

Suplemen Materi

Class Diagram



Dr. Sutedi, S.Kom., M.T.I.

Class Diagram

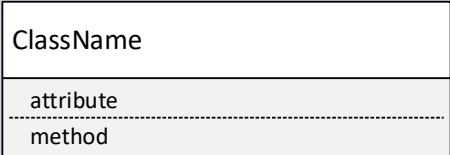
- ❖ *Class diagram* menggambarkan struktur *system* dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun *system*.
- ❖ *Class diagram* mirip *ER-Diagram* pada perancangan *database*, bedanya pada *ER-diagram* tidak terdapat operasi/metode tapi hanya atribut. *Class diagram* terdiri dari nama kelas, atribut dan operasi/metode.
- ❖ *Class diagram* mampu memberikan kita pandangan yang lebih luas mengenai suatu *system* dengan cara menunjukkan kelas serta hubungan-hubungannya.

Class Diagram

- ❖ *Class diagram* menawarkan banyak manfaat diantaranya:
 - Menggambarkan suatu model data untuk *system* informasi, tidak peduli apakah model data tersebut sederhana maupun kompleks.
 - Dengan mempelajari *class diagram* maka akan meningkatkan pemahaman mengenai gambaran umum skema dari suatu aplikasi.
 - Mampu menyatakan secara visual kebutuhan spesifik suatu informasi.
 - Dapat menghasilkan bagan jelas dan terperinci serta memperhatikan hal-hal yang dibutuhkan suatu program agar dapat diimplementasikan sesuai dengan apa yang dimodelkan.

Class Diagram

- ❖ Karakteristik dasar *class diagram* terdiri dari komponen-komponen berikut.

Simbol	Keterangan
 <p>The diagram shows a rectangular box representing a class. The top section is labeled 'ClassName'. Below it, a horizontal line separates the 'attribute' section from the 'method' section. The word 'Class' is written to the right of the box.</p>	<p><i>Class</i> merupakan sesuatu yang membungkus informasi tentang suatu obyek dan perilakunya.</p> <p>Hal tersebut merupakan salah satu perbedaan pendekatan terstruktur dengan pendekatan berorientasi obyek, dimana pendekatan berorientasi obyek menggabungkan informasi dan perilaku pengolahnya ke dalam suatu <i>class</i>, sedangkan pendekatan terstruktur menyimpan informasi pada sisi data dan perilaku pengolahnya pada sisi aplikasi.</p>

Class Diagram

Aturan penulisan *class* adalah sebagai berikut.

- ***Class name*** merupakan sebutan yang mewakili sebuah *class*. Pada konsep relasional, *class* mirip seperti *entity* atau *table*.
- ***Attribute*** merupakan *property* dari sebuah *class* yang menjadi ciri-ciri dari *class* tersebut. Penulisan nama *attribute* diawali dengan huruf kecil. Setelah nama *attribute* dilanjutkan dengan titik dua (:), dan diikuti dengan *type data*.
- ***Method*** merupakan sesuatu yang dapat dilakukan atau diproses pada sebuah *class*. Penulisan *method* diawali huruf kecil dan diikuti dengan tanda kurung () untuk menandakan fungsi serta dapat juga ditambahkan *variable* dan *type data* jika diperlukan.


Class Diagram

Aturan penulisan *class* adalah sebagai berikut (lanjutan).

- Penulisan nama **attribute** dan **methode** diawali dengan *visibility* (-,+,#,~) yang menentukan *accessibility* untuk suatu *attribute* atau *method*.
 - Tanda *minus* (-) artinya *private*, dimana *attribute* atau *method* hanya dapat digunakan oleh *class* itu sendiri dan tidak dapat diakses oleh *class* atau *subclass* lainnya.
 - Tanda *plus* (+) artinya *public*, dimana *attribute* atau *method* dapat diakses oleh *class* atau *subclass* manapun yang memerlukan.
 - Tanda *Hashtag* (#) artinya *protected*, dimana *attribute* atau *method* dilindungi dan hanya dapat diakses oleh *class* tersebut dan *subclass* yang merupakan turunannya.
 - Tanda *Tilde* ~ artinya *package*, dimana *attribute* dan *method* dapat digunakan oleh *class* atau *sub class* yang berada dalam satu paket yang sama.

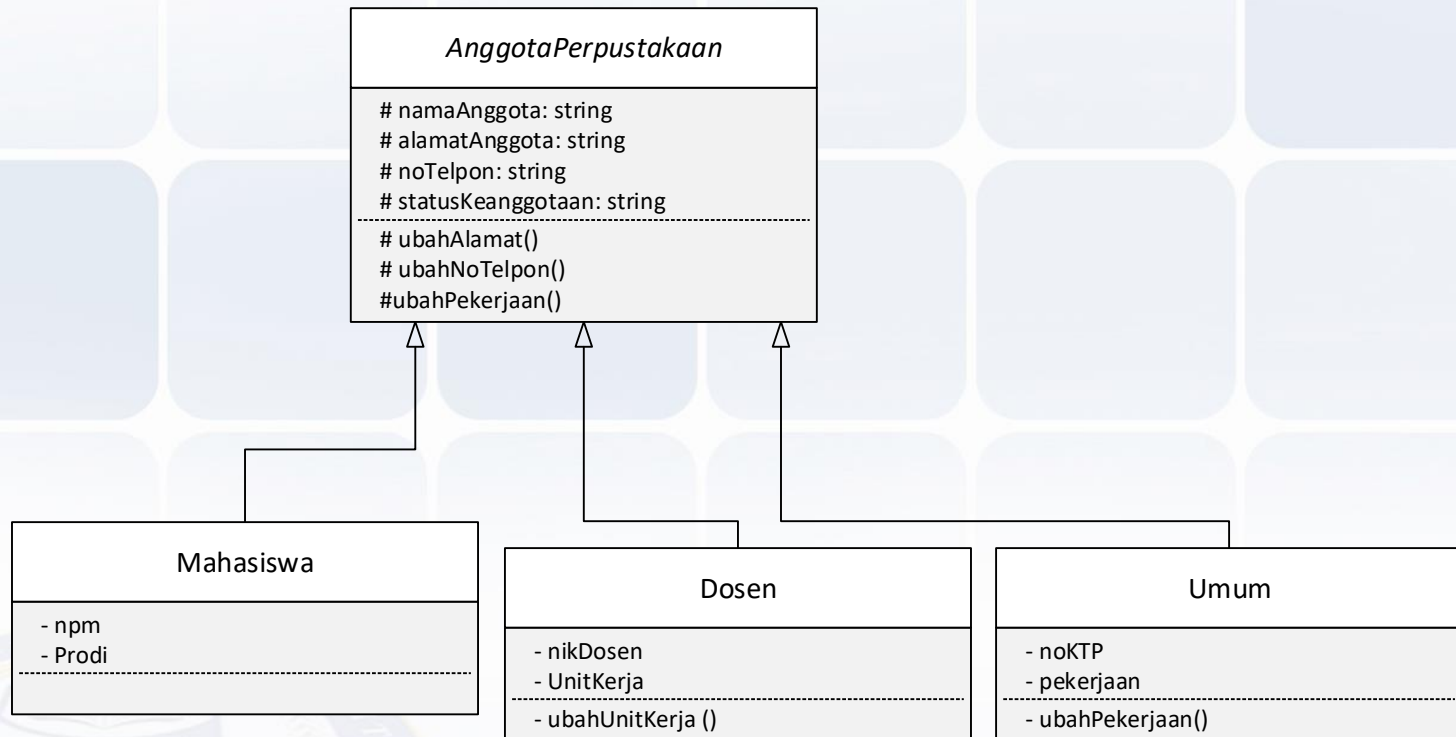
Class Diagram

- ❖ Karakteristik dasar *class diagram* terdiri dari komponen-komponen berikut (lanjutan)..

Simbol	Keterangan
 <p data-bbox="658 586 880 625"><i>Inheritance</i></p>	<p data-bbox="987 529 1779 972"><i>Inheritance</i> menunjukkan relasi antar kelas dengan makna generalisasi-spesialisasi (Umum-khusus), dimana <i>class</i> khusus mewarisi seluruh <i>attribute</i> dan <i>method</i> yang ada pada <i>class</i> umumnya. Dalam penggambarannya, mata panah mengarah dari <i>class</i> khusus menuju <i>class</i> yang lebih umum.</p> <p data-bbox="987 1043 1779 1200">Dalam konsep relasional, penerapan <i>inheritance</i> sama seperti konsep <i>supertype-subtype</i>.</p>

Class Diagram

Contoh penerapan *inheritance*.




Catatan:

- *Class* umum seringkali disebut juga dengan istilah *superclass* atau *parent class* atau *abstraction class*, sedangkan class khusus disebut *subclasses* atau *child classes*.
- Untuk mencirikan *abstraction class* maka penamaannya dicetak miring (*italic*) atau jika digambar manual diberi tanda <<*abstraction*>>

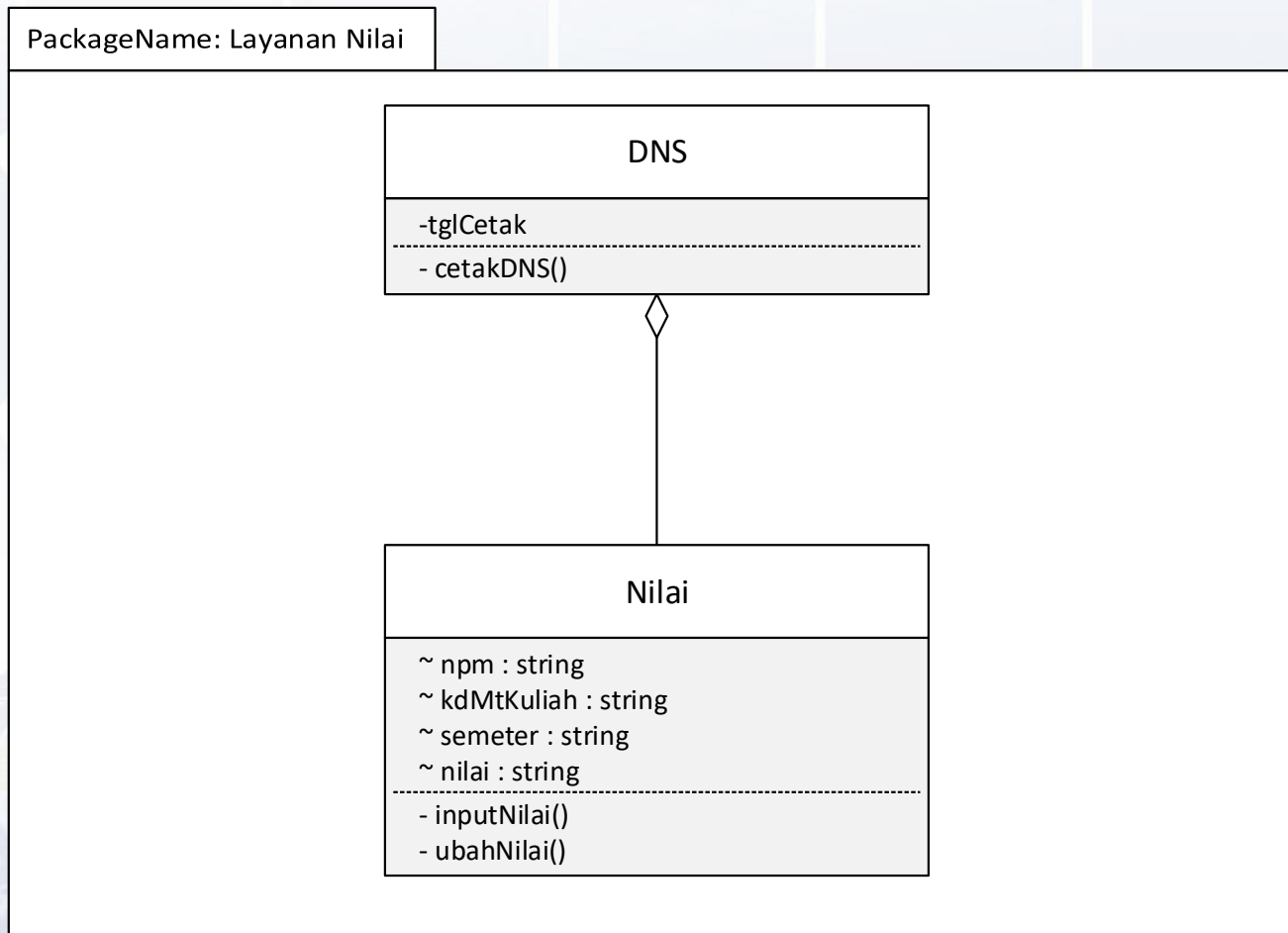
Class Diagram

- ❖ Karakteristik dasar *class diagram* terdiri dari komponen-komponen berikut (lanjutan)..

Simbol	Keterangan
 <i>Aggregation</i>	<p><i>Aggregation</i> merupakan hubungan yang menyatakan bagian (<i>whole-part</i>). Pada hubungan ini <i>class</i> yang merupakan bagian dari <i>class</i> keseluruhan dapat berdiri sendiri meskipun <i>class</i> keseluruhannya tidak ada. Dalam penggambarannya, tanda belah ketupat (simbol <i>diamond</i>) diletakkan pada <i>class</i> gabungan (<i>whole class</i>) dan <i>class</i> bagiannya (<i>part class</i>) pada sisi lainnya.</p> <p>Pada konsep relasional, <i>part class</i> merupakan <i>strong entity</i>.</p>

Class Diagram

Contoh penerapan *Aggregation*.




Catatan:

Pada contoh tersebut ditampilkan juga pengaplikasian *package*, dimana *class* DNS dan *class* Nilai berada dalam satu *package* Layanan Nilai.

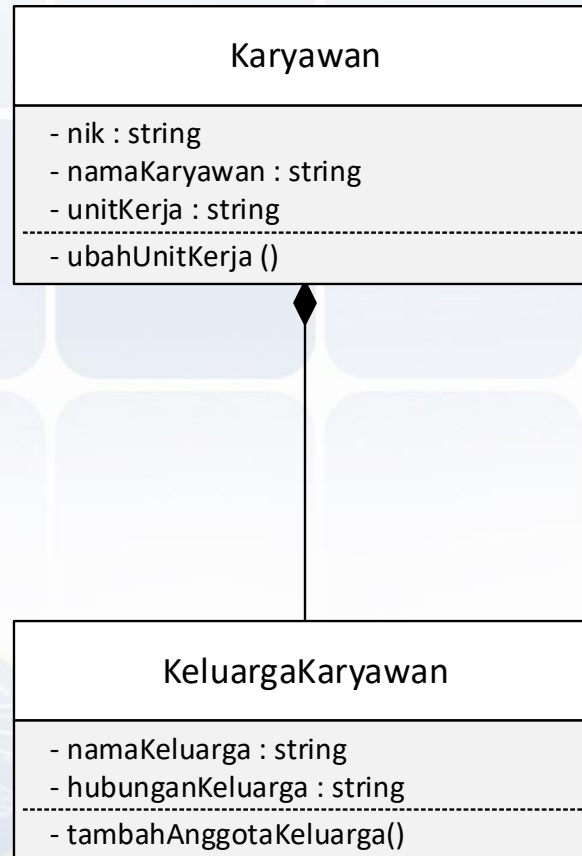
Class Diagram

- ❖ Karakteristik dasar *class diagram* terdiri dari komponen-komponen berikut (lanjutan)..

Simbol	Keterangan
 <i>Composition</i>	<p><i>Composition</i> merupakan <i>strong association</i> atau <i>death relationship</i>, dimana <i>part class</i> tidak dapat <i>exist</i> jika <i>whole class</i>-nya dihapus. <i>Composition</i> digambarkan dengan belah ketupat (simbol <i>diamond</i>) hitam solid, dimana tanda tersebut diletakkan pada <i>whole class</i> dan <i>part class</i>-nya pada sisi yang lain.</p> <p>Dalam konsep relasional, <i>part class</i> merupakan <i>weak entity</i>.</p>

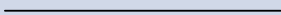
Class Diagram

Contoh penerapan *Composition*.



Class Diagram

- ❖ Karakteristik dasar *class diagram* terdiri dari komponen-komponen berikut (lanjutan)..

Simbol	Keterangan
 <i>Association</i>	<p><i>Association</i> merupakan hubungan yang sifatnya statis di antara <i>class</i>. Biasanya memperlihatkan hubungan antar <i>class</i> yang bersifat umum.</p> <p>Dalam konsep relasional, hubungan ini sama dengan <i>binary relationship</i> antar <i>table</i>.</p> <p><i>Association</i> biasanya dilengkapi dengan <i>multiplicity</i> atau dalam relasional biasa dikenal dengan derajat kardinalitas.</p>

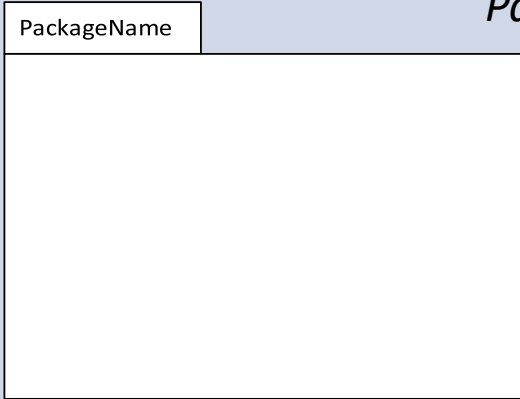
Class Diagram

Penjelasan *multiplicity*:

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

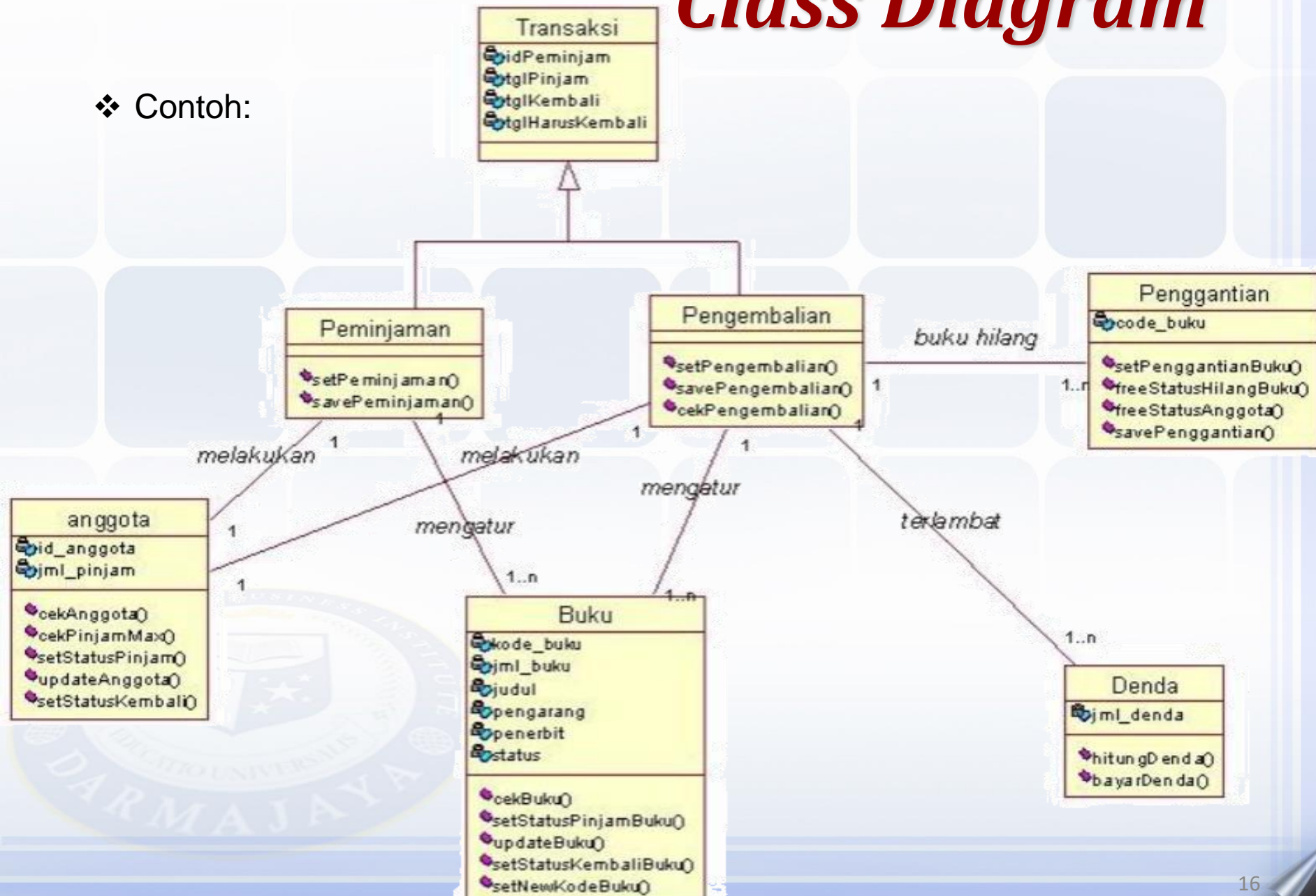
Class Diagram

- ❖ Karakteristik dasar *class diagram* terdiri dari komponen-komponen berikut (lanjutan)..

Simbol	Keterangan
 <p>The diagram shows a UML Package symbol. It consists of a rectangular box with a smaller rectangle attached to its top-left corner. The smaller rectangle contains the text 'PackageName'. To the right of the main box, the word 'Package' is written in an italicized font.</p>	<p><i>Package diagram</i> dapat diasumsikan seperti halnya <i>folder</i> yang terdapat dalam <i>operating system</i> seperti <i>windows</i> maupun <i>linux</i>.</p> <p><i>Package</i> adalah sekumpulan <i>class</i> atau <i>element</i> dengan sifat yang sama.</p> <p><i>Package</i> dapat diimplementasikan untuk suatu <i>modul</i> atau <i>subsystem</i> tertentu</p>

Class Diagram

❖ Contoh:



THANK YOU

Sampai jumpa di sesi berikutnya



PresentaMedia 