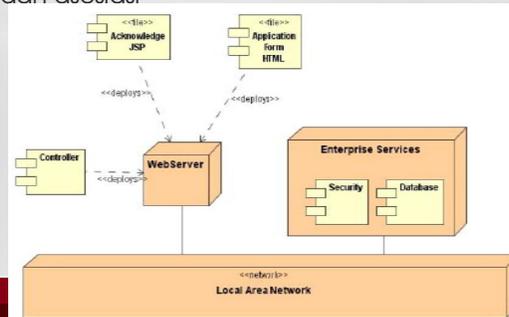
- Fokus pada lingkungan implementasi menggunakan diagram deployment untuk mengkomunikasikan bagaimana sistem diimplementasikan dalam lingkungan tsb.
- **Diagram Deployment**
 - Pemodelan deployment adalah jenis khusus pemodelan struktural yang fokus pada pemodelan lingkungan implementasi dari sistem
 - Berbeda dengan pemodelan komponen sistem, model deployment menunjukkan sumber daya eksternal yang dibutuhkan komponen
 - Pemodelan deployment diaplikasikan selama aktifitas desain untuk menentukan bagaimana aktifitas deployment menyebabkan sistem tersedia untuk user; sehingga untuk menentukan elemen dari sistem yang aktifitas deployment akan difokuskan
 - Seperti pemodelan komponen, pemodelan deployment biasanya mulai setelah desain sistem cukup lengkap, sebagaimana ditentukan oleh proses pengembangan sistem

COMPANY

ARSITEKTUR HARDWARE #2

14

- **Deployment**
 - 1) menunjukkan konfigurasi simpul2 pemrosesan saat berjalan dan komponen yang berada didalamnya
 - 2) Menggambarkan bentuk arsitektur deployment statis
 - 3) Dihubungkan dengan diagram komponen dimana simpul berada
 - 4) Fokus pada simpul2 sistem yang terdiri dari
 - a) Simpul (node)
 - b) Hubungan dependensi dan asosiasi
 - c) Komponen
 - d) Paket (package)



COMPANY

ARSITEKTUR HARDWARE #3

15

- **Simpul dan Komponen**
 - **Komponen**
 - 1) Berpartisipasi dalam eksekusi sistem
 - 2) Menyatakan paket fisik dari elemen logika
 - **Simpul (Node)**
 - 1) Mengeksekusi elemen
 - 2) Menyatakan deployment fisik dari komponen
 - Relasi deploy antara simpul dan komponen dapat ditunjukkan dengan relasi dependency
- **Mengorganisasikan Simpul**
 - Simpul dapat diorganisasi
 - 1) Sama dengan class dan komponen
 - 2) Dengan menentukan relasi dependensi, generalisasi, asosiasi, agregasi dan realisasi diantara mereka
 - Jenis relasi yang umum digunakan diantara simpul adalah asosiasi yang merepresentasikan hubungan fisik diantara mereka

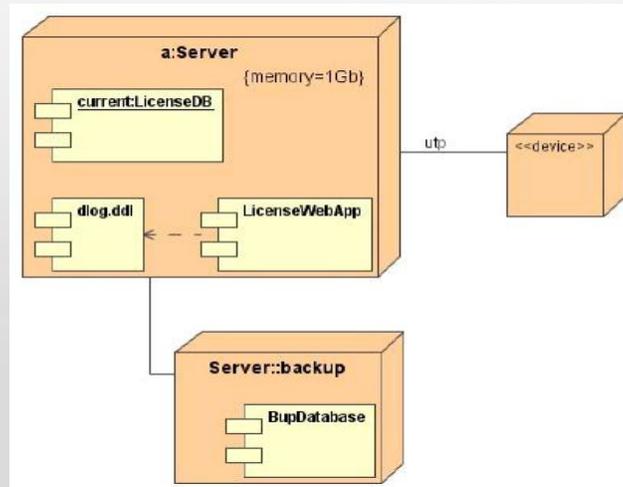
ARSITEKTUR HARDWARE #4

16

- **Processor dan Perangkat (Device)**
 - Processor adalah simpul yang mempunyai kemampuan memproses. Prosesor dapat mengeksekusi komponen
 - Perangkat adalah simpul yang tidak mempunyai kemampuan memproses (setidaknya ditunjukkan pada level abstraksi)
- **Prosedur pemodelan Simpul**
 - 1) Identifikasi elemen komputasi dalam deployment sistem dan memodelkannya sebagai simpul
 - 2) Tambahkan stereotype yang berhubungan ke simpul
 - 3) Tentukan atribut dan operasi yang dapat diaplikasikan ke setiap simpul
- **Distribusi Komponen**
 - Untuk memodelkan topologi sistem perlu menentukan distribusi fisik dari komponen yang melalui prosesor dan perangkat dari sistem
- **Prosedur**
 - 1) Alokasikan setiap komponen dalam simpul yang diberikan
 - 2) Pertimbangkan lokasi duplikat untuk komponen jika diperlukan
 - 3) Berikan alokasi dengan satu dari cara berikut
 - a) Jangan membuat alokasi visibel
 - b) Gunakan relasi dependency antar simpul dan komponen deploy
 - c) Uraikan komponen yang di-deploy pada simpul pada bagian tambahan

ARSITEKTUR HARDWARE #5

17



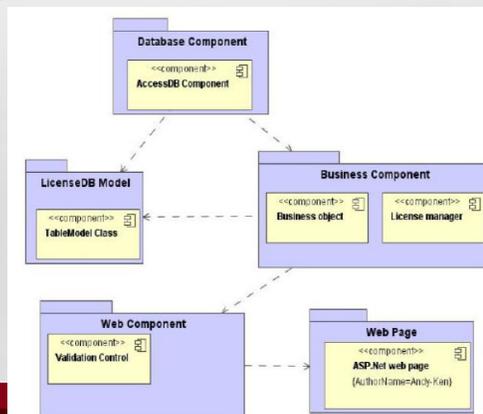
COMPANY

ARSITEKTUR IMPLEMENTASI #1

18

- Fokus pada implementasi sistem, menggunakan diagram komponen untuk berkomunikasi bagaimana sistem diimplementasikan
- **Diagram Komponen**
 - Komponen utamanya adalah

- 1) Komponen
- 2) Interface
- 3) Package



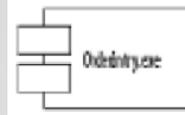
COMPANY

ARSITEKTUR IMPLEMENTASI #2

19

- **Komponen**

- Definisi : Komponen adalah bagian fisik yang dapat dipindah, yang menyatakan dan menyediakan realisasi dari sekumpulan antar muka (interface)
- Sebuah komponen:
 - 1) Enkapsulasi implementasi dari class yang berada di dalamnya
 - 2) Tidak mempunyai fitur sendiri, tetapi berfungsi sebagai wadah untuk elemennya
 - 3) Bagian yang dapat dipindah dan diganti dalam sistem
- Contoh komponen bernama OrderEntry.exe.



COMPANY

ARSITEKTUR IMPLEMENTASI #3

20

- **Definisi Interface (antar muka):**

- Antar muka adalah kumpulan operasi yang digunakan untuk menentukan layanan dari class atau komponen
- Interface
 - 1) Menyatakan bagian utama dari sistem
 - 2) Direalisasikan oleh komponen dalam implementasi
 - 3) Mempromosikan deployment dari sistem dimana layanan ditempatkan independen dan dapat dipindah
- Relasi antara komponen dan antarmuka digambarkan sbb
 - a) Interface ditampilkan dalam bentuk yang lebih besar
 - b) Komponen realisasi dihubungkan menggunakan relasi realisasi penuh



COM

ARSITEKTUR IMPLEMENTASI #4

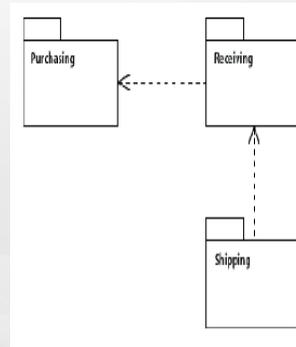
21

- **Paket (Package):**

- 1) Adalah mekanisme umum untuk mengorganisasi elemen yang dimodelkan dalam kelompok
- 2) Elemen kelompok secara semantik tertutup dan cenderung berubah bersama2

- **Notasi Package:**

- Digambarkan seperti folder
- Referensi package satu sama lain menggunakan notasi dependensi standar
- Contoh: package Purchasing tergantung pada packageReceiving
- Package dan dependensinya mungkin di-stereotype



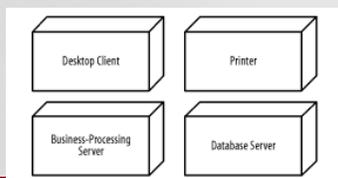
COMPANY

ARSITEKTUR IMPLEMENTASI #5

22

- **Simpul (Node)**

- Simpul adalah sumber daya yang tersedia selama waktu eksekusi. Secara tradisional, simpul merupakan komputer dalam suatu jaringan, tetapi dalam UML, simpul bisa berupa komputer, printer, server, internet atau sumber daya lain yang tersedia dalam komponen
- Contohnya sistem manajemen proyek, terdiri dari simpul berikut:
 - Desktop client: dimana komponen antar muka user dieksekusi
 - Printer: dimana sistem manajemen proyek menggunakannya untuk mencetak laporan
 - Business-processing server : dimana komponen business-processing component dieksekusi
 - Database server : dimana komponen data mengeksekusi dan dimana informasi yang berhubungan dengan proyek disimpan



COMPANY