

## **BAB II**

### **LANDASAN TEORI**

#### **II.1. Sistem Pendukung Keputusan**

Sistem Pendukung Keputusan adalah sebuah sistem yang dimaksudkan untuk mendukung para pengambil keputusan manajerial dalam situasi keputusan semi terstruktur. Sistem pendukung keputusan dimaksudkan untuk menjadi alat bantu bagi para pengambil keputusan untuk memperluas kapabilitas mereka, namun tidak untuk menggantikan penilaian mereka. Sistem pendukung keputusan ditujukan untuk keputusan-keputusan yang memerlukan penilaian atau pada keputusan keputusan yang sama sekali tidak dapat didukung oleh algoritma (Siti Kholijah Ritonga, 2013 : 143).

Sistem Pendukung Keputusan (SPK) merupakan suatu pendekatan atau metodologi untuk mendukung keputusan. SPK menggunakan CBIS (*Computer Based Information System*) yang fleksibel, interaktif dan dapat diadaptasi, yang dikembangkan untuk mendukung solusi untuk masalah manajemen spesifik yang tidak terstruktur. SPK menggunakan data, memberikan antarmuka pengguna yang mudah dan dapat menggabungkan pemikiran pengambil keputusan.

Sebagai tambahan, SPK biasanya menggunakan berbagai model dan dibangun oleh suatu proses interaktif dan iterative. Ia mendukung semua fase pengambilan keputusan dan dapat memasukkan suatu komponen pengetahuan. SPK dapat digunakan oleh pengguna tunggal pada satu PC atau bisa menjadi

berbasis Web untuk digunakan oleh banyak orang pada beberapa lokasi (Nuri Guntur Perdana, 2013 : 266).

Sistem Pendukung Keputusan (SPK) merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan dan pemanipulasian data. Sistem digunakan untuk membantu pengambilan keputusan dalam situasi yang semi terstruktur dan situasi yang tidak terstruktur, dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat (Bunga Annete Benning, 2015 : 3).

### **II.1.1. Komponen-Komponen Sistem Pendukung Keputusan**

Adapun komponen-komponen dalam sistem pendukung keputusan sebagai berikut (Bunga Annete Benning, 2015 : 2) :

#### **1. Subsistem Manajemen Data**

Subsistem manajemen data terdiri dari elemen berikut ini :

- a. *Decision Support System (DSS) database* adalah kumpulan data yang saling terkait yang diorganisir untuk memenuhi kebutuhan dan struktur sebuah organisasi dan dapat digunakan oleh lebih dari satu orang untuk lebih dari satu aplikasi.
- b. Sistem manajemen basis data adalah pembuatan, pengaksesan, dan pembaharuan (*update*) oleh DBMS yang mempunyai fungsi utama sebagai tempat penyimpanan, mendapatkan kembali (*retrieval*) dan pengontrolan.

- c. Direktori merupakan sebuah katalog dari semua data di dalam basis data.
- d. *Query Facility*, yang menyediakan fasilitas akses data. Fungsi utamanya adalah untuk operasi seleksi dan manipulasi data dengan menggunakan model-model yang sesuai dari model management.

## 2. Subsistem Manajemen Model

Subsistem manajemen model terdiri atas elemen-elemen berikut ini :

### a. Basis Model

Berisikan model-model seperti manajemen keuangan, statistik, ilmu manajemen yang bersifat kuantitatif yang memberikan kapabilitas analisis pada sebuah SPK. Model Strategis digunakan untuk mendukung manajemen puncak dalam menjalankan tanggung jawab perencanaan strategis.

### b. Sistem Manajemen Basis Model

Merupakan sistem *software* yang fungsi utamanya untuk membuat model dengan menggunakan bahasa pemrograman, alat SPK dan atau subrutin, dan balok pembangunan lainnya; membangkitkan rutin baru dan laporan; pembaruan dan perubahan model; dan manipulasi model.

### c. Direktori Model Peran

Direktori model sama dengan direktori basis data. Direktori model adalah katalog dari semua model dan perangkat lunak lainnya pada basis model. Ia berisi defenisi model dan fungsi utamanya adalah menjawab pertanyaan tentang ketersediaan dan kapabilitas model.

#### d. Eksekusi Model, Integrasi, dan Prosesor

Perintah Eksekusi model adalah proses mengontrol jalannya model saat ini. Integrasi model mencakup gabungan operasi beberapa model saat diperlukan atau mengintegrasikan SPK dengan aplikasi lain. Sedangkan prosesor model digunakan untuk menerima dan menginterpretasikan berbagai macam instruksi pemodelan.

### 3. Subsistem Dialog

Komponen dialog SPK adalah perangkat lunak dan perangkat keras yang menyediakan antarmuka untuk SPK. Istilah antarmuka pengguna mencakup semua aspek komunikasi antara satu pengguna dan SPK. Cakupannya tidak hanya perangkat lunak dan perangkat keras, tapi juga faktor-faktor yang berkaitan dengan kemudahan pengguna, kemampuan untuk dapat diakses, dan interaksi manusia-mesin.

### 4. Subsistem Manajemen Knowledge

Banyak masalah tak terstruktur dan bahkan semi terstruktur yang sangat kompleks sehingga solusinya memerlukan keahlian. Oleh karena itu banyak SPK canggih yang dilengkapi dengan komponen manajemen knowledge. Komponen ini menyediakan keahlian untuk memecahkan beberapa aspek masalah dan memberikan pengetahuan yang dapat meningkatkan operasi komponen SPK lainnya.

### **II.1.2. Keuntungan Dan Keterbatasan Sistem Pendukung Keputusan**

Sistem pendukung keputusan dapat memberikan berbagai manfaat atau keuntungan bagi pemakainya, antara lain (Nungsiati, 2013: 28) :

1. Memperluas kemampuan pengambil keputusan dalam memproses data/informasi bagi pemakainya.
2. Membantu pengambilan keputusan dalam hal penghematan waktu yang dibutuhkan untuk memecahkan masalah terutama berbagai masalah yang sangat kompleks dan tidak terstruktur.
3. Dapat menghasilkan solusi dengan lebih cepat serta hasilnya dapat diandalkan.
4. Walaupun suatu Sistem Pendukung Keputusan, mungkin saja tidak mampu memecahkan masalah yang dihadapi oleh pengambil keputusan, namun dapat menjadi stimulant bagi pengambil keputusan dalam memahami persoalannya, karena sistem pendukung keputusan mampu menyajikan berbagai alternatif.

### **II.2. Pengertian Sertifikasi Guru**

Sertifikasi guru adalah proses pemberian sertifikat pendidik kepada guru. Sertifikat pendidik diberikan kepada guru yang telah memenuhi standar profesional guru. Sertifikasi guru dilakukan oleh Lembaga Pendidikan Tenaga Kependidikan (LPTK) yang terakreditasi dan ditetapkan oleh pemerintah. Proses penilaian sertifikasi guru ini menggunakan dokumen portofolio, yang penilaian portofolionya masih menggunakan cara manual. Banyaknya portofolio yang harus

dinilai, tentu membutuhkan waktu yang lama dalam proses penilaian khususnya dalam menghitung nilai peserta sertifikasi dan merekap hasil penilaian (I Gusti Made Murjana, 2015 : 64).

Sertifikasi adalah proses uji kompetensi yang dirancang untuk mengungkapkan penguasaan kompetensi seseorang sebagai landasan pemberian sertifikat pendidik (Mulyasa, 2007). Sedangkan Kunandar (2009) menyatakan bahwa sertifikasi profesi guru adalah proses untuk memberikan sertifikat kepada guru yang telah memenuhi standar kualifikasi dan standar kompetensi. Sehingga dapat disimpulkan bahwa sertifikasi guru adalah proses pemberian sertifikat kepada guru yang telah memenuhi standat sebagai bukti atau pengakuan atas kemampuan profesionalnya sebagai tenaga pendidik (Hesti Murwati, 2013: 14-15).

### **II.2.1. Tujuan dan Manfaat Sertifikasi Guru**

Wibowo (2004), mengungkapkan bahwa sertifikasi guru bertujuan untuk hal-hal sebagai berikut:

1. Melindungi profesi pendidik dan tenaga kependidikan.
2. Melindungi masyarakat dari praktek-praktek yang tidak kompeten, sehingga merusak citra pendidik dan tenaga kependidikan.
3. Membantu dan melindungi lembaga penyelenggara pendidikan, dengan menyediakan rambu-rambu dan instrument untuk melakukan seleksi terhadap pelamar yang kompeten.

4. Membangun citra masyarakat terhadap profesi pendidik dan tenaga kependidikan.
5. Memberikan solusi dalam rangka meningkatkan mutu pendidik dan tenaga kependidikan (Hesti Murwati, 2013: 15).

### **II.2.2. Dasar Hukum Sertifikasi Guru**

Menurut Dirjen PMPTK Departemen Pendidikan Nasional tahun 2007, dasar hukum sertifikasi profesi guru adalah sebagai berikut:

1. Undang-Undang Republik Indonesia Nomor 20 tahun 2003 Tentang Sistem Pendidikan Nasional:
  - a. Pasal 42 ayat (1), Pendidik harus memiliki kualifikasi minimum dan sertifikasi sesuai dengan jenjang kewenangan mengajar, sehat jasmani dan rohani, serta memiliki kemampuan untuk mewujudkan tujuan pendidikan nasional.
  - b. Pasal 43 ayat (2), Sertifikasi pendidik diselenggarakan oleh perguruan tinggi yang memiliki program pengadaan tenaga kependidikan yang terakreditasi.
2. Undang-Undang Republik Indonesia Nomor 14 Tahun 2005 tentang Guru dan Dosen:
  - a. Pasal 8, Guru wajib memiliki kualifikasi akademik, kompetensi, sertifikat pendidik, sehat jasmani dan rohani, serta memiliki kemampuan untuk mewujudkan tujuan pendidikan nasional.

- b. Pasal 11 ayat (1), Sertifikat pendidik sebagaimana dimaksud dalam pasal 8 diberikan kepada guru yang telah memenuhi persyaratan, ayat (2) Sertifikasi pendidik diselenggarakan oleh perguruan tinggi yang memiliki program pengadaan tenaga kependidikan yang terakreditasi dan ditetapkan oleh Pemerintah, ayat (3) Sertifikasi pendidik dilaksanakan secara objektif, transparan, dan akuntabel, ayat (4) Ketentuan lebih lanjut mengenai sertifikasi pendidik sebagaimana dimaksud pada ayat (2) dan ayat (3) diatur dengan peraturan Pemerintah
3. Peraturan Menteri Pendidikan Nasional nomor 16 tahun 2007 tentang Standar Kualifikasi dan Kompetensi Guru.
  4. Peraturan Menteri Pendidikan Nasional nomor 18 tahun 2007 tentang Sertifikasi bagi Guru dalam Jabatan (Hesti Murwati, 2013: 15-16).

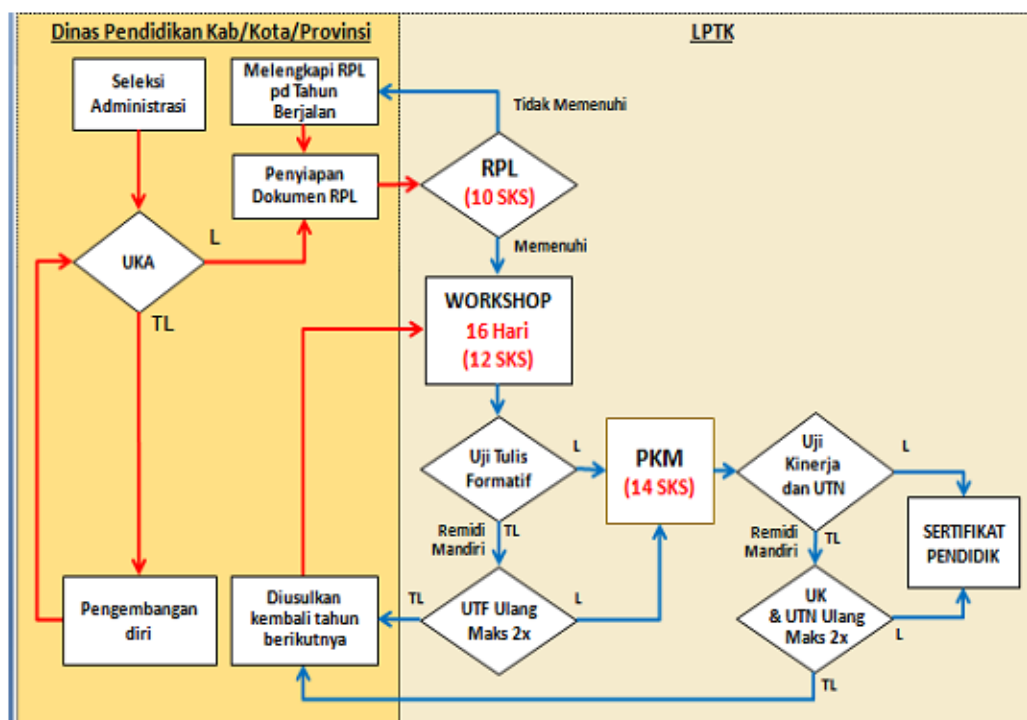
### **II.2.3. Persyaratan Guru Yang Layak Untuk Sertifikasi**

Adapun beberapa persyaratan guru yang layak untuk sertifikasi adalah sebagai berikut:

1. Usia maksimal 40 tahun
2. Masa Kerja berkisar 5 tahun
3. Golongan IV-a
4. Pendidikan terakhir minimal S1 atau D-IV.
5. Jam mengajar sekurang-kurangnya 8 jam.

## II.2.4. Alur Proses Sertifikasi Guru

Dalam proses sertifikasi juga memiliki alur penilaian untuk mengajukan sertifikasi guru. Adapun alur dalam proses penilaian berkas sertifikasi guru yang dilakukan seperti pada gambar dibawah ini:



**Gambar II.1. Alur Proses Sertifikasi Guru.**

(Sumber : I Gusti Made Murjana, 2015 : 69)

Adapun Penjelasan terhadap gambar diatas adalah sebagai berikut:

1. Guru calon peserta sertifikasi guru melalui pendidikan profesi guru dalam jabatan (PPGJ) mengikuti seleksi administrasi yang dilakukan oleh dinas pendidikan provinsi/ kabupaten/kota.
2. Semua guru calon peserta sertifikasi guru melalui PPGJ yang telah memenuhi persyaratan administrasi diikutkan dalam seleksi akademik berbasis data hasil Uji Kompetensi (UKA dan UKG).

3. Bagi peserta yang lulus seleksi akademik dilanjutkan dengan penyusunan rekognisi pengalaman lampau (RPL).
4. Bagi guru yang telah memiliki RPL setara dengan 10 SKS atau lebih ditetapkan sebagai peserta workshop di LPTK. Sedangkan guru yang sudah mencapai sekurang-kurangnya 7 SKS dapat melengkapi kekurangan RPL tersebut dengan durasi waktu maksimal 20 hari sejak diumumkan.
5. Workshop dilaksanakan selama 16 hari (168 JP) di LPTK meliputi kegiatan pendalaman materi, pengembangan perangkat pembelajaran, Penelitian Tindakan Kelas (PTK)/Penelitian Tindakan layanan Bimbingan dan Konseling (PTBK) dan peer teaching/peer counseling yang diakhiri dengan ujian tulis formatif (UTF) dengan instrumen yang disusun oleh LPTK penyelenggara. Peserta sertifikasi guru melalui PPGJ yang lulus UTF akan dilanjutkan dengan melaksanakan Pemantapan Kemampuan Mengajar (PKM) di sekolah tempat guru bertugas. Bagi peserta sertifikasi guru melalui PPGJ yang tidak lulus UTF, diberi kesempatan mengikuti UTF ulang maksimum 2 (dua) kali dan apabila tidak lulus setelah 2 (dua) kali mengikuti ujian ulang, dikembalikan ke dinas pendidikan provinsi/kabupaten/kota untuk memperoleh pembinaan dan dapat langsung diusulkan kembali untuk mengikuti workshop pada tahun berikutnya.

6. PKM dilaksanakan di sekolah selama 2 bulan (di luar libur antar semester) dengan kegiatan-kegiatan sesuai tugas pokok guru yang meliputi penyusunan perangkat pembelajaran (RPP/RPPBK), melaksanakan proses pembelajaran/layanan konseling/layanan TIK, implementasi PTK/PTBK, melaksanakan penilaian, pembimbingan, dan kegiatan persekolahan lainnya.

Rambu-rambu pelaksanaa PKM adalah sebagai berikut:

- 1). PKM dilaksanakan di sekolah tempat guru bertugas.
- 2). Beban belajar PKM 14 SKS dengan durasi waktu 2 bulan, dengan ekivalen waktu 10 jam per hari.
- 3). Supervisi dilakukan sebanyak 2 (dua) kali oleh guru inti atau pengawas/kepala sekolah yang ditunjuk.
- 4). Peserta PKM wajib melaksanakan dan membuat laporan PTK/PTBK sesuai dengan format dan waktu yang ditentukan dan disahkan oleh kepala sekolah dan dipublikasikan di perpustakaan/ruang baca sekolah.
- 5). Uji Kinerja dilaksanakan di akhir PKM oleh Asesor LPTK Penyelenggara dan guru inti (supervisor setempat), peserta wajib menyerahkan perangkat pembelajaran (RPP/RPPBK) yang akan dipraktikkan pada saat uji kinerja.
- 6). Peserta yang belum lulus ujian kinerja, diberikan kesempatan menempuh ujian ulang maksimum 2 (dua) kali.

- 7). Uji kinerja dilaksanakan di sekolah *cluster* dan penetapannya disesuaikan dengan kondisi geografis setempat dan/atau disesuaikan dengan KKG dan MGMP.
  - 8). Ujian Tulis Nasional (UTN) dilaksanakan secara *on-line* dan untuk daerah tertentu secara *off-line*.
7. Peserta sertifikasi guru melalui PPGJ yang lulus uji kinerja dan UTN akan memperoleh sertifikat pendidik, sedangkan peserta yang belum lulus, diberi kesempatan mengulang sebanyak 2 (dua) kali untuk ujian yang belum memenuhi syarat kelulusan. Bagi peserta yang tidak lulus pada ujian ulang kedua, peserta dikembalikan ke dinas pendidikan provinsi/kabupaten/kota untuk memperoleh pembinaan dan dapat diusulkan mengikuti PKM tahun berikutnya.

#### **II.2.5. Komponen-Komponen Penilaian Portofolio**

Berikut ini adalah sepuluh komponen penilaian portofolio:

1. Kualifikasi akademik.
2. Pendidikan dan pelatihan.
3. Pengalaman mengajar.
4. Perencanaan dan pelaksanaan pembelajaran.
5. Penilaian dari atasan dan pengawas.
6. Prestasi akademik.
7. Karya pengembangan profesi.
8. Keikutsertaan dalam forum ilmiah.

9. Pengalaman organisasi dibidang kependidikan dan social.

10. Penghargaan yang relevan dengan bidang pendidikan.

Penilaian portofolio peserta sertifikasi guru dilakukan oleh Perguruan Tinggi Penyelenggara Sertifikasi/Lembaga Pendidikan Tenaga Kependidikan (LPTK) penyelenggara sertifikasi guru dalam bentuk Rayon yang terdiri atas LPTK Induk (LPTK Perguruan Tinggi Negeri) dan LPTK Mitra (LPTK Mitra yaitu Perguruan Tinggi Negeri atau Perguruan Tinggi Swasta yang di perbantukan di LPTK Induk) yang dikoordinasikan oleh konsorsium sertifikasi guru (KSG). Unsur KSG ini terdiri atas LPTK, Direktorat Jendral Pendidikan Tinggi (Ditjendikti), dan Direktorat Jendral Pendidikan Tinggi Peningkatan Mutu Pendidik dan Tenaga Kependidikan (Ditjen PMPTK) (I Gusti Made Murjana, 2015 : 68).

#### **II.2.6. Faktor Pendukung Program Sertifikasi Guru Dalam Jabatan**

1. Kebijakan Sertifikasi Guru dalam Jabatan Meningkatkan Motivasi Guru.

Salah satu dampak dari program sertifikasi guru dalam jabatan adalah untuk meningkatkan kesejahteraan guru, seperti yang tertuang dalam Undang-Undang Republik nomor 14 tahun 2005 tentang Guru dan Dosen pasal 16 menyatakan bahwa guru yang memiliki sertifikat pendidik memperoleh tunjangan profesi sebesar satu kali gaji pokok, baik guru pegawai negeri sipil maupun swasta dibayar oleh pemerintah. Meningkatkan kesejahteraan guru merupakan dampak dari sertifikasi

guru dan bukanlah merupakan tujuan yang utama dari sertifikasi guru dalam jabatan, tetapi ini diharapkan menjadi motivasi bagi guru, Seperti pernyataan dari guru Sekolah Menengah Pertama (SMP) di Kabupaten Kubu Raya bahwa “motivasi saya mengikuti sertifikasi guru dalam jabatan adalah untuk meningkatkan kesejahteraan dan supaya memperoleh pengakuan sebagai guru yang profesional”.

Muncul kesadaran guru untuk meningkatkan dan mengembangkan diri menuju guru yang profesional. Hal ini diungkapkan oleh Pengawas Sekolah Menengah Pertama(SMP) Kabupaten Kubu Raya (Hasil Wawancara : 4 Juli 2012) menyatakan sebagai berikut :

“Dengan adanya kebijakan program sertifikasi guru dalam jabatan, guru Sekolah Menengah Pertama (SMP) Negeri di Kabupaten Kubu Raya mempunyai keinginan untuk lebih giat mengajar, membuat perangkat dan analisis mengajar menjadi lebih lengkap, lebih disiplin dalam mengajar, lebih giat menambah pengetahuan, mengajar 24 jam pelajaran seminggu, yang belum memenuhi standar mau melanjutkan kuliah ke (S1). Adanya sertifikasi guru dapat memberikan dampak positif terhadap peningkatan profesionalisme guru”.

## 2. Meningkatkan Kesejahteraan dan Martabat Guru.

Dengan bekal pendidikan formal dan adanya berbagai kegiatan yang diikuti oleh guru, yang ditunjukkan dari dokumentasi data yang dikumpulkan dalam proses sertifikasi, guru dapat mentransfer lebih banyak ilmu yang dimiliki siswa. Jika guru telah memperoleh pengakuan

sebagai guru yang profesional, dengan dibuktikan adanya “sertifikat pendidik” dan telah dihargai dengan diberi tunjangan profesi satu kali gaji pokok, secara psikologis kondisi itu dapat meningkatkan harkat dan martabat guru dalam kehidupan bermasyarakat, berbangsa dan bernegara.

### 3. Guru Mendukung Program Kebijakan Sertifikasi Guru dalam Jabatan.

Para guru Sekolah Menengah Pertama (SMP) Negeri di Kabupaten Kubu Raya sangat antusias menyambut dengan baik adanya kebijakan program sertifikasi guru dalam jabatan. Hal ini diungkapkan oleh salah satu guru yang sudah bersrtifikasi di Kabupaten Kubu Raya (Hasil wawancara : 24 Mei 2012) menyatakan sebagai berikut :

“Dengan adanya kebijakan program sertifikasi guru dalam jabatan, Guru Sekolah Menengah Pertama (SMP) sangat mendukung program ini. Kebijakan program sertifikasi guru dalam jabatan yang merupakan program pemerintah pusat sudah lama ditunggu implementasinya oleh guru di seluruh Indonesia, karena salah satu dampak dari program sertifikasi guru dalam jabatan yaitu dapat menambah kesejahteraan dan penghasilan setiap bulannya sebesar satu kali gaji pokok”.

Akibat adanya program kebijakan sertifikasi, pemerintah mau tidak mau menghargai guru dengan memberikan tunjangan profesi dan guru merasa dihargai. Sekarang ini masih ada guru Sekolah Menengah Pertama (SMP) negeri di Kabupaten Kubu Raya yang belum memenuhi syarat untuk mengikuti program sertifikasi guru dalam jabatan, sehingga tidak dapat tunjangan profesi. Meskipun demikian guru tetap menyambut

baik dan mendukung adanya kebijakan program sertifikasi guru dalam jabatan, sebab mereka mempunyai harapan bahwa masa mendatang ketika telah memenuhi syarat, juga akan mendapatkan tunjangan profesi (Widiyaka, 2013 : 13-14).

### **II.3. Metode TOPSIS**

Technique for Order Performance by Similarity to Ideal Solution (TOPSIS) adalah salah satu metode pengambilan keputusan multikriteria yang pertama kali diperkenalkan oleh Yoon dan Hwang (1981). TOPSIS menggunakan prinsip bahwa alternatif yang terpilih harus mempunyai jarak terdekat dari solusi ideal positif dan jarak terpanjang (terjauh) dari solusi ideal negative dari sudut pandang geometris dengan menggunakan jarak Euclidean (jarak antara dua titik) untuk menentukan kedekatan relatif dari suatu alternatif dengan solusi optimal.

Solusi ideal positif didefinisikan sebagai jumlah dari seluruh nilai terbaik yang dapat dicapai untuk setiap atribut, sedangkan solusi negatif-ideal terdiri dari seluruh nilai terburuk yang dicapai untuk setiap atribut. TOPSIS mempertimbangkan keduanya, jarak terhadap solusi ideal positif dan jarak terhadap solusi ideal negatif dengan mengambil kedekatan relatif terhadap solusi ideal positif (Siti Kholijah Ritonga, 2013 : 143).

TOPSIS (Technique For Others Reference By Similarity To Ideal Solution) 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 negative (Bunga Annete Benning, 2015 : 266).

Secara umum, prosedur TOPSIS mengikuti langkah-langkah sebagai berikut:

- a. Membuat matriks keputusan yang ternormalisasi
- b. Membuat matriks keputusan ternormalisasi yang terbobot
- c. Menentukan matriks solusi ideal positif dan matriks solusi ideal negatif
- d. Menentukan jarak antara nilai setiap alternatif dengan matriks solusi ideal positif dan matriks solusi ideal negatif
- e. Menentukan nilai preferensi untuk setiap alternatif

TOPSIS membutuhkan rating kinerja setiap alternatif  $A_i$  pada setiap kriteria  $C_j$  yang ternormalisasi, yaitu:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \dots \dots \dots (1)$$

$i=1,2,\dots,m$ ; dan  $j=1,2,\dots,n$ .

Ket :

$r_{ij}$  = matriks ternormalisasi  $[i][j]$

$x_{ij}$  = matriks keputusan  $[i][j]$

Solusi ideal positif  $A^+$  dan solusi ideal  $A^-$  dapat ditentukan berdasarkan rating bobot ternormalisasi ( $y_{ij}$ ) sebagai :

$$y_{ij} = W_i r_{ij} \dots \dots \dots (2)$$

dengan  $i=1,2,\dots,m$ ; dan  $j=1,2,\dots,n$ .

$$A^+ = (y_1^+, y_2^+, \dots, y_n^+) \dots \dots \dots (3)$$

$$A^- = (y_1^-, y_2^-, \dots, y_n^-) \dots \dots \dots (4)$$

Ket :

$$\text{Dengan } Y_j^+ = \begin{cases} \max y_{ij} & \text{jika } j \text{ adalah atribut keuntungan} \\ \min y_{ij} & \text{jika } j \text{ adalah atribut biaya} \end{cases}$$

$$Y_j^- = \begin{cases} \min y_{ij} & \text{jika } j \text{ adalah atribut keuntungan} \\ \max y_{ij} & \text{jika } j \text{ adalah atribut biaya} \end{cases}$$

Jarak antara alternatif  $A_i$  dengan solusi ideal positif dirumuskan sebagai :

$$D_i^+ = \sqrt{\sum_{j=1}^n (y_{ij}^+ - y_{ij})^2}; \quad i=1,2,\dots,m \quad (5)$$

Jarak antara alternatif  $A_i$  dengan solusi ideal negative dirumuskan sebagai berikut :

$$D_i^- = \sqrt{\sum_{j=1}^n (y_{ij} - y_{ij}^-)^2}; \quad i=1,2,\dots,m \quad (6)$$

Ket :

$D_i^+$  = jarak alternatif  $A_i$  dengan solusi ideal positif

$D_i^-$  = jarak alternatif  $A_i$  dengan solusi ideal negatif

$y_i^+$  = solusi ideal positif [i]

$y_i^-$  = solusi ideal negatif

$y_{ij}$  = matriks normalisasi terbobot [i] [j]

Nilai preferensi untuk setiap alternatif ( $V_i$ ) diberikan sebagai:

$$V_i = \frac{D_i^-}{D_i^- + D_i^+}; \quad i=1,2,\dots,m \quad (7)$$

Ket :  $V_i$  = kedekatan tiap alternatif terhadap solusi ideal

$D_i^+$  = jarak alternatif  $A_i$  dengan solusi ideal positif

$D_i^-$  = jarak alternatif  $A_i$  dengan solusi ideal negative

Nilai  $V_i$  yang lebih besar menunjukkan bahwa alternatif  $A_i$  lebih dipilih (Bunga Annete Benning, 2015 : 3).

#### II.4. Basis Data (*Database*)

Database sering didefinisikan sebagai kumpulan data yang terkait. Secara teknis, yang berada dalam sebuah database adalah sekumpulan tabel atau objek lain (indeks, *view*, dan lain-lain). Tujuan utama pembuatan database adalah untuk memudahkan dalam mengakses data (Eka Choliviana dan Lies Yulianto, 2013).

Basis data dapat dipahami sebagai suatu kumpulan data terhubung (*interrelated data*) yang disimpan secara bersama-sama pada suatu media, tanpa *mengatap* satu sama lain atau tidak perlu suatu kerangkapan data (kalaupun ada maka kerangkapan data tersebut harus seminimal mungkin dan terkontrol [*controlled redundancy*]), data disimpan dengan cara-cara tertentu sehingga mudah digunakan/atau ditampilkan kembali; data dapat digunakan oleh satu atau lebih program-program aplikasi secara optimal; data disimpan tanpa mengalami ketergantungan dengan program yang akan menggunakannya; data disimpan sedemikian rupa sehingga proses penambahan, pengambilan, dan modifikasi data dapat dilakukan dengan mudah dan terkontrol (Edhy Sutanta, 2011 : 29-30).

Berdasarkan tingkat kompleksitas nilai data, tingkatan data dapat disusun dalam sebuah hierarki, mulai dari yang paling sederhana hingga paling sederhana hingga paling kompleks (Edhy Sutanta, 2011 : 35-36).

1. Sistem basis data, merupakan sekumpulan subsistem yang terdiri atas basis data dengan para pemakai yang menggunakan basis data secara bersama-sama, personal-personal yang merancang dan mengelola basis data, teknik-teknik untuk merancang dan mengelola basis data, serta sistem komputer untuk mendukungnya.
2. Basis data, merupakan sekumpulan dari bermacam-macam tipe *record* yang memiliki hubungan antar-*record* dan rincian data terhadap obyek tertentu.
3. *File*, merupakan sekumpulan *record* sejenis secara relasi yang tersimpan dalam media penyimpanan sekunder.
4. *Record*, merupakan *field*/atribut/data item yang saling berhubungan terhadap obyek tertentu.
5. *Data item/field*/atribut, merupakan unit terkecil yang disebut data, sekumpulan *byte* yang mempunyai makna.
6. *Data aggregate*, merupakan sekumpulan data item/*field*/atribut dengan ciri tertentu dan diberi nama.
7. *Byte*, adalah bagian terkecil yang dialamatkan dalam memori. *Byte* merupakan sekumpulan *bit* yang secara konvensional terdiri atas kombinasi 8 *bit* biner yang menyatakan sebuah karakter dalam memori (1 *byte* = 1 karakter).
8. *Bit*, adalah sistem biner yang terdiri atas dua macam nilai, yaitu 0 dan 1. Sistem biner merupakan dasar yang dapat digunakan untuk komunikasi antara manusia dan mesin (komputer).

## II.5. Normalisasi

Normalisasi diartikan sebagai suatu teknik yang menstrukturkan/mendekomposisi data dalam cara-cara tertentu untuk mencegah timbulnya permasalahan pengolahan data dalam basis data. Permasalahan yang dimaksud adalah berkaitan dengan penyimpangan-penyimpangan (*anomalies*) yang terjadi akibat adanya kerangkapan data dalam relasi dan in-efisiensi pengolahan (Edhy Sutanta ; 2011 : 174).

Secara berturut-turut masing-masing level normal tersebut dibahas berikut ini, dimulai dari bentuk tidak normal. (Edhy Sutanta ; 2011 : 176-179)

### 1. Relasi bentuk tidak normal (*Un Normalized Form* / UNF)

Relasi-relasi yang dirancang tanpa mengindahkan batasan dalam defisi basis data dan karakteristik *Relational Database Management System* (RDBM) menghasilkan relasi *Un Normalized Form* (UNF). Bentuk ini harus di hindari dalam perancangan relasi dalam basis data. Relasi *Un Normalized Form* (UNF) mempunyai kriteria sebagai berikut :

- a. Jika relasi mempunyai bentuk *non flat file* (dapat terjadi akibat data disimpan sesuai dengan kedatangannya, tidak memiliki struktur tertentu, terjadi duplikasi atau tidak lengkap)
- b. Jika relasi membuat *set atribut* berulang (*non single values*)
- c. Jika relasi membuat *atribut non atomic value*

### 2. Relasi bentuk normal pertama (*First Norm Form* / 1NF)

Relasi disebut juga *First Norm Form* (1NF) jika memenuhi kriteria sebagai berikut.

- a. Jika seluruh atribut dalam relasi bernilai *atomic* (*atomic value*)
- b. Jika seluruh atribut dalam relasi bernilai tunggal (*single value*)
- c. Jika relasi tidak memuat set atribut berulang
- d. Jika semua record mempunyai sejumlah atribut yang sama.

Permasalahan dalam *First Normal Form* (1NF) adalah sebagai berikut.

- a. Tidak dapat menyisipkan informasi parsial
  - b. Terhapusnya informasi ketika menghapus sebuah *record*
3. Bentuk normal kedua (*Second Normal Form* / 2NF)

Relasi disebut sebagai *Second Normal Form* (2NF) jika memenuhi kriteria sebagai berikut :

- a. Jika memenuhi kriteria *First Normal Form* (1NF)
- b. Jika semua atribut nonkunci *Functional Dependence* (FD) pada *Primary Key* (PK)

Permasalahan dalam *Second Normal Form* / 2NF adalah sebagai berikut:

- a. Kerangkapan data (*data redundancy*)
- b. Pembaharuan yang tidak benar dapat menimbulkan inkonsistensi data (*data inconsistency*)
- c. Proses pembaharuan data tidak efisien

Kriteria tersebut mengidentifikasi bahwa antara atribut dalam *Second Normal Form* masih mungkin mengalami *Third Normal Form*.

Selain itu, relasi *Second Normal Form* (2NF) menuntut telah didefinisikan atribut *Primary Key* (PK) dalam relasi. Mengubah relasi

*First Normal Form* (1NF) menjadi bentuk *Second Normal Form* (2NF) dapat dilakukan dengan mengubah struktur relasi dengan cara :

- a. Identifikasikan *Functional Dependence* (FD) relasi *First Normal Form* (1NF) .
  - b. Berdasarkan informasi tersebut, dekomposisi relasi *First Normal Form* (1NF) menjadi relasi-relasi baru sesuai *Functional Dependence* nya. Jika menggunakan diagram maka simpul-simpul yang berada pada puncak diagram ketergantungan data bertindak *Primary Key* (PK) pada relasi baru.
4. Bentuk normal ketiga (*Third Normal Form* / 3NF)

Suatu relasi disebut sebagai *Third Normal Form* jika memenuhi kriteria sebagai berikut :

- a. Jika memenuhi kriteria *Second Normal Form* (2NF)
- b. Jika setiap atribut nonkunci tidak (*TDF*) (*Non Transitive Dependency*) terhadap *Primary Key* (PK)

Permasalahan dalam *Third Normal Form* (3NF) adalah keberadaan penentu yang tidak merupakan bagian dari *Primary Key* (PK) menghasilkan duplikasi rinci data pada atribut yang berfungsi sebagai *Foreign Key* (FK) (duplikasi berbeda dengan keterangan data).

Mengubah relasi *Second Normal Form* (2NF) menjadi bentuk *Third Normal Form* (3NF) dapat dilakukan dengan mengubah struktur relasi dengan cara :

- a. Identifikasi TDF relasi *Second Normal Form* (2NF)

- b. Berdasarkan informasi tersebut, dekomposisi relasi *Second Normal Form* (2NF) menjadi relasi-relasi baru sesuai TDF-nya.
5. Bentuk normal *Boyce-Codd* (*Boyce-Codd Norm Form* / BCNF)  
Bentuk normal *Boyce-Codd Norm Form* (BCNF) dikemukakan oleh R.F. Boyce dan E.F. Codd. Suatu relasi disebut sebagai *Boyce-Codd Norm Form* (BCNF) jika memenuhi kriteria sebagai berikut.
  - a. Jika memenuhi kriteria *Third Norm Form* (3NF)
  - b. Jika semua atribut penentu (determinan) merupakan CK
6. Bentuk normal keempat (*Forth Norm Form* / 4NF)  
Relasi disebut sebagai *Forth Norm Form* (4NF) jika memenuhi kriteria sebagai berikut.
  - a. Jika memenuhi kriteria *Boyce-Codd Norm Form*.
  - b. Jika setiap atribut didalamnya tidak mengalami ketergantungan pada banyak nilai.
7. Bentuk normal kelima (*Fifth Norm Form* / 5NF)  
Suatu relasi memenuhi kriteria *Fifth Norm Form* (5NF) jika kerelasian antar data dalam relasi tersebut tidak dapat direkonstruksi dari struktur relasi yang sederhana.
8. Bentuk normal kunci domain (*Domain Key Norm Form* / DKNF)
  - a. Relasi disebut sebagai *Domain Key Norm Form* (DKNF) jika setiap batasan dapat disimpulkan secara sederhana dengan mengetahui sekumpulan nama atribut dan domainnya selama menggunakan sekumpulan atribut pada kuncinya.

## **II.6. UML (*Unified Model Language*)**

*Unified Modelling Language* (UML) adalah sebuah “bahasa” yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah system. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT (Object Modeling Technique), dan Ivar Jacobson OOSE (Object-Oriented Software Engineering) (Satriawaty Mallu, 2015 : 38).

*Unified Modelling Language* (UML) merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek. UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan bahasa pemodelan visual yang memungkinkan pengembang sistem membuat *blue print* atas visinya dalam bentuk yang baku. UML berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui jumlah elemen grafis yang bisa dikombinasikan menjadi diagram (Rosana Junita Sirait, et al., 2015).

### **II.6.1. Fungsi *Unified Modelling Language* (UML)**

*Unified Modeling Language* (UML) biasa digunakan untuk (Aris, et al., 2015).

- a. Menggambarkan batasan sistem dan fungsi -fungsi sistem secara umum, dibuat dengan *use case* dan *actor*.
- b. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.
- c. Menggambarkan representasi struktur *static* sebuah sistem dalam bentuk *class diagrams*.
- d. Membuat model behavior “yang menggambarkan kebiasaan atau sifat sebuah sistem” dengan *state transition diagrams*.
- e. Menyatakan arsitektur implementasi fisik menggunakan *component and development*.
- f. Menyampaikan atau memperluas *functionality* dengan *stereotypes*.

UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek. Karena UML menyediakan bahasa pemodelan visual yang memungkinkan pengembang sistem membuat *blue print* atas visinya dalam bentuk yang baku. UML berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui sejumlah elemen grafis yang bisa dikombinasikan menjadi diagram. UML mempunyai banyak diagram yang dapat mengakomodasikan berbagai sudut pandang dari suatu perangkat lunak yang akan dibangun. Diagram-diagram tersebut digunakan untuk :

- a. Mengkomunikasikan ide.
- b. Melahirkan ide-ide baru dan peluang-peluang baru.
- c. Menguji ide dan membuat prediksi.
- d. Memahami struktur dan relasi-relasinya.

## II.6.2. Diagram-Diagram *Unified Modelling Language* (UML)

Adapun jenis-jenis dari diagram UML adalah sebagai berikut :

### 1. *Use Case Diagram*

Suatu *use case* diagram menampilkan sekumpulan *use case* dan aktor (pelaku) dan hubungan diantara *use case* dan aktor tersebut. *Use case* diagram digunakan untuk penggambaran *use case* statik dari suatu sistem. *Use case* diagram penting dalam mengatur dan memodelkan kelakuan dari suatu sistem. *Use case* menjelaskan apa yang dilakukan sistem (atau subsistem) tetapi tidak menspesifikasi cara kerjanya. *Flow of event* digunakan untuk menspesifikasi cara kerjanya kelakuan dari *use case*. *Flow of event* menjelaskan *use case* dalam bentuk tulisan dengan sejelass-jelasnya, diantaranya bagaimana, kapan *use case* dimulai dan berakhir, ketika *use case* berinteraksi dengan aktor, objek apa yang digunakan, alur dasar dan alur alternatif. Terdapat beberapa simbol dalam menggambarkan diagram *use case*, yaitu *use cases*, aktor dan relasi (Achmad Hamzah Nasrullah dan Dadang Sudrajat, 2015 : 6).

Pembuatan *use case* diagram merupakan tahap awal dalam pembangunan sistem, tahapan ini mempresentasikan interaksi antara aktor dengan sistem (Bunga Annete Benning, 2015 : 4).


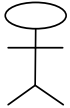

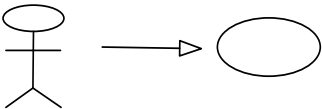
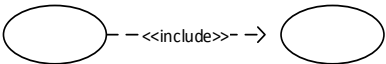
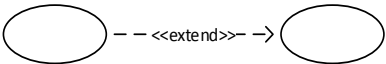
Berbagai simbol yang hadir didalam *use case* diagram antara lain adalah :

- a. *Use Case*, untuk mengetahui action atau prosedur apa yang ada didalam sistem.

- b. *Actor*, siapa saja yang terlibat dalam action tersebut.
- c. *Relationship*, bagaimana actions saling berelasi satu sama lain didalam sistem (Edgar Winata dan Johan Setiawan, 2013).

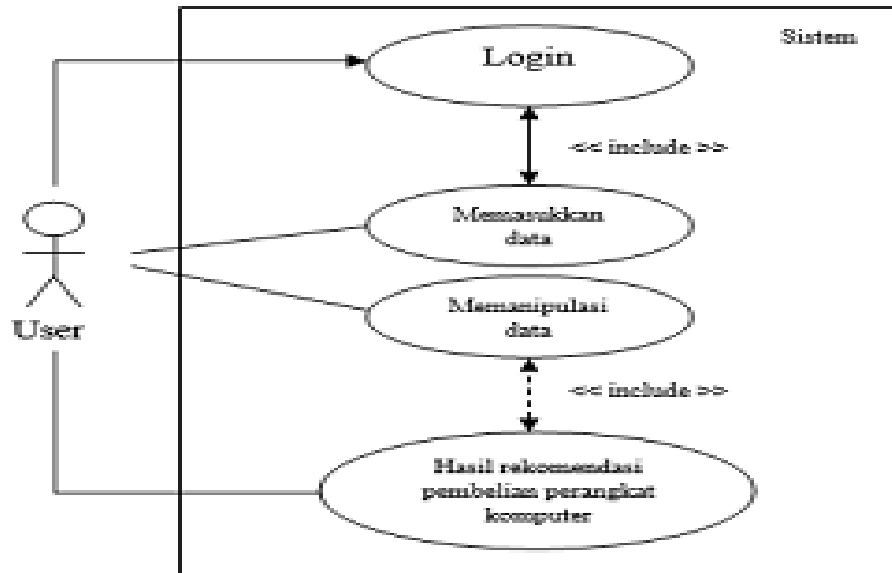
Adapun keterangan terhadap notasi / symbol dari gambar dibawah dapat dilihat pada tabel. II.1.

**Tabel II.1 . Simbol *Use Case Diagram***

Simbol	Keterangan
	<b>Use Case:</b> menggambarkan bagaimana seseorang akan menggunakan / memanfaatkan sistem
	<b>Aktor:</b> seseorang / sesuatu yang berinteraksi dengan sisitem yang sedang kita kembangkan
	<b>Relasi:</b> sebagai penghubung antara aktorusecase, usecase-usecase dll.
	<b>Relasi Asosiasi:</b> relasi terjadi antara aktor dengan usecase biasanya berupa garis lurus dengan kepala panah dis alah satu ujung nya
	<b>Include Relationship (relasi cakupan):</b> memungkinkan suatu usecase untuk menggunakan fungsionalitas yang disediakan oleh usecase yang lainnya.
	<b>Extend Relationship:</b> memungkinkan usecase memiliki kemungkinan untuk memperluas fungsionalitas yang disediakan oleh usecase yang lainnya.

(Sumber : Oktafiansyah, 2012 : 5)

Adapun *Use Case* diagram pada sistem ini dapat dilihat pada gambar II.2.



**Gambar II.2. Use Case Diagram SPK Pembelian Perangkat Komputer**  
(Sumber : Bunga Annete Benning, 2015 : 4)

## 2. Activity Diagram

*Activity* diagram dibuat setelah *use case* diagram telah terbentuk. *Activity* diagram berguna untuk menggambarkan alir aktifitas yang dilakukan oleh *user* dalam sistem yang sedang dirancang (Bunga Annete Benning, 2015 : 4).






Teknik untuk menjelaskan *business process*, *procedural logic*, dan *work flow*. Bisa dipakai untuk menjelaskan teks *use case* dalam notasi grafis dengan menggunakan notasi yang mirip *flow chart*, meskipun terdapat sedikit perbedaan notasi (Edgar Winata dan Johan Setiawan, 2013).

Berbagai simbol yang hadir didalam *use case* diagram antara lain adalah :

- a. *Nodes*, menandakan initial dan *final node*, *final node* boleh lebih dari 1.
- b. *Activity*, aktivitas sistem dapat berupa aktivitas fisik juga bagi *user*.
- c. *Flow/edge*, arah sebuah proses.
- d. *Fork*, awal sebuah proses paralel.
- e. *Join* akhir proses paralel.
- f. *Condition*, kondisi yang dituliskan dalam bentuk teks
- g. *Decision*, implementasi *if* dan *then*.
- h. *Merge*, penyatuan beberapa *flow*.
- i. *Partition*, siapa atau apa yang menjalankan aktivitas (Edgar Winata dan Johan Setiawan, 2013).

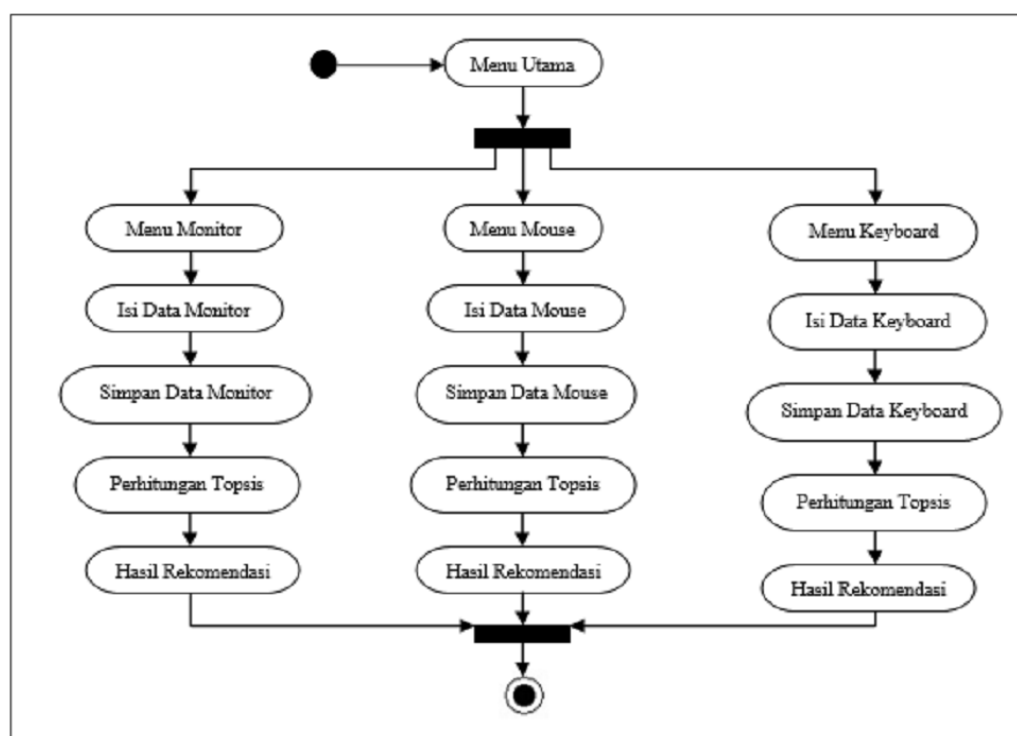
Adapun keterangan terhadap notasi / symbol dari gambar dibawah dapat dilihat pada tabel. II.2.

**Tabel II.2. Simbol *Activity Diagram***

NSymbol	Keterangan
	Titik Awal
	Titik Akhir
	Activity
	Gari tidak putus. Urutan dari suatu kejadian atau aktivitas ke yang berikutnya.
	Fork; digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.

(Sumber : Oktafiansyah, 2012 : 5)

Adapun *Activity* Diagram sistem ini dapat dilihat pada gambar II.3.



Gambar II.3. *Activity* Diagram SPK Pembelian Perangkat Komputer

(Sumber : Bunga Annete Benning, 2015 : 4)

### 3. *Sequence Diagram*

Menjelaskan interaksi obyek-obyek yang saling berkolaborasi (berhubungan), mirip dengan *activity* diagram yaitu menggambarkan alur kejadian sebuah aktivitas tetapi lebih detil dalam menggambarkan aliran data termasuk data atau behaviour yang dikirimkan atau diterima namun kurang mampu menjelaskan detil dari sebuah algoritma (Edgar Winata dan Johan Setiawan, 2013).


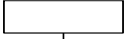

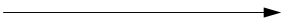

*Sequence* diagram berfungsi untuk menggambarkan interaksi antara class yang ada dan sampai ketahapan apa yang dihasilkan oleh system (Bunga Annete Bening, 2015 : 4).

Berbagai simbol yang hadir didalam *sequence* diagram antara lain adalah :

- a. *Participant*, yaitu objek yang terkait dengan sebuah urutan proses.
- b. *Lifeline*, menggambarkan daur hidup sebuah objek.
- c. *Activation*, suatu titik waktu dimana sebuah objek mulai berpartisipasi dalam sebuah sequence.
- d. *Time*, elemen paling penting dalam sequence diagram yang konteksnya adalah urutan, bukan durasi.
- e. *Return*, suatu hasil kembalian sebuah operasi. Operasi mengembalikan hasil tetapi boleh tidak ditulis jika tidak ada perbedaan dengan Getternya (Edgar Winata dan Johan Setiawan, 2013).

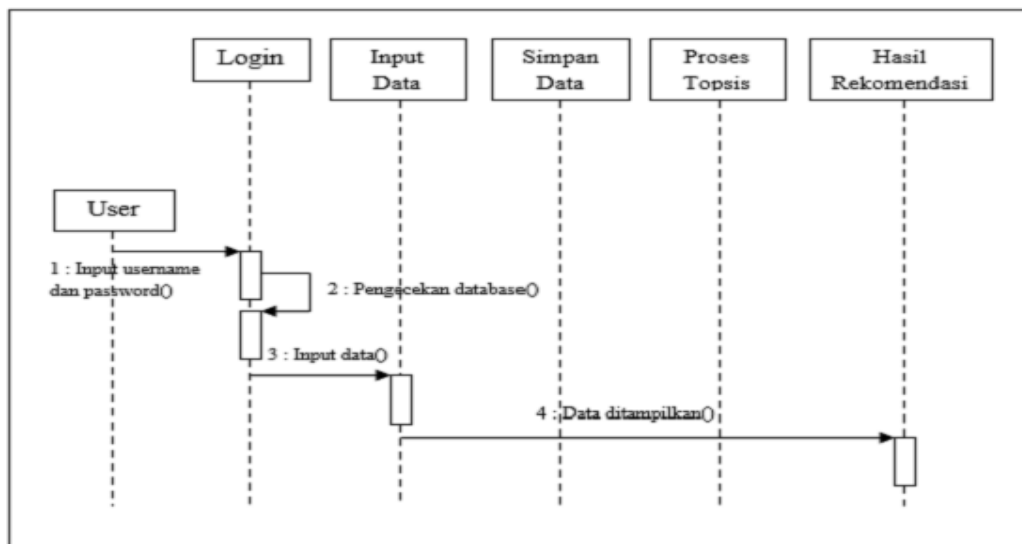
Adapun keterangan terhadap notasi / symbol dari gambar dibawah dapat dilihat pada tabel. II.3.

**Tabel II.3. Simbol *Sequence Diagram***

NSymbol	Keterangan
	<b>Aktor</b> : Seseorang/sesuatu yang berinteraksi dengan sistem yang sedang kita kembangkan
	<b>Objek</b> : Menambah objek baru pada diagram
	<b>Aktivasi</b> : Menggambarkan langkah-langkah dalam aliran kerja
	<b>Pesan</b> : Menggambarkan pesan antara dua objek.
	<b>Pengulangan</b> : menggambarkan pesan yang menuju dirinya sendiri

(Sumber : Oktafiansyah, 2012 : 6)

Adapun *Sequence* diagram pada sistem ini dapat terlihat pada gambar II.4.



**Gambar II.4. *Sequence Diagram* SPK Pembelian Perangkat Komputer**

(Sumber : Bunga Annete Bening, 2015 : 5)

#### 4. *Class Diagram*

*Class* diagram merupakan diagram paling umum yang dijumpai dalam pemodelan berbasis UML. Didalam *Class* diagram terdapat *class* dan *interface* beserta atribut-atribut dan operasinya, relasi yang terjadi antar objek, *constraint* terhadap objek-objek yang saling berhubungan dan *inheritance* untuk organisasi *class* yang lebih baik. *Class* diagram juga terdapat *static view* dari elemen pembangun sistem. Pada intinya *Class* diagram mampu membantu proses pembuatan sistem dengan memanfaatkan konsep *forward* ataupun *reverse engineering* (Edgar Winata dan Johan Setiawan, 2013).

*Class* diagram menggambarkan rancangan basis data pada sistem yang dibuat, rancangan inilah yang menjadi dasar pembuatan basis data pada sistem. Selain itu *class* diagram juga berguna untuk menggambarkan relasi antar tabel pada basis data (Sumber : Bunga Annete Bening, 2015 : 5).

Berbagai simbol yang hadir didalam *use case* diagram antara lain adalah :

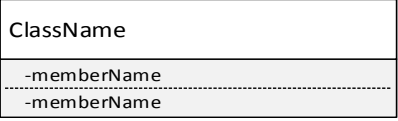

- a. *Class*, yang berfungsi untuk merepresentasikan tipe dari data yang dimilikinya. *Class* diagram dapat ditampilkan dengan menunjukkan atribut dan operasi yang dimilikinya atau hanya menunjukkan nama *class*-nya saja. Dapat juga kita tuliskan nama *class* dengan atributnya saja atau nama *class* dengan operasinya.
- b. *Attribute*, merupakan data yang terdapat didalam *class* dan *instance*-nya dengan operator.

- c. *Operation*, berfungsi untuk merepresentasikan fungsi-fungsi yang ditampilkan oleh *class* dan *instance*-nya dengan operator.
- d. *Association*, digunakan untuk menunjukkan bagaimana dua *class* berhubungan satu sama lainnya. *Association* ditunjukkan dengan sebuah garis yang terletak diantara dua *class*. Didalam setiap *association* terdapat *multiplicity*, yaitu simbol yang mengindikasikan berapa banyak *instance* dari *class* pada ujung *association* yang satu dengan *instance class* di ujung *association* lainnya.
- e. *Generalizations*, berfungsi untuk mengelompokkan *class* ke dalam hirarki *inheritance*.
- f. *Aggregation*, merupakan bentuk khusus dari *association* yang merepresentasikan hubungan “*part-whole*”. Bagian “*whole*” dari hubungan ini sering disebut dengan *assembly* atau *aggregate*. *Class* yang satu dapat dikatakan merupakan bagian dari *class* yang lain yang ikut membentuk *class* tersebut.
- g. *Composition*, merupakan jenis *aggregation* yang lebih kuat diantara dua *class* yang memiliki *association* dimana jika *whole* ditiadakan, maka *part*-nya juga ikut ditiadakan. Berbeda dengan *aggregation*, *part* akan tetap bisa berdiri sendiri meskipun bagian *whole*-nya ditiadakan.

- h. Penggunaan operator (+) dalam *class* diagram diartikan dengan *public*, operator (-) diartikan *private*, dan operator (#) diartikan *protected* (Edgar Winata dan Johan Setiawan, 2013).

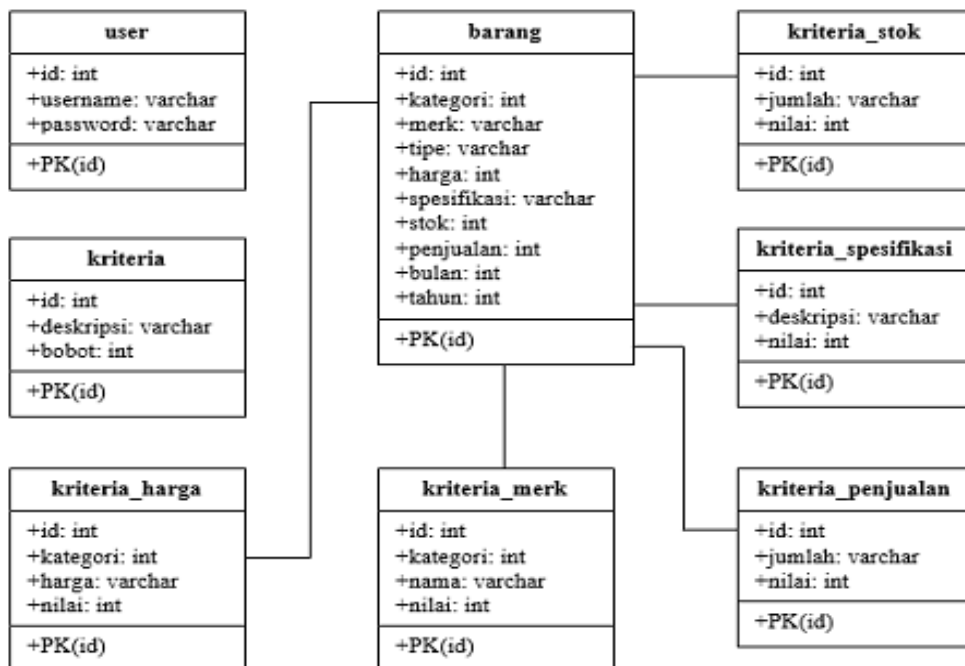
Adapun keterangan terhadap notasi / symbol dari gambar dibawah dapat dilihat pada tabel. II.4.

**Tabel II.4. Simbol *Class Diagram***

NSymbol	Keterangan
	<p><b>Class diagram; Notasi utama dan yang paling mendasar pada diagram UML adalah notasi untuk mempresentasikan suatu class beserta dengan atribut dan operasinya. Class adalah pembentuk utama dari sistem berorientasi objek.</b></p>
	<p><b>Association; Association menggambarkan navigasi antar class (navigation), berapa banyak obyek lain yang bisa berhubungan dengan satu obyek (multiplicity antar class) dan apakah suatu class menjadi bagian dari class lainnya (aggregation).</b></p>

(Sumber : HAVILUDDIN, 2011 : 6-7)

Adapun *Class* diagram sistem ini dapat dilihat pada gambar II.5.



## **Gambar II.5. Class Diagram SPK Pembelian Perangkat Komputer**

*(Sumber : Bunga Annete Bening, 2015 : 5)*

### **II.7. Microsoft Visual Basic 2010**

Pada akhir tahun 1999, Teknologi .NET diumumkan. Microsoft memosisikan teknologi tersebut sebagai *platform* untuk membangun XML Web Services. XML Web services memungkinkan aplikasi tipe manapun dan dapat mengambil data yang tersimpan pada server dengan tipe apapun melalui internet.

Visual Basic.NET adalah Visual Basic yang direkayasa kembali untuk digunakan pada *platform* .NET sehingga aplikasi yang dibuat menggunakan Visual Basic .NET dapat berjalan pada sistem komputer apa pun, dan dapat mengambil data dari server dengan tipe apa pun asalkan terinstal .NET Framework. (Priyanto Hidayatullah, 2012 ; 5).

### **II.8. SQL Server 2008**

SQL Server adalah sebuah *database* relasional yang dirancang untuk mendukung aplikasi dengan arsitektur *client/server* dimana *database* terdapat pada komputer pusat yang disebut *server*, dan informasi digunakan bersama-sama oleh beberapa *user* yang menjalankan aplikasi didalam komputer lokalnya yang disebut dengan *client*. Arsitektur semacam ini memberikan integritas data yang tinggi karena semua *user* bekerja dengan informasi yang sama. Melalui aturan-aturan bisnis, kendali diterapkan kepada semua *user* mengenai informasi yang ditambahkan ke dalam *database*. Database SQL Server dibagi kedalam beberapa

komponen logikal, seperti misalnya tabel, view, dan elemen–elemen lain yang terlihat oleh *user* (Rika Yunitarini, 2013).

Arsitektur jaringan *Client Server* merupakan model konektivitas pada jaringan yang membedakan fungsi komputer apakah sebagai *client* atau *server*. Arsitektur ini menempatkan sebuah komputer sebagai *server* yang bertugas memberikan layanan kepada terminal-terminal lain (*client*) yang terhubung dalam sistem jaringan itu. *Server* dapat bertugas untuk memberikan layanan berbagi pakai berkas (*file sever*), *printer* (*printer server*), jalur komunikasi (*server komunikasi*) (Oktafiansyah, 2012).

Pada dasarnya, Client Server dibentuk oleh tiga komponen dasar, yaitu (Oktafiansyah, 2012) :

a. Client

Client merupakan terminal yang digunakan oleh pengguna untuk meminta layanan tertentu yang dibutuhkan. Terminal client dapat berupa PC, ponsel, komunikator, robot, televisi, dan peralatan lain yang membutuhkan informasi.

b. Middleware

Middleware merupakan komponen perantara yang memungkinkan client dan server untuk saling terhubung dan berkomunikasi satu sama lain. Middleware ini dapat berupa Transaction Monitor (TP), Remote Prosedure Calling (RPC), atau Object Request Broker (ORB). Middleware memiliki peran yang sangat strategis karena dengan adanya middleware maka client dapat mengirimkan pesan atau permintaan kepada server,

menerjemahkan pesan dari client agar dapat dimengerti oleh server, menerjemahkan hasil proses dari server agar dapat dipahami oleh client dan mengirimkan hasil proses yang telah diterjemahkan kembali ke client.

c. Server

Server merupakan pihak yang menyediakan layanan. Server ini dapat berupa basis data SQL, Monitor TP, server groupware, server objek, atau Web.