BAB II

LANDASAN TEORI

II.1. Sistem

Sistem adalah sekelompok unsur yang erat berhubungan satu dengan yang lainnya yang berfungsi bersama-sama untuk mencapai tujuan tertentu (Lasminiasih, dkk : 2016).

Secara garis besar terdapat dua kelompok pendekatan yang dilakukan yaitu (Asbon Hendra : 2012:157):

1. Pendekatan sistem yang lebih menekan pada elemen-elemen atau kelompoknya, didalam hal ini sistem ini didefenisikan sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu.
2. Pendekatan sistem sebagai jaringan kerja dari prosedur, yang lebih menekankan urutan operasi didalam sistem. Prosedur didefenisikan sebagai urutan operasi kerja (tulis-menulis), yang bisanya melibatkan beberapa orang didalam satu atau lebih departement yang diterapkan untuk menjamin penanganan yang seragam dari transaksi bisnis yang terjadi.

II.1.1 Karakteristik Sistem

Menurut Asbon Hendra (2012:158), sistem mempunyai beberapa karakteristik atau sifat-sifat tertentu antara lain:

1. Komponen (*Components*)

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama untuk membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupaya suatu subsistem atau bagian dari sistem. Setiap sistem tidak peduli berapapun kecilnya, selalu mengandung komponen-komponen atau subsistem.

2. Batas sistem (*Boundary*)

   Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarannya.

3. Lingkungan luar sistem (*Environtment*)

   Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung sistem (*InterfaceI)*

   Merupakan media penghubung antara suatu subsistem dengan subsistem yang lainnya. Melalui perhubungan ini memungkinkan sumber-sumber daya mengalir dari subsistem yang lainnya.

5. Masukkan sistem (*input*)

   Merupakan energi yang dimasukkan ke dalam sistem. Masukkan dapat berupa masukkan perawatan (*maintenance input*) dan masukkan sinyal (*signal input*).

6. Keluaran sistem (*output*)

   Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolah sistem (*Process*)

   Suatu sistem yang dapat mempunyai suatu bagian pengolah yang akan merubah masukkaan menjadi keluaran.

8. Sasaran sistem (*Objective*)

   Suatu sistem pasti mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya, suatu sistem dikatakn berhasil bila mengenai sasaran atau tujuannya.

II.1.2 Klasifikasi Sistem

Menurut Asbon Hendra (2012:160), sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya sebagai berikut:

1. Sistem abstrak dan sistem fisik

   Sistem abstrak adalah sistem yang berisi gagasan atau konsep.misalnya, sistem teknologi yang berisi gagasan tentang hubungan manusia dan Tuhan. Sistem fisik merupakan sistem yang secara fisik dapat dilihat. Misalnya sistem komputer, sistem sekolah, sistem akuntansi, dan sistem transportasi.

2. Sistem alamiah dan sistem buatan manusia

   Sistem alamiah adalah sistem yang terjadi karena alam (tidak dibuat manusia). Misalnya, sistem tata surya. Sistem buatan manusia adalah sistem yang dibuat oleh manusia. Misalnya, sistem komputer dan sistem mobil.

3. Sistem tertentu dan sistem tak tertentu

   Sistem tertentu adalah sistem yang operasinya dapat diprediksi secara tepat. Misalnya, sistem komputer. Sistem tak tertentu adalah sistem yang tidak dapat

diramal diramal dengn pasti karena mengandung unsur probabilitas. Misalnya, sistem arisan dan sistem sediaan.

4. Sistem tertutup dan sistem terbuka

   Sistem tertutup adalah sistem yang tidak bertukar materi, informasi, atau energi dengan lingkungannya. Misalnya, reaksi kimia dalam tabung terisolasi. Sistem terbuka adalah sistem yang berhubungan dengan lingkungan dan diperbaharui oleh lingkungan.

II.2. Sistem Penunjang Keputusan

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).

Komponen sistem penunjang keputusan adalah (Eko Darmanto : 2014):

1. *Data Management* (Manajemen Data)

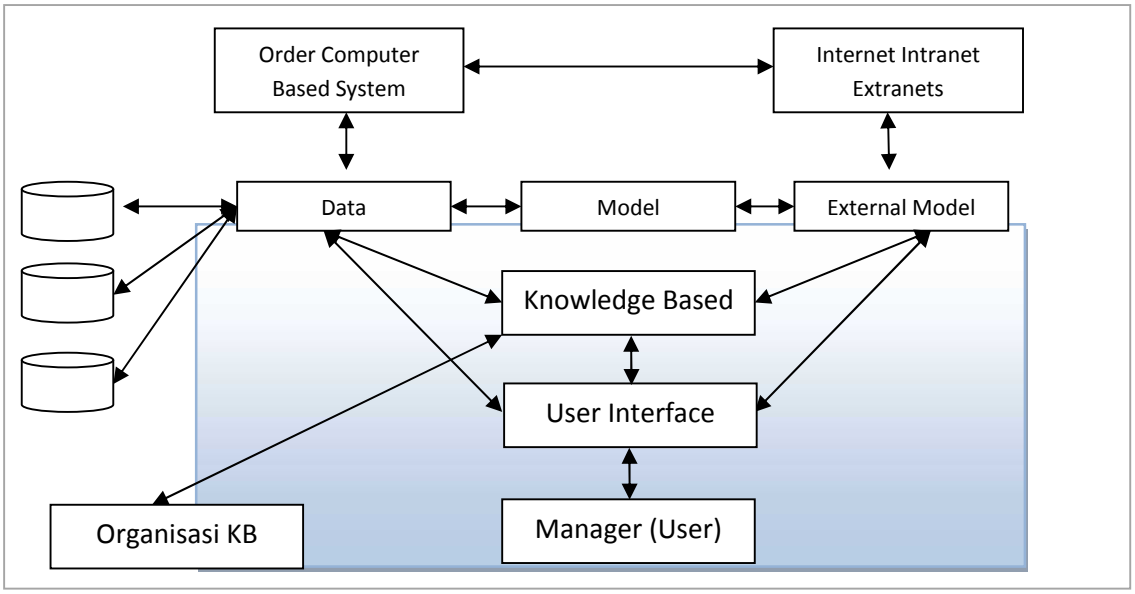
   Merupakan komponen SPK sebagai penyedia data bagi sistem, yang mana data disimpan dalam *Database Management System*(DBMS), sehingga dapat diambil dan diekstraksi dengan cepat.

2. *Model Management* (Manajemen Model)

   Melibatkan model finansial, statistikal, manajemen science, atau berbagai model kuantitatif lainnya, sehingga dapat memberikan ke sistem suatu kemampuan analistis, dan manajemen software yang diperlukan.

3. *Communication* (dialog subsistem)

   User dapat berkomunikasi dan memberikan perintah pada SPK melalui subsistem ini. Ini berarti menyediakan antarmuka.



Gambar II.1 Skema Sistem Pengambil Keputusan

(Sumber : Munarwan : 2012)

Tujuan dari SPK :

1. Membantu menyelesaikan masalah semi-terstruktur
2. Mendukung manajer dalam mengambil keputusan
3. Meningkatkan efektifitas bukan efisiensi pengambilan keputusan

Dalam pemrosesannya, SPK dapat menggunakan bantuan dari sistem lain seperti *Artificial Intelligence*, *Expert Systems*, *Fuzzy Logic*, dll (Munarwan : 2012).

II.3. MFEP

Multi Factor Evaluation Process (MFEP) adalah metode kuantitatif yang menggunakan *Weighting System*. Dalam pengambilan keputusan multi faktor,

pengambil keputusan secara subyektif dan intuitif menimbang berbagai faktor atau kriteria yang mempunyai pengaruh penting terhadap alternatif pilihannya. Untuk keputusan yang berpengaruh secara strategis, lebih dianjurkan menggunakan sebuah pendekatan kuantitatif seperti MFEP (Muhammad Reza Okaviana : 2014).

Dalam MFEP pertama-tama seluruh kriteria yang menjadi faktor penting dalam melakukan pertimbangan diberikan pembobotan (weighting) yang sesuai. Langkah yang sama juga dilakukan terhadap alternatif-alternatif yang akan dipilih, yang kemudian dapat dievaluasi berkaitan dengan faktor–faktor pertimbangan tersebut. Metode MFEP menentukan bahwa alternatif dengan nilai tertinggi adalah solusi terbaik berdasarkan kriteria yang telah dipilih (Erna Lovita : 2013).

$$\boldsymbol{WE = FW \times E}$$

Keterangan :

WE : Nilai bobot evaluasi

FW : Nilai bobot factor

E : Nilai evaluasi faktor

II.4. Mutasi Pegawai Cabang Kejaksaan Deli Serdang di Labuhan Deli

Kejaksaan menurut Undang- Undang No.16 Tahun 2004 tentang Kejaksaan R.I. dalam Pasal 2 memberikan pengertian bahwa Kejaksaan adalah lembaga pemerintahan yang melaksanakan kekuasaan negara dibidang penuntutan

serta kewenangan lain berdasarkan undang-undang. Kekuasaan ini dilakukan secara merdeka dan tidak dapat dipisahkan (Serli Patulak : 2013).

Kekuasaan Kejaksaan dilakukan oleh Kejaksaan Agung, Kejaksaan Tinggi, Kejaksaan Negeri dan didalam menyelesaikan suatu perkara pidana harus memperhatikan norma-norma keagamaan, perikemanusiaan, kesopanan dan kesusilaan (Pasal 3 UU Nomor 16 Tahun 2004). Kejaksaan Negeri sendiri adalah pelaksana kekuasaan Kejaksaan pada tingkat pertama yang menangani terjadinya tindak pidana. Kejaksaan Negeri berkedudukan di ibukota Kabupaten/Kota yang daerah hukumnya meliputi daerah Kabupaten/Kota (M. Yuhdi : 2014).

Cabang Kejaksaan Negeri Deli Serdang Di Labuhan Deli sebagai bagian dari Kejaksaan Negara Kesatuan Republik Indonesia melakukan beberapa kali mutasi terhadap pegawai negeri sipil kejaksaan yang diambil dengan melakukan perhitungan nilai kinerja secara manual. Kriteria-keriteria yang diambil dalam penilaian ialah orientasi pelayanan, integritas, komitmen, disiplin, dan kerja sama.

II.5. *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).

Visual Basic dibuat oleh Microsoft, merupakan salah satu bahasa pemrograman berorientasi objek yang mudah dipelajari. Selain digunakan dalam pembuatan berbagai aplikasi, terutama aplikasi *database*. *Visual Basic* merupakan bahasa pemrograman *event drive*, dimana program akan menunggu sampai ada respons dari *user* / pemakai program aplikasi yang dapat berupa kejadian atau event, misalnya ketika *user* mengklik tombol atau menekan Enter (Edy Winarno : 2010).

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).

Visual Studio sebelumnya versi *Visual Studio* 9.0.21022.08, atau dikenal dengan sebutan *Microsoft Visual Studio* 2008 yang diluncurkan pada 19 November 2007, yang ditujukan untuk *platform Microsoft .NET Framework* 3.5. Versi sebelumnya, *Visual Studio* 2005 ditujukan untuk *platform .NET Framework* 2.0 dan 3.0. *Visual Studio* 2003. ditujukan untuk *.NET Framework* 1.1, dan *Visual Studio* 2002 ditujukan untuk *.NET Framework* 1.0. Versi-versi tersebut di atas kini dikenal dengan sebutan *Visual Studio .NET*, karena memang membutuhkan

Microsoft .NET Framework. Sementara itu, sebelum muncul *Visual Studio .NET*, terdapat *Microsoft Visual Studio* 6.0 (Herpendi, 2016).

II.6. *SQL Server* 2008

Database merupakan kumpulan data yang saling berhubungan satu dengan yang lainnya, tersimpan di dalam perangkat keras komputer dan perangkat lunak untuk memanipulasi (Anisya :2013).

SQL Server 2008 adalah sebuah RDBMS (*Relational Database Management System*) yang sangat *powerful* dan telah terbukti kekuatannya dalam mengolah data. Dalam versi terbarunya ini, *SQL Server* 2008 memiliki banyak fitur yang bisa dihandalkan untuk meningkatkan performa database. *SQL Server* 2008 memiliki suatu GUI (*Graphic User Interface*) yang kita gunakan untuk melakukan aktivitas sehari hari berkaitan dengan database,seperti menulis T-SQL, melakukan beckup dan restore database, melakukan security database terhadap aplikasi, dan sebagainya. Pada GUI tersebut kita bisa melakukan settingan terhadap *SQL Server* untuk berkerja lebih optimal. Settingan juga bisa dilakukan menggunakan *script* untuk memudahkan *developer* mengubah *Setting* Opsions pada *SQL Server* 2008 (Ruslan, 2013).

II.7. *UML (Unified Modeling Language)*

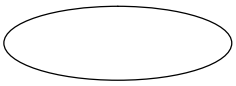
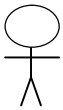
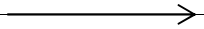
UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML

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 berbasiskan UML adalah sebagai berikut (Gellysa Urva dan Helmi Fauzi Siregar : 2015):

a. *Use case* Diagram

Use case mendeskripsikan fungsi dari sebuah sistem dari perspektif/sudut pandang para pengguna sistem. *Use case* mendefinisikan apa yang dilakukan oleh sistem dan elemen-elemennya, bukan bagaimana sistem dan elemen-elemennya saling berinteraksi (Munarwan : 2012).

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

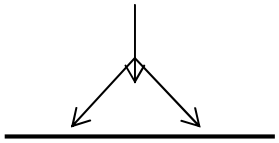
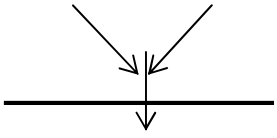
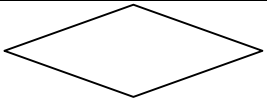
	panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	Include, merupakan di dalam *use case* lain (*required*) atau pemanggilan *use case* oleh *use case* lain, contohnya adalah pemanggilan sebuah fungsi program.
	Extend, merupakan perluasan dari *use case* lain jika kondisi atau syarat terpenuhi.

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar : 2014)

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

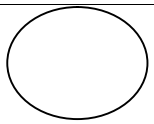
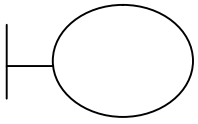
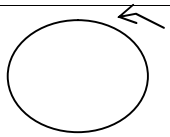
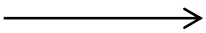
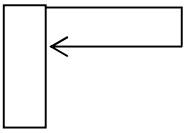
	menunjukkan siapa melakukan apa.

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

c. Diagram Urutan (*Sequence Diagram*)

Sequence diagram mendokumentasikan komunikasi atau interaksi antar kelas-kelas. *Sequence diagram* menunjukkan sejumlah objek dan *message* (pesan) yang diletakkan diantara objek-objek di dalam *use case* (Munarwan : 2012).

Tabel II.3. Simbol *Sequence Diagram*

Gambar	**Keterangan**
	*Entity Class,* 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)