|  |  |  |
| --- | --- | --- |
| **BAHAGIAN PENDIDIKAN DAN LATIHAN TEKNIK VOKASIONAL**  **KEMENTERIAN PENDIDIKAN MALAYSIA**  **ARAS 5 & 6, BLOK E14, KOMPLEKS E,**  **PUSAT PENTADBIRAN KERAJAAN PERSEKUTUAN**  **KERTAS PENERANGAN**  ***(INFORMATION SHEET)***  C:\Users\norsu\Desktop\jata negara.png | | |
| **KOD DAN NAMA PROGRAM NOSS** | IT-010-3:2016 PEMBANGUNAN APLIKASI | |
| **TAHAP NOSS** | 3 | |
| **KOD, NAMA CU DAN WA NOSS** | **CU02/WA1 – INTERPRET APPLICATION MODULE DEVELOPMENT REQUIREMENT** | |
| **NAMA PROGRAM KV** | SISTEM PENGURUSAN PANGKALAN DATA DAN APLIKASI WEB | |
| **KOD DAN NAMA KURSUS KSKV** | KPD 1013 INTRODUCTION TO APPLICATION SYSTEM DEVELOPMENT | |
| **NO DAN TAJUK STANDARD KANDUNGAN KSKV** | K2 REPRESENT THE PROCESS USING UNIFIED MODELLING LANGUAGE | |
| **NO. KOD NOSS** | IT-010-3:2016-C02/P(15/70) | Muka Surat : 1 Drp : 7 |
| **NO. KOD KSKV** | KPD1013/P(7/17) |

**TAJUK/***TITLE***:**

**PENGENALAN KEPADA UML BAGI CLASS DIAGRAM**

**TUJUAN/***PURPOSE* **:**

Pelatih – pelatih mesti boleh :

1. Terangkan konsep Class Diagram
2. Senaraikan struktur bagi Class Diagram
3. Terangkan contoh *Class Diagram* berdasarkan situasi

**PENERANGAN/***INFORMATION* **:**

**1.0 KONSEP CLASS DIAGRAM**

* Class diagram adalah ***decryption* (penyahsulitan) kumpulan objek-objek dengan *property*, tingkah laku (operasi) dan hubungan** yang sama.
* Class diagram dapat memberikan pandangan global terhadap sebuah sistem. Hal tersebut dipaparkan melalui *class - class* yang ada dan hubungannya dengan yang lain.
* Sebuah sistem biasanya mempunyai beberapa class diagram.
* Class diagram **sangat membantu dalam visualisasi struktur kelas dari sesebuah sistem.**
* Bersifat statik.
* Berfungsi untuk **menjelaskan jenis dari objek sistem dan hubungannya dengan objek yang lain.**
* *Object* adalah nilai tertentu dari setiap atribut kelas entiti.

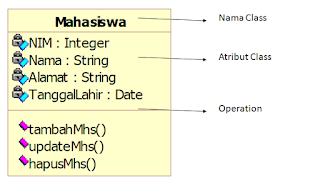
**2.0 STRUKTUR BAGI CLASS DIAGRAM**

Perkara utama dalam class diagram:

a. Nama 🡪 berfungsi memberi **identiti** kepada class

b. Atribut 🡪 berfungsi memberi **karakteristik** kepada data yang dimiliki pada suatu objek dalam kelas

c. Operasi 🡪 **kefungsian** ke pada sebuah objek



Rajah 1: contoh sebuah class diagram

Atribut dan method(kaedah) dapat memiliki salah satu sifat berikut:

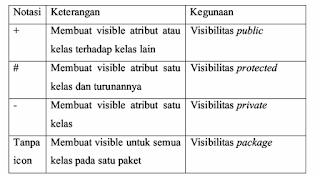
1. *Private:* tidak dapat dipanggil/ digunakan dari luar class yang bersangkutan

*2.Protected:* hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak (*child)* yang mewarisinya

3.*Public:*dapat dipanggil oleh sesiapa sahaja.

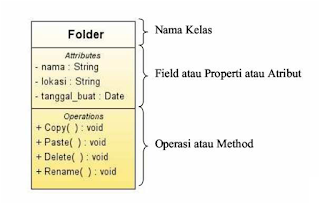
**Modifier**

Digunakan untuk **memberikan batasan atau penanda** bagaimana kelas lain menggunakan atribut, constructor dan method dari suatu kelas.



Rajah 2: Penerangan modifier pada sesuatu class

Contoh penggunaan Modifier:



Rajah 3: contoh penggunaan

Method digunakan untuk menyimpan tingkahlaku sistem pada bahasa pengaturcaraan. Method yang digunakan adalah:

1. Mengembalikan nilai (*non void method)*

*2.*Tidak mengembalikan nilai (*void method)*

contoh: getNama, getAlamat,getUsia,jalan

Struktur untuk penulisan modifier:

|  |
| --- |
| Modifier nama\_method ([namaParameter:jenisParameter]) : nilai\_kembali |

contoh penggunaan:

* **+ getNama() : String**

getNama memiliki modifier jenis public, tidak memiliki parameter dan memiliki nilai kembalian jenis string

* **+ setNama(nama : String) : void**

setNama memiliki modifier jenis public, mempunyai 1 parameter jenis string dan tidak mempunyai nilai kembalian

**Hubungan dalam objek.**

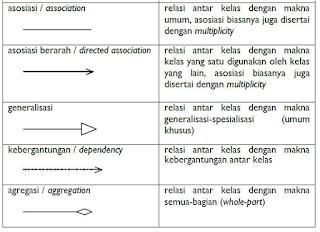
Semua system terdiri daripada class-class dan objek.

kelakuan system dicapai melalui kerjasama antara objek.

contoh: seorang mahasiswa ditambah ke dalam daftar kelas, jika daftar kelas memperoleh *mesej* untuk menambah mahasiswa.

Interaksi antara objek ini disebut *object relationship.*

Dua jenis hubungan yang ditemukan semasa analisis adalah *association* dan *aggregation.*

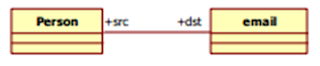


**Association**

Dikenali juga dengan hubungan “..has a.. “ atau MEMILIKI.

Digambarkan sebagai garis lurus antara dua kelas.

Tidak bermaksud satu kelas memiliki atau di miliki oleh kelas yang lain, tetapi kelas lain dapat berhubung dengan kelas yang sama.



**Directed Association**

Hubungan seperti *association* namun menggambarkan objek atau aliran kejadian berasal dari salah satu kelas, dan kelas lain juga ada yang bersifat pasif.

https://3.bp.blogspot.com/-586Lvl-1xto/WDVNaiQ0VvI/AAAAAAAAADE/op9I2G2MYWckgu-z_r8gbsuy5iV9tIl6ACLcB/s320/Untitled.png

**Aggregation**

Dikenali juga dengan hubungan “..own as…” atau terdiri atas..

Terdapat kelas yang berfungsi sebagai sebahagian kelas daripada kelas keseluruhan.

Namun, jika kelas asal keseluruhan tidak ada, kelas sebahagian itu tetap boleh berdiri dengan sendiri.

https://1.bp.blogspot.com/-WL5657uBV9w/WDVN6uMLF4I/AAAAAAAAADI/uie14Cl5b5QYtFy0Aoalg7EMHD9LzejVACLcB/s320/Untitled.png

**Composition**

Disebut juga sebagai *strong aggregation*. (“..is part of.. “sebahagian dari..”)

Seperti hubungan agregasi, jika kelas keseluruhan hilang atau tidak ada, maka sebahagian kelas juga tidak ada (mustahil untuk ia wujud)

https://3.bp.blogspot.com/-A-Wrsm-th3k/WDVOtPqH7kI/AAAAAAAAADU/5k_9fYojHCEjlugpYmPsfLNflaClyN6mwCEw/s320/Untitled.png

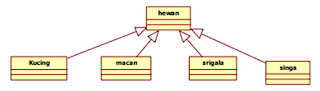
\*apabila kereta tiada, maka tiada juga tangki kereta

**Generalization**

Dikenali juga dengan “..is a…”

digunakan untuk menggambarkan perwarisan. Suatu kelas (*child class)* dapat diturunkan kepada kelas lain dan mewarisi semua atribut dan method induknya (*parent class)* dan dapat juga menambah method baru atau atribut baru.

Sebagai contoh kucing, serigala dan singa adalah haiwan. Maka kita boleh buat generalisasi bahawa hubungan antara kelas adalah seperti berikut:



**Dependency**

Hubungan kebergantungan antara kelas. Satu kelas memiliki kebergantungan terhadap kelas lain, tetapi kebergantungan tersebut adalah satu hala sahaja.

Perubahan pada kelas asal akan memberi efek kepada kelas yang bergantungan.

https://3.bp.blogspot.com/-oAlGrz68tiI/WDVQRxxXfGI/AAAAAAAAADc/BlXBdNdK_1898WqHOD4j7hBCeiuYoT1tACLcB/s320/Untitled.png

**Realization**

Sebuah hubungan antara dua buah kelas yang mengharuskan kelas sekunder mengikut aturan dari kelas utama.

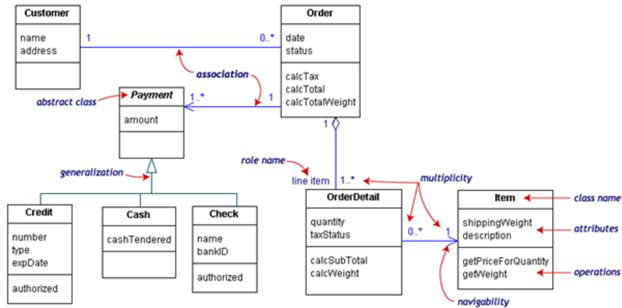
Ini biasa terjadi pada antaramuka (*interface*)



**Kelemahan:**

1. Sukar untuk penentuan antara sifat-sifat atau kelas
2. Pelaksanaan struktur data sukar dilakukan

**3.0 CONTOH *CLASS DIAGRAM* BERDASARKAN SITUASI**

****

*Figure: A customer order from a retail catalog*

**SOALAN/***QUESTION* **:**

1. Terangkan maksud UML (Unified Modeling Language )
2. Nyatakan perkara utama dalam Class Diagram.
3. Lakarkan gambarajah bagi Class Diagram

**RUJUKAN/***REFERENCE* **:**

1. *Unified Modeling Language.* (2014, December 22). Retrieved from http://sigitprabowoo.blogspot.my/2013/11/jenis-jenis-diagram-uml.html
2. *Pengertian UML Dan Jenis-Jenisnya Serta Contoh Diagramnya.* (2015, September 24). Retrieved from

http://www.pengertianku.net/2015/09/pengertian-uml-dan-jenis-jenisnya-serta-contoh-diagramnya.html

1. *Tugas Kuliah.* (2013, April 30). Retrieved from

http://tugas-kuliah-stmik.blogspot.my/2013/04/statechart-diagram-uml.html

1. *Free Sequence Diagram Tool*. (2004). Retrieved from

https://www.slideshare.net/mohammadkarim3785/employee-management-system-uml-diagrams

1. *UML Diagram @ Software engineering discussion.* ( 2014, August 30).

Retrieved from

https://www.slideshare.net/CherryBerry2/uml-diagram-software-engineering-discussion