BAB II

TINJAUAN PUSTAKA

II.1 *Software Engineering*

Menurut Janner Simarmata (2010:10) Rekayasa merupakan penerapan ilmu dan teknologi untuk menyelesaikan permasalahan manusia. Sedangkan rekayasa perangkat lunak (RPL) atau *Software engineering* adalah satu bidang yang mendalami cara-cara pengembangan perangkat lunak termasuk pembuatan, pemeliharaan, manajemen organisasi pengembangan perangkat lunak, dan sebagainya (Yuhendra : 2015).

Perangakat Lunak merupakan seluruh perintah yang digunakan untuk memproses informasi. Perangkat lunak dapat berupa program maupun prosedur yang didalam nya merupakan kumpulan perintah yang dimengerti oleh komputer sedangkan prosedur adalah perintah yang dibutuhkan oleh pengguna dalam memproses informasi (Ganda Yoga Swara : 2016).

Dalam merancangkan suatu program perlu memperhatikan *IDE*, Bahasa Pemrograman, dan *Databas*e agar suatu program yang dirancang dapat memenuhi standar yang telah ditetapkan.

II.1.1 *IDE (Integrated Development Environment)*

IDE (*Integrated Development Environment*) adalah program komputer yang digunakan untuk menciptakan aplikasi. *IDE* inilah yang memungkinkan

pemrograman secara *visual* merancang tampilan untuk para *user* (antar muka pemakai) dan menuliskan *listing* program (kode) (Syahminan: 2012).

Dengan menggunakan *IDE* (*Integrated Development Environment*) tertentu, semua kebutuhan pemprograman akan dijadikan menjadi satu tempat. Berikut merupakan perangkat lunak *IDE* (*Integrated Development Environment*) antara lain seperti *NetBeans, Eclipse, Visual Delfi, Dreamweaver, Microsoft visual studio 2010* dan sebagainya.

II.1.1.1 *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.*

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*) (Herpendi : 2016).

II.1.2 Bahasa Pemrograman

Bahasa yang dipakai untuk menginstruksikan komputer disebut bahasa pemrograman. Ada 2 jenis bahasa pemrograman terdiri dari bahasa tingkat tinggi & bahasa tingkat rendah. Kita kenal diantaranya: Basic, Algol, Cobol, Pascal, PL-1,RPG, SNOBOL, APL, LISP, GPSS, ADA, DEAL dan sebagainya yang merupakan bahasa tingkat tinggi. Bahasa tingkat tinggi yaitu bahasa komputer yang memakai instruksi berasal dari unsur kata-kata bahasa manusia, contohnya *begin, end, if, for, while, and, or*, dsb. Bahasa tingkat rendah, atau dikenal dengan istilah bahasa rakitan (bahasa inggris *Assembly*) yaitu memberikan perintah kepada komputer dengan memakai kode-kode singkat (*kode mnemonic*), contohnya MOV, SUB, CMP, JMP, JGE, JL, LOOP, dsb. (Jusuf Wahyudi, dkk : 2013).

Didalam bahasa program, perintah-perintah/rumus-rumus kebanyakan dituliskan dalam bahasa *inggris*. Dari bahasa *inggris* yang sama itu telah dibuat orang berbagai macam cara memberikan perintah pada komputer (Jusuf Wahyudi, dkk : 2013). Ada beberapa contoh bahasa pemrograman yang sering digunakan untuk merancang sebuah program antara lain *PHP, Java, visual basic, C++* dan sebagainya.

II.1.2.1 *Visual Basic*

Visual Basic 6.0 adalah salah satu aplikasi pemrograman *under Windows* yang berbasis pada *visual* atau *grafis*. Aplikasi ini dikeluarkan oleh *Microsoft Cooperation* yang juga pemilik dari sistem operasi *Microsoft Windows*. Pada

awalnya BASIC (*Beginner's Allpurpose Symbolic Instruction Code*) adalah bahasa pemerogramanan yang merupakan awal dari bahasa pemrograman tingkat tinggi sesudahnya, yang berbasis DOS (*Diskette Operating sistem*). *BASIC* memiliki struktur bahasa yang sulit dan memiliki tampilan yang tidak menarik, dengan kemajuan teknologi maka diperlukan suatu aplikasi pemrograman yang bukan hanya cepat tapi juga menarik dan *user friendly* atau mudah digunakan. Maka *Microsoft* mengembangkan *Visual Basic* sebagai salah satu bahasa pemerograman tingkat tinggi berdasarkan dari bahasa pemerograman *BASIC* (Indra Kanedi,dkk: 2013).

II.1.3 *Database*

Basis data merupakan kumpulan data *non redundant* yang dapat digunakan secara bersamaan (*share*) untuk aplikasi yang berbeda-beda. Untuk mendapatkan informasi yang berguna dari kumpulan data maka diperlukan suatu perangkat lunak (*software*) untuk memanipulasi data sehingga mendapatkan informasi yang berguna (Meiska Firstiara Maudi,dkk : 2014). *Database juga* merupakan kumpulan data yang saling berhubungan satu dengan yang lainnya, tersimpan didalam perangkat keras komputer dan perangkat lunak untuk memanipulasi data.

Ada beberapa jenis *DBMS* (*Data Base Management System*) yang sering digunakan dalam merancang suatu aplikasi atau program antara lain *MySQL, SQL Server, Oracle* dan sebagainya.

II.1.3.1 *SQL Server* 2008

Microsoft SQL Server adalah sebuah sistem manajemen basis data *relasional* (*RDBMS*) produk *Microsoft*. *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: 2010).

II.2. Perancangan Sistem

Perancangan sistem adalah tahap setelah analisis dari siklus pengembangan sistem pendefinisian dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi,menggambarkan bagaimana suatu sistem dibentuk (Rara Sri Artati Rejeki : 2011). Perancangan sistem juga dapat di definisikan sebagai penggambaran, perencanaan dan pembuatan *sketsa* atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi (Rara Sri Artati Rejeki : 2011).

Dalam proses perancangan sistem, ada beberapa diagram yang dapat digunakan seperti *DFD* (*Data Flow Diagram*) dan *UML* (*Unified Modelling Language*).

II.2.1 *Unified Modelling Language* (*UML*)

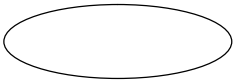
UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. *UML* merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. *UML* saat ini sangat banyak dipergunakan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

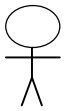
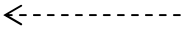
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML antara lain *Use case Diagram, Activity Diagram, Sequence Diagram, Class Diagram* dan berikut penjelasannya :

a. *Use case* Diagram

Use case diagram merupakan pemodelan untuk kelakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut(Gellysa Urva dan Helmi Fauzi Siregar : 2015).

Tabel II.1. Simbol *Use case*

Gambar	Keterangan
	*Use case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama *use case*.

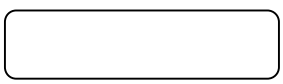
	Aktor adalah *abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan *use case*, tetapi tidak memiliki control terhadap *use case*.
	Asosiasi antara aktor dan *use case*, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan *use case* yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	Include, merupakan di dalam *use case* lain (*required*) atau pemanggilan *use case* oleh *use case* lain, contohnya adalah pemanggilan sebuah fungsi program.
	*Extend*, merupakan perluasan dari *use case* lain jika kondisi atau syarat terpenuhi.

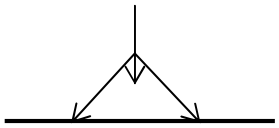
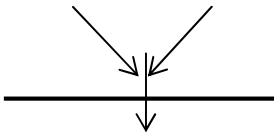
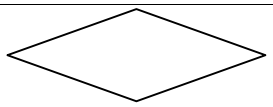
(Sumber : Gellysa Urva dan Helmi Fauzi Siregar : 2015: 94)

b. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis (Gellysa Urva dan Helmi Fauzi Siregar : 2015).

Tabel II.2. Simbol *Activity Diagram*

Gambar	Keterangan
	*Start point*, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	*End point*, akhir aktifitas.
	*Activites*, menggambarkan suatu proses/kegiatan bisnis.

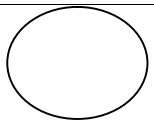
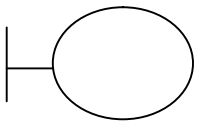
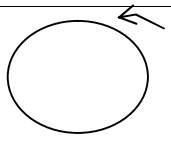
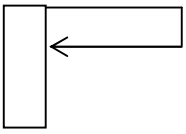
	*Fork* (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	*Join* (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	*Decision Points*, menggambarkan pilihan untuk pengambilan keputusan, *true, false*.
	*Swimlane*, pembagian *activity* diagram untuk menunjukkan siapa melakukan apa.

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

c. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *usecase* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek (Gellysa Urva dan Helmi Fauzi Siregar : 2015).

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	*EntityClass*, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	*Boundary Class*, berisi kumpulan kelas yang menjadi *interface* atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan *form* cetak.
	*Control class*, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	Message, simbol mengirim pesan antar *class*.
	*Recursive*, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.

	*Activation, activation* mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	Lifeline, garis titik-titik yang terhubung dengan objek, sepanjang *lifeline* terdapat *activation*.

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

d. *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 (Gellysa Urva dan Helmi Fauzi Siregar : 2015).

Tabel II.4. *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)

II.3 *Granite*

Granite merupakan batuan beku *plutonik* yang terbentuk dari hasil pembekuan *magma* yang bersifat asam dimana memiliki kandungan *silika* lebih besar dari 66% (Muhammad Edwin,dkk : 2014). Tetapi *granite* pada perusahaan PT. Jui Shin Indonesia merupakan *granite* yang diproduksi dengan beberapa material tanah dan kimia yang di proses melalui pembakaran di suhu yang lebih

tinggi (1230° C) dan dipress dengan kekuatan 7200 PH. Hal ini menjadikan material *granite* lebih menyatu dan lebih padat.

Karakteristik *granite* yang keras, kuat, serta memiliki *motif* dan warna yang cerah serta kekuatan dari lapisan atas sampai lapisan bawah mempunyai mutu material yang sama membuat *granite* tersebut banyak digunakan dalam berbagai bidang bangunan. Setelah diasah dan dihaluskan, batuan *granite* lembaran dapat dipotong- potong dan dijadikan ubin dengan warna- warna dan *motif* yang alami. Pada umumnya ubin tersebut digunakan untuk ubin lantai, anak tangga maupun dinding berbagai ruangan seperti kamar mandi dan dapur.

Granite juga memiliki mineral-mineral *asesoris* yang dapat diketahui ketika melakukan pengamatan *petrografi*. Mineral-mineral *asesoris* tersebut dapat memberikan informasi tentang keberadaan kandungan mineral-mineral yang ekonimis seperti mineral *radioaktif*. Mineral *radioaktif* umumnya banyak dijumpai pada batuan beku yang bersifat asam terutama pada batuan *granite* (Muhammad Edwin, dkk : 2014).

II.4 Teknologi Komputer

Teknologi komputer adalah pemanfaat komputer sebagai penunjang sarana kehidupan manusia. Teknologi komputer berkembang dengan pesat sehingga dapat digunakan manusia untuk membantu setiap pekerjaannya. Perkembangan teknologi komputer telah menyentuh seluruh bidang ilmu pengetahuan. Beberapa perwujudan dari teknologi komputer ialah *data mining*, sistem pakar, dan sistem pendukung keputusan.

II.4.1 *Data Mining*

Data mining adalah proses menganalisa data dari *perspektif* yang berbeda dan menyimpulkannya menjadi informasi-informasi penting yang dapat dipakai untuk meningkatkan keuntungan,memperkecil biaya pengeluaran, atau bahkan keduanya. Secara teknis, *data mining* dapat disebut sebagai proses untuk menemukan *korelasi* atau *poladari* ratusan atau ribuan *field* dari sebuah *relasional database* yang besar. (Angga Ginanjar Mabrurdan Riani Lubis : 2012)

Kemampuan *Data mining* untuk mencari informasi bisnis yang berharga dari basis data yang sangat besar, dapat dianalogikan dengan penambangan logam mulia dari lahan sumbernya,teknologi ini dipakai untuk (Angga Ginanjar Mabrurdan Riani Lubis : 2012) :

1. Prediksi *trend* dan sifat-sifat bisnis, dimana *data mining* mengotomatisasi proses pencarian informasi memprediksi di dalam basis data yangbesar.
2. Penemuan pola-pola yang tidak diketahui sebelumnya, dimana *data mining* menyapu basis data, kemudian mengidentifikasi pola-pola yang sebelumnya tersembunyi dalam satu sapuan.

II.4.2 Sistem Pakar

Sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tersebut. Sistem pakar juga merupakan salah satu cabang dari *Artificial Intelligence* yang membuat penggunaan secara luas *knowledge* yang khusus untuk penyelesaian masalah tingkat manusia yang pakar. Seorang pakar adalah orang yang

mempunyai keahlian dalam bidang tertentu yaitu pakar yang mempunyai *knowledge* atau kemampuan khusus yang orang lain tidak mengetahui atau mampu dalam bidang yang dimilikinya (Muklis Budi Rackman,dkk : 2014).

Tujuan mengembangkan sistem pakar sebenarnya bukan untuk menggantikan peran manusia, tetapi untuk mengalihkan pengetahuan manusia ke dalam bentuk sistem, sehingga dapat digunakan oleh orang banyak dan tidak terbatas oleh waktu (Feriani A. Tarigan :2014).

II.4.3 Sistem Pendukung Keputusan (SPK)

Sistem pendukung keputusan (SPK) adalah bagian dari sistem informasi berbasis komputer (termasuk sistem pengetahuan) yang dipakai untuk mendukung pengambilan keputusan dalam suatu organisasi atau perusahaan (Murnawan 2012).

Menurut Dicky Nofriansyah (2014:1), Sistem pendukung keputusan (SPK) dibangun untuk mendukung solusi atas suatu masalah atau untuk suatu peluang. Aplikasi Sistem pendukung keputusan (SPK) digunakan untuk pengambilan keputusan. Aplikasi Sistem pendukung keputusan (SPK) menggunakan *CBIS (Computer Based Information System)* yang *fleksibel, interaktif* dan dapat di

adaptasi yang dikembangkan untuk mendukung solusi masalah manajemen spesifik yang tidak terstruktur. Dalam penelitian ini digunakanlah metode *topsis* yang merupakan salah satu metode untuk menyelesaikan masalah yang ada.

II.4.3.1 Metode *TOPSIS*

Metode *TOPSIS* adalah salah satu metode yang digunakan untuk menyelesaikan masalah *Multi Attribute Decision Making* (MADM). Metode *TOPSIS* didasarkan pada konsep dimana alternatif terpilih yang terbaik tidak hanya memiliki jarak terpendek dari solusi ideal positif, namun juga memiliki jarak terpanjang dari solusi ideal negatif .

Konsep ini banyak digunakan pada beberapa model *Multi Attribute Decision Making (MADM)* untuk menyelesaikan masalah keputusan secara praktis. Hal ini disebabkan karena konsepnya yang sederhana dan mudah dipahami, komputasinya efisien,dan memiliki kemampuan untuk mengukur kinerja relatif dari alternatif-alternatif keputusan dalam bentuk matematis yang sederhana. Secara umum, prosedur *TOPSIS* mengikuti langkah-langkah sebagai berikut :

1. Membuat *matriks* keputusan yang ternormalisasi.
2. Membuat *matriks* keputusan yang ternormalisasi terbobot.
3. Menentukan *matriks* solusi ideal positif & *matriks* solusi ideal negatif.
4. Menentukan jarak antara nilai setiap alternatif dengan *matriks* solusi ideal positif & *matriks* solusi ideal negatif.
5. Menentukan nilai preferensi untuk setiap alternatif. TOPSIS membutuhkan rating kinerja setiap alternatif Ai pada setiap criteria Cj yang ternormalisasi, yaitu :

a. Rangking Tiap Alternatif

TOPSIS membutuhkan ranking kinerja setiap alternatif Ai pada setiap kriteria Cj yang ternormalisasi yaitu : kriteria Cj yang ternormalisasi yaitu:

$$r_{ij} = \frac{x}{\sqrt{\sum_{i-1}^{m} x_i^2}}$$ dengan i=1,2,....m; dan j=1,2,.....n;

b. *Matriks* keputusan ternormalisasi terbobot

$$y_{ij} = w_i r_{ij};$$ dengan i=1,2,...,m dan j=1,2,...,n

c. Solusi Ideal Positif Dan Negatif

Solusi ideal positif A+ dan solusi ideal negatif A- dapat ditentukan berdasarkan ranking bobot ternormalisasi (yij) sebagai berikut :

$$A^+ = (y_1^+, y_2^+, \dots, y_n^+);$$

$$A^- = (y_1^-, y_2^-, \dots, y_n^-);$$

dimana :

y_j^+ adalah :
- max y_{ij}, jika j adalah atribut keuntungan
- min y_{ij}, jika j adalah atribut biaya

y_j^- adalah :
- min y_{ij}, jika j adalah atribut keuntungan
- max y_{ij}, jika j adalah atribut biaya

d. Jarak Dengan Solusi Ideal

Jarak adalah alternatif Ai dengan solusi ideal positif dirumuskan sebagai :

$$D_i^+ = \sqrt{\sum_{j-1}^{n} \left(y_i^+ - y_{ij}\right)^2}$$; i=1,2,...,m

Jarak adalah alternatif Ai dengan solusi ideal negatif dirumuskan sebagai :

$$D_i^- = \sqrt{\sum_{j-1}^{n} \left(y_i^- - y_{ij}\right)^2}$$; i=1,2,...,m

e. Nilai Preferensi Untuk Setiap Alternatif

Nilai preferensi untuk setiap alternatif (Vi) diberikan sebagai :

$$V_i = \frac{D_i^-}{D_i^- + D_i^+} \quad ; i=1,2,...,m$$

Nilai Vi yang lebih besar menunjukkan bahwa alternatif Ai lebih dipilih.