

BAB II

TINJAUAN PUSTAKA

II.1 Penelitian Terdahulu

Penelitian yang sedang dilakukan juga mengambil beberapa referensi dari penelitian terdahulu. Adapun penelitian terdahulu tersebut dapat dilihat dalam tabel II.1 sebagai berikut:

Tabel II.1. Penelitian Terdahulu

1	Nama	Hartatik
	Tahun	Seminar Nasional Informatika 2014
	Judul	Pengelompokan Mahasiswa Berdasarkan Nilai Ujian Nasional Dan IPK Menggunakan Metode K-Means
	Hasil	Metode K-Means dapat digunakan untuk melakukan pengelompokan atau peng-cluster-an mahasiswa berdasarkan dua parameter yaitu IPK dan nilai UN mahasiswa tersebut ketika lulus SMU. Namun berdasarkan hasil yang didapatkan dengan sample 40 orang mahasiswa belum didapatkan nilai cluster yang bisa mewakili pengelompokan mahasiswa STMIK Amikom Yogyakarta secara keseluruhan.
2	Nama	Randi Rian Putra ¹ , Cendra Wadisman ²
	Tahun	<i>Journal of Information Technology and Computer Science (INTECOMS)</i> Volume 1 No 1, Maret 2018
	Judul	Implementasi Data Mining Pemilihan Pelanggan Potensial Menggunakan Algoritma <i>K-Means</i>
	Hasil	Data pelanggan yang potensial didapatkan setelah perhitungan algoritma K-Means selesai, data dengan pusat <i>centroid</i> terbesar lah yang termasuk ke dalam pelanggan yang paling potensial.
3	Nama	Windha Mega Pradnya Duhita
	Tahun	<i>Jurnal Informatika, Vol. 15, No. 2, Bulan Desember 2015</i>

	Judul	Clustering Menggunakan Metode K-Means Untuk Menentukan Status Gizi Balita
	Hasil	Kesimpulan dari penelitian ini adalah dengan membandingkan hasil pengelompokan menggunakan tabel Growth Chart dan algoritma K-Means didapat 17 data yang memiliki kelompok yang sama. Dari angka ini dapat disimpulkan bahwa algoritma K-Means hanya memiliki nilai akurasi 34% benar. Nilai ini bisa berubah seiring dengan penambahan data latih.
4	Nama	Anindya Khrisna Wardhani
	Tahun	JURNAL TRANSFORMATIKA, Volume 14, Nomor 1, Juli 2016
	Judul	Implementasi Algoritma K-Means Untuk Pengelompokan Penyakit Pasien Pada Puskesmas Kajen Pekalongan
	Hasil	Inisialisasi jumlah <i>cluster</i> sebanyak 2 buah sesuai dengan pendefinisian nilai k dengan jumlah <i>cluster</i> akut ada 376 item, <i>cluster</i> tidak akut ada 624 item dengan total jumlah data adalah 1000.
5	Nama	Roni Setiawan
	Tahun	JURNAL LENTERA ICT Vol.3 No.1, Mei 2016
	Judul	Penerapan <i>Data Mining</i> Menggunakan Algoritma <i>K-Means Clustering</i> Untuk Menentukan Strategi Promosi Mahasiswa Baru (Studi Kasus : Politeknik LP3I Jakarta)
	Hasil	Profil yang dihasilkan dari <i>data mining</i> dengan algoritma K-Means <i>Clustering</i> dapat memberikan pengetahuan untuk menentukan strategi promosi di Politeknik LP3I Jakarta. Pengetahuan yang didapat dari hasil <i>clustering</i> yaitu calon mahasiswa didominasi dari masyarakat ekonomi rendah dan menengah

Penelitian yang sedang dilakukan akan melakukan proses *K-Means Clustering* untuk mengolah data penduduk kurang mampu pada Desa Klambir V. Hasil dari penelitian ini yaitu berupa aplikasi yang dapat digunakan untuk mengolah data penduduk kurang mampu di Desa Klambir V, sehingga dari hasil

perhitungan metode dapat diketahui mana wilayah yang paling banyak membutuhkan bantuan dengan segera.

II.2 Data Mining

Data *mining* adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai database besar (Agus Nursikuwagus dan Tono Hartono : 2016 : 702).

II.1.1 Karakteristik Data Mining

Karakteristik data *mining* adalah (Heroe Santono, dkk : 2016):

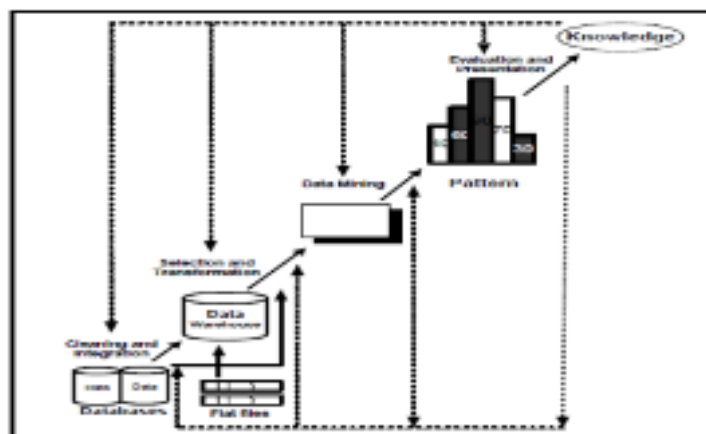
1. *Data mining* berhubungan dengan penemuan sesuatu yang tersembunyi dan pola data tertentu yang tidak diketahui sebelumnya.
2. *Data mining* biasa menggunakan data yang sangat besar.
3. *Data mining* berguna untuk membuat keputusan yang kritis, terutama dalam strategi.

II.1.2 Tahap - Tahap Data Mining

Tahap - tahap data *mining* adalah sebagai berikut (Heroe Santono, dkk : 2016 : 20):

1. Pembersihan data (*data cleaning*). Merupakan proses menghilangkan noise dan data yang tidak konsisten atau data tidak relevan.

2. Integrasi data (*data integration*). Merupakan penggabungan data dari berbagai *database* ke dalam satu *database* baru.
3. Seleksi Data (*Data Selection*). Data yang ada pada *database* sering kali tidak semuanya dipakai, oleh karena itu hanya data yang sesuai untuk dianalisis yang akan diambil dari *database*.
4. *Data Transformation*. Data diubah atau digabung ke dalam format yang sesuai untuk diproses dalam *data mining*.
5. Proses *mining*. Merupakan suatu proses utama saat metode diterapkan untuk menemukan pengetahuan berharga dan tersembunyi dari data.
6. Evaluasi pola (*pattern evaluation*). Untuk mengidentifikasi pola - pola menarik ke dalam *knowledge based* yang ditemukan bermanfaat.
7. Presentasi pengetahuan. Merupakan visualisasi dan penyajian pengetahuan mengenai metode yang digunakan untuk memperoleh pengetahuan yang diperoleh pengguna. Dalam presentasi ini, visualisasi juga bisa membantu mengkomunikasikan hasil *data mining*.



Gambar II.1. Tahap - Tahap Data Mining

II.3 Metode *K-Means Clustering*

K-Means merupakan algoritma *clustering* yang berulang - ulang. Algoritma *K-Means* dimulai dengan pemilihan secara acak K , K disini merupakan banyaknya cluster yang ingin dibentuk (Putra, Cendra: 2018 : 87).

Adapun urutan algoritma dari metode *K-Means Clustering* adalah sebagai berikut:

1. Tentukan k sebagai jumlah cluster yang di bentuk. Untuk menentukan banyaknya *cluster* k dilakukan dengan beberapa pertimbangan seperti pertimbangan teoritis dan konseptual yang mungkin diusulkan untuk menentukan berapa banyak *cluster*.
2. Bangkitkan k *Centroid* (titik pusat *cluster*) awal secara random. Penentuan centroid awal dilakukan secara random/acak dari objek - objek yang tersedia sebanyak k *cluster*, kemudian untuk menghitung centroid *cluster* ke- i berikutnya,

Menggunakan rumus sebagai berikut :

$$v = \frac{\sum_{i=1}^n x_i}{n} ; i = 1, 2, 3, \dots, n \dots\dots\dots(1)$$

dimana;

v : centroid pada *cluster*

x_i : objek ke- i

n : banyaknya objek/jumlah objek yang menjadi anggota *cluster*

3. Hitung jarak setiap objek ke masing - masing centroid dari masing - masing cluster. Untuk menghitung jarak antara objek dengan centroid penulis menggunakan *Euclidian Distance*.

$$d(x,y) = |x - y| = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} ; i = 1, 2, 3, \dots, n \dots\dots\dots(2)$$

dimana;

x_i : objek x ke-i

y_i : daya y ke-i

n : banyaknya objek

4. Alokasikan masing - masing objek ke dalam centroid yang paling terdekat. Untuk melakukan pengalokasian objek kedalam masing - masing *cluster* pada saat iterasi secara umum dapat dilakukan dengan dua cara yaitu dengan *hard k-means*, dimana secara tegas setiap objek dinyatakan sebagai anggota *cluster* dengan mengukur jarak kedekatan sifatnya terhadap titik pusat *cluster* tersebut, cara lain dapat dilakukan dengan *fuzzy C-Means*.
5. Lakukan iterasi, kemudian tentukan posisi centroid baru dengan menggunakan persamaan (1).
6. Ulangi langkah 3 jika posisi centroid baru tidak sama. Pengecekan konvergensi dilakukan dengan membandingkan matriks *group assignment* pada iterasi sebelumnya dengan matrik *group assignment* pada iterasi yang sedang berjalan. Jika hasilnya sama maka algoritma *k-means cluster analysis* sudah konvergen, tetapi jika berbeda maka belum konvergen sehingga perlu dilakukan iterasi berikutnya.

II.4 *Unified Modelling Language (UML)*


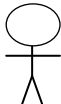
Hasil pemodelan pada OOAD (*Object Oriented Analysis And Design*) terdokumentasikan dalam bentuk *Unified Modeling Language (UML)*. UML

adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut (Gellysa Urva, Helmi Fauzi Siregar: 2015 : 93):

1. *Usecase* Diagram

Usecase diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Usecase* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *usecase* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi - fungsi tersebut.

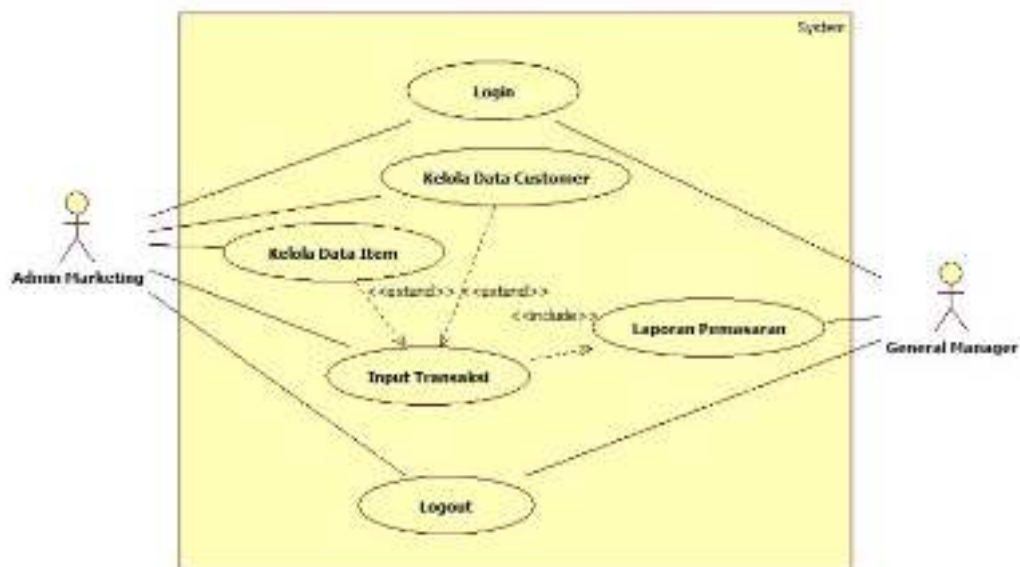
Tabel II.2. Simbol *Use Case*

Simbol	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, dan dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .

—————	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
—————>	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
----->	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
<-----	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar : 2015 : 94)

Adapun contoh dari penggunaan *Usecase Diagram* dapat dilihat pada gambar sebagai berikut:




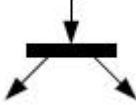

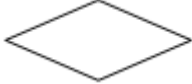
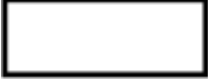


Gambar II.2. Diagram Usecase

2. Diagram Aktivitas (*Activity Diagram*)

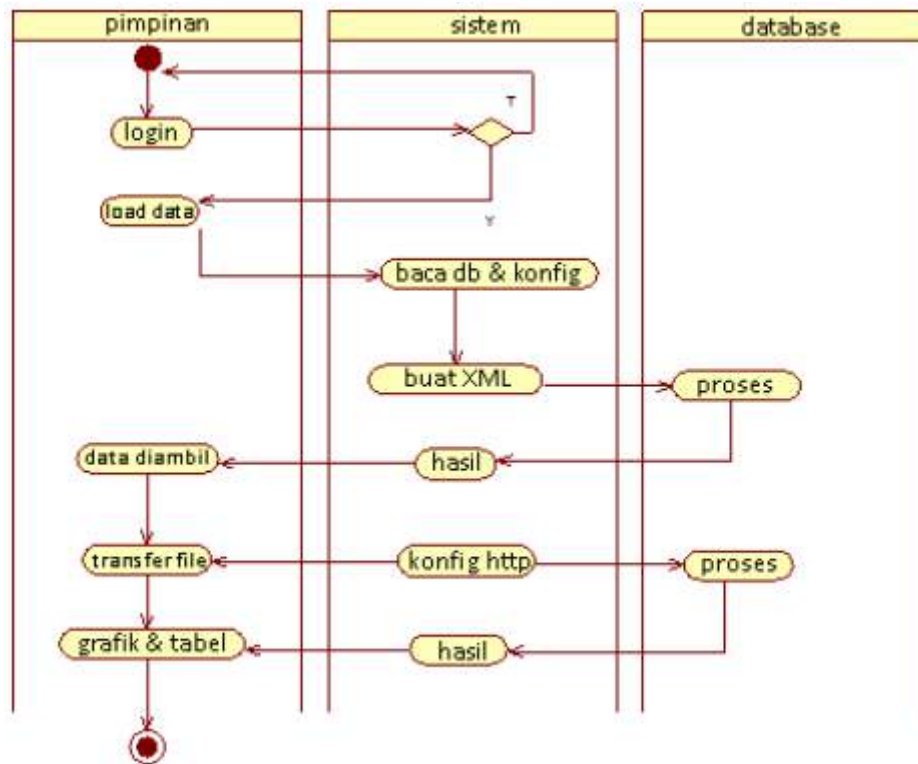
Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis.

Tabel II.3. Simbol *Activity Diagram*

Simbol	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar : 2015 : 94)

Adapun contoh dari penggunaan *Activity Diagram* dapat dilihat pada gambar sebagai berikut:

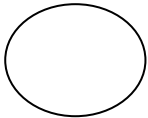
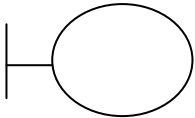


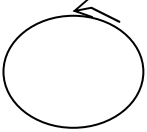

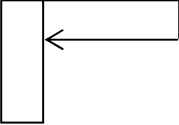


Gambar II.3. Diagram Activity

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *usecase* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek.

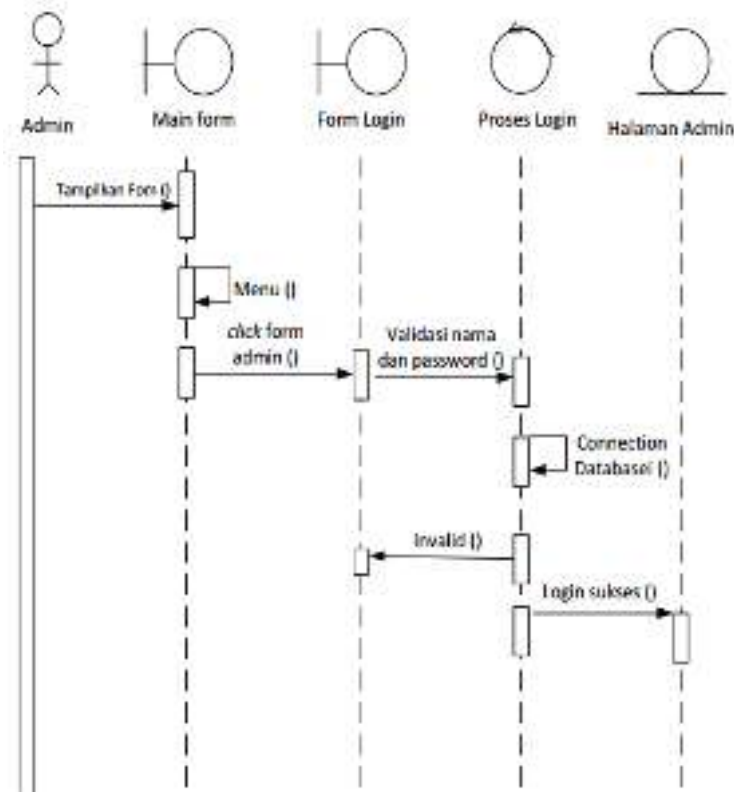
Tabel II.4. Simbol *Sequence Diagram*

Simbol	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.

	<p><i>Control class</i>, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.</p>
	<p><i>Message</i>, simbol mengirim pesan antar <i>class</i>.</p>
	<p><i>Recursive</i>, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.</p>
	<p><i>Activation</i>, <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar : 2015 : 95)

Adapun contoh dari penggunaan *Sequence Diagram* dapat dilihat pada gambar sebagai berikut:



Gambar II.4. Diagram Sequence

4. *Class Diagram* (Diagram Kelas)

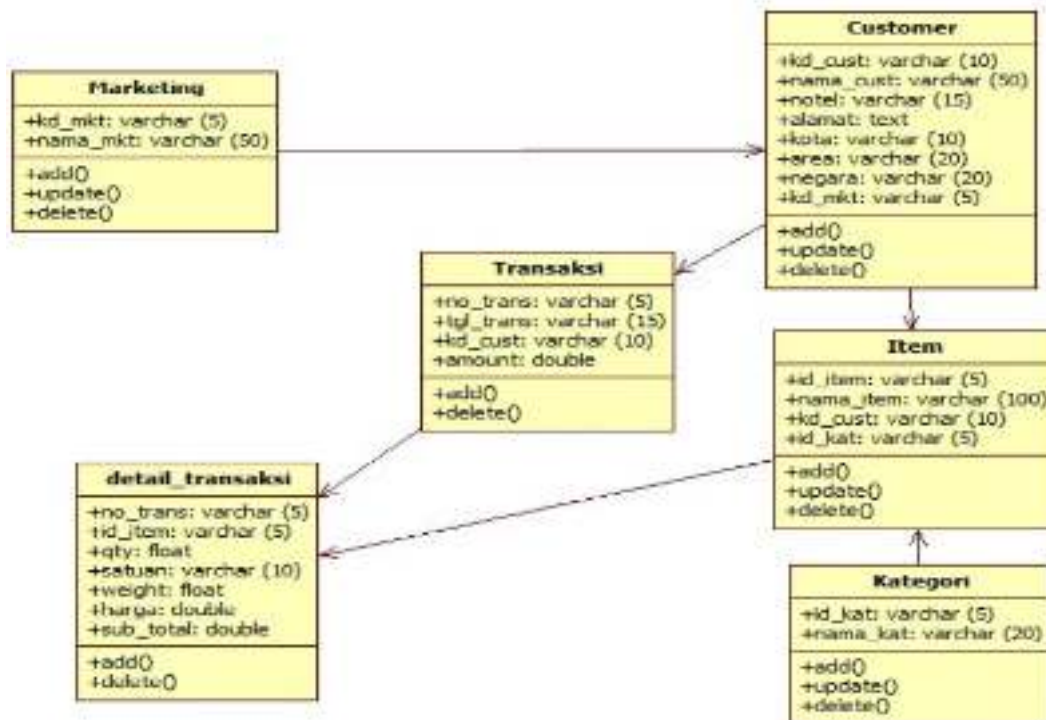
Merupakan hubungan antar kelas dan penjelasan detail tiap - tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan - aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Tabel II.5. Multiplicity Class Diagram

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar : 2015 : 95)

Adapun contoh dari penggunaan *Class Diagram* dapat dilihat pada gambar sebagai berikut:



Gambar II.5. Diagram Class

II.5 SQL Server 2008 R2

SQL Server 2008 adalah sebuah terobosan baru dari Microsoft dalam bidang *database*. SQL Server adalah DBMS (*Database Management System*) yang dibuat oleh Microsoft untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan *Oracle*. SQL Server 2008 dibuat pada saat kemajuan dalam bidang *hardware* sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa SQL Server 2008 membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data. Microsoft merilis

SQL *Server* 2008 dalam beberapa versi yang disesuaikan dengan segment-segment pasar yang dituju (Wenny Widya, dkk : 2013 : 3).

II.6 Microsoft Visual Studio 2010

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi *console*, *aplikasi Windows*, ataupun *aplikasi Web*. *Visual Studio* mencakup kompiler, *Software Development Kit (SDK)*, *Integrated Development Environment (IDE)*, dan dokumentasi (umumnya berupa *MSDN Library*). Kompiler yang dimasukkan ke dalam paket *Visual Studio* antara lain *Visual C++*, *Visual C#*, *Visual Basic*, *Visual Basic .NET*, *Visual InterDev*, *Visual J++*, *Visual J#*, *Visual FoxPro*, dan *Visual SourceSafe* (Herpendi: 2016 : 1).

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas *Windows*) ataupun *managed code* (dalam bentuk *Microsoft Intermediate Language* di atas *.NET Framework*). Selain itu, *Visual Studio* juga dapat digunakan untuk mengembangkan aplikasi *Silverlight*, aplikasi *Windows Mobile* (yang berjalan di atas *.NET Compact Framework*) (Herpendi : 2016 : 1).



Gambar II.6. Tampilan Awal Microsoft Visual Studio 2010