

BAB II LANDASAN TEORI

2.1 Pengertian Sistem Pendukung Keputusan

Sistem Pendukung Keputusan merupakan salah satu produk perangkat lunak yang dikembangkan secara khusus untuk membantu dalam proses pengambilan keputusan. Sesuai dengan namanya, tujuan dari dipergunakannya sistem ini adalah sebagai “*second opinion*” atau “*information sources*” yang dapat dipakai sebagai bahan pertimbangan sebelum memutuskan kebijakan tertentu.

2.1.1 Proses Pengambilan Keputusan

Pengambilan keputusan adalah pemilihan beberapa tindakan alternatif yang ada untuk mencapai satu atau beberapa tujuan yang telah ditetapkan (Turban, 2005). Pengambilan keputusan meliputi beberapa tahap dan melalui beberapa proses. Proses pengambilan keputusan meliputi empat tahap yang saling berhubungan dan berurutan. Empat proses tersebut adalah (Turban, 2005):

a. *Intelligence*

Tahap ini merupakan proses penelusuran dan pendeteksian dari lingkup problematika serta proses pengenalan masalah. Data masukan diperoleh, diproses, dan diuji dalam rangka mengidentifikasi masalah.

b. *Design*

Tahap ini merupakan proses menemukan dan mengembangkan alternatif. Tahap ini meliputi proses untuk mengerti masalah, menurunkan solusi dan menguji kelayakan solusi.

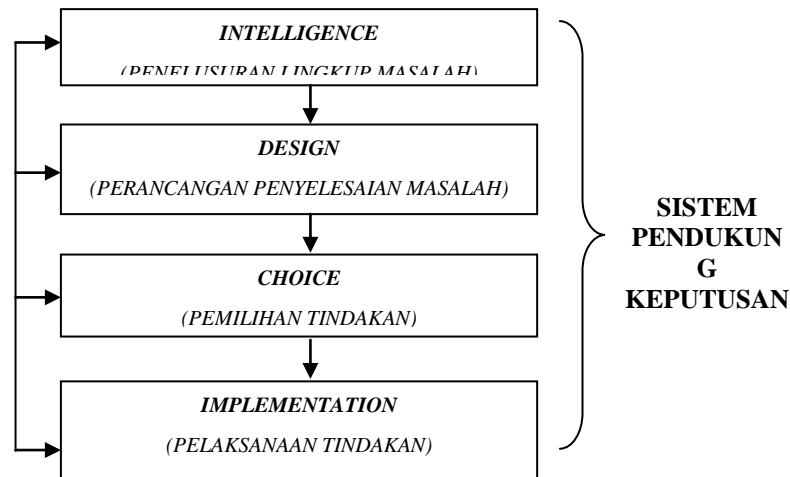
c. *Choice*

Pada tahap ini dilakukan proses pemilihan di antara berbagai alternatif tindakan yang mungkin dijalankan. Tahap ini meliputi pencarian, evaluasi, dan rekomendasi solusi yang sesuai untuk model yang telah dibuat. Solusi dari model merupakan nilai spesifik untuk variabel hasil pada alternatif yang dipilih.

d. *Implementation*

Tahap implementasi adalah tahap pelaksanaan dari keputusan yang telah diambil. Pada tahap ini perlu disusun serangkaian tindakan yang terencana, sehingga hasil keputusan dapat dipantau dan disesuaikan apabila diperlukan perbaikan.

Proses pengambilan keputusan, seperti terlihat pada gambar berikut.



Gambar 2.1 Fase Proses Pengambilan Keputusan

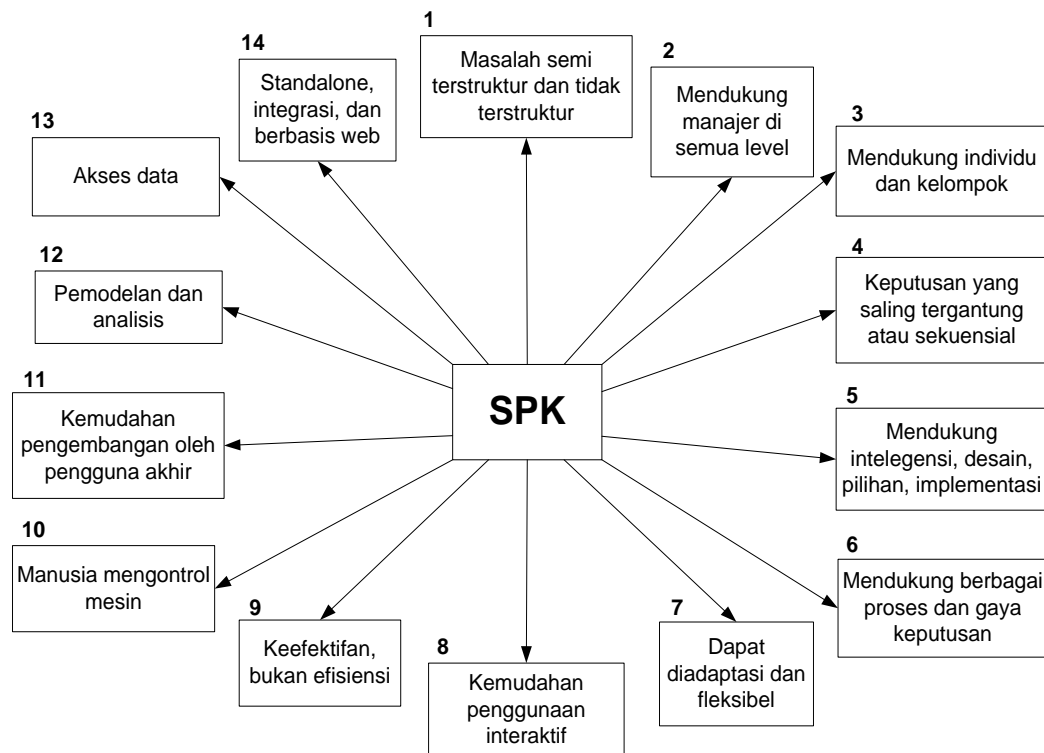
Sumber (Turban, 2005)

2.1.2 Karakteristik Sistem Pendukung Keputusan

Turban (2005) mengemukakan karakteristik dan kapabilitas kunci dari Sistem Pendukung Keputusan adalah sebagai berikut (Gambar 2.2):

1. Dukungan untuk pengambil keputusan, terutama pada situasi semi terstruktur dan tak terstruktur.
2. Dukungan untuk semua level manajerial, dari eksekutif puncak sampai manajer lini.
3. Dukungan untuk individu dan kelompok.
4. Dukungan untuk semua keputusan independen dan atau sekuensial.
5. Dukungan di semua fase proses pengambilan keputusan: inteligensi, desain, pilihan, dan implementasi.
6. Dukungan pada berbagai proses dan gaya pengambilan keputusan.

7. Kemampuan sistem beradaptasi dengan cepat dimana pengambil keputusan dapat menghadapi masalah-masalah baru dan pada saat yang sama dapat menanganinya dengan cara mengadaptasikan sistem terhadap kondisi-kondisi perubahan yang terjadi.
8. Pengguna merasa seperti di rumah. *User-friendly*, kapabilitas grafis yang kuat, dan sebuah bahasa interaktif yang alami.
9. Peningkatan terhadap keefektifan pengambilan keputusan (akurasi, *timelines*, kualitas) dari pada efisiensi (biaya).
10. Pengambil keputusan mengontrol penuh semua langkah proses pengambilan keputusan dalam memecahkan masalah.
11. Pengguna akhir dapat mengembangkan dan memodifikasi sistem sederhana.
12. Menggunakan model-model dalam penganalisan situasi pengambilan keputusan.
13. Disediakkannya akses untuk berbagai sumber data, format, dan tipe, mulai dari sistem informasi geografi (GIS) sampai sistem berorientasi objek.
14. Dapat dilakukan sebagai alat *standalone* yang digunakan oleh seorang pengambil keputusan pada satu lokasi atau didistribusikan di satu organisasi keseluruhan dan di beberapa organisasi sepanjang rantai persediaan.



Gambar 2.2 Karakteristik dan Kapabilitas SPK

Sumber (Turban, 2005)

2.1.3 Keuntungan Sistem Pendukung Keputusan

Keuntungan dari sistem pendukung keputusan sebagai berikut :

1. Dapat memperluas kemampuan seseorang untuk mengambil keputusan dalam memproses data atau informasi pemakainya.
2. Membantu mengambil 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. Dapat menjadi stimulan bagi pengambil keputusan dalam memahami permasalahannya, karena sistem pendukung keputusan mampu menyajikan berbagai alternatif.
5. Mampu menyediakan bukti tambahan untuk memberikan pembenaran, sehingga dapat memperluas posisi pengambilan keputusan.

Sistem Pendukung Keputusan terdiri atas 3 komponen atau subsistem, sebagai berikut:

1. Subsistem Data (*Data Subsystem*)

Subsistem data merupakan bagian yang menyeleksi data – data yang dibutuhkan oleh Base Management Subsystem (DBMS). DBMS sendiri merupakan subsistem data yang terorganisasi dalam suatu basis data. Data – data yang merupakan dalam suatu Sistem Pendukung Keputusan dapat berasal dari luar lingkungan. Keputusan pada manajemen level atas seringkali harus memanfaatkan data dan informasi yang bersumber dari luar perusahaan. Kemampuan subsistem data yang diperlukan dalam suatu Sistem Pendukung Keputusan, antara lain :

- a. Mampu mengkombinasikan sumber – sumber data yang relevan melalui proses ekstraksi data.
- b. Mampu menambah dan menghapus secara cepat dan mudah.
- c. Mampu menangani data personal dan non personal, sehingga user dapat bereksperimen dengan berbagai alternatif keputusan.
- d. Mampu mengolah data yang bervariasi dengan fungsi manajemen data yang luas.

2. Subsistem Model (*Model Subsystem*)

Subsistem model dalam Sistem Pendukung Keputusan memungkinkan pengambil keputusan menganalisa secara utuh dengan mengembangkan dan membandingkan alternatif solusi. Integrasi model – model dalam Sistem Informasi yang berdasarkan integrasi data – data dari lapangan menjadi suatu Sistem Pendukung Keputusan.

Kemampuan subsistem model dalam Sistem Pendukung Keputusan antara lain :

- a. Mampu menciptakan model – model baru dengan cepat dan mudah
- b. Mampu mengkatalogkan dan mengelola model untuk mendukung semua tingkat pemakai
- c. Mampu menghubungkan model – model dengan basis data melalui hubungan yang sesuai

d. Mampu mengelola basis model dengan fungsi manajemen yang analog dengan database manajemen

3. Subsistem Dialog (*User Sistem Interface*)

Subsistem dialog merupakan bagian dari Sistem Pendukung Keputusan yang dibangun untuk memenuhi kebutuhan representasi dan mekanisme kontrol selama proses analisa dalam Sistem Pendukung Keputusan ditentukan dari kemampuan berinteraksi antara sistem yang terpasang dengan user. Pemakai terminal dan sistem perangkat lunak merupakan komponen – komponen yang terlibat dalam subsistem dialog yang mewujudkan komunikasi antara user dengan sistem tersebut. Komponen dialog menampilkan keluaran sistem bagi pemakai dan menerima masukan dari pemakai ke dalam Sistem Pendukung Keputusan. Adapun subsistem dialog dibagi menjadi tiga, antara lain :

- a. Bahasa Aksi (*The Action Language*) Merupakan tindakan – tindakan yang dilakukan user dalam usaha untuk membangun komunikasi dengan sistem. Tindakan yang dilakukan oleh user untuk menjalankan dan mengontrol sistem tersebut tergantung rancangan sistem yang ada.
- b. Bahasa Tampilan (*The Display or Presentation Language*) Merupakan keluaran yang dihasilkan oleh suatu Sistem Pendukung Keputusan dalam bentuk tampilan – tampilan akan memudahkan user untuk mengetahui keluaran sistem terhadap masukan – masukan yang telah dilakukan.
- c. Bahasa Pengetahuan (*Knowledge Base Language*) Meliputi pengetahuan yang harus dimiliki user tentang keputusan dan tentang prosedur pemakaian Sistem Pendukung Keputusan agar sistem dapat digunakan secara efektif. Pemahaman user terhadap permasalahan yang dihadapi dilakukan diluar sistem, sebelum user menggunakan sistem untuk mengambil keputusan.

2.2 SAW (*Simple Additive Weighting*)

Metode SAW (Kusumadewi dan Purnomo, 2013) merupakan metode yang paling sederhana dan paling banyak digunakan. Metode ini juga metode yang paling mudah untuk diaplikasikan, karena mempunyai algoritma yang tidak terlalu rumit. Metode SAW sering juga dikenal sebagai metode penjumlahan terbobot. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut. Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada.

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\text{Max}_i x_{ij}} & \text{jika } j \text{ atribut keuntungan (benefit)} \\ \frac{\text{Min}_i x_{ij}}{x_{ij}} & \text{jika } j \text{ atribut biaya (cost)} \end{cases}$$

Dimana r_{ij} adalah rating kinerja ternormalisasi dari alternatif A_i pada atribut C_j ; $i=1,2,\dots,m$ dan $j=1,2,\dots,n$. Nilai preferensi untuk setiap alternatif (V_i) diberikan sebagai:

Beda antara atribut keuntungan dan atribut biaya yaitu : Dikatakan atribut keuntungan jika atribut yang diberikan itu dimaksudkan untuk meningkatkan keuntungan dari pengambilan keputusan yang diambil. Jika nilai kecocokan setiap kriteria itu semakin tinggi nilainya semakin baik atau semakin diprioritaskan maka kriteria tersebut dikatakan kriteria atau atribut keuntungan. Kemudian dikatakan atribut biaya jika atribut yang diberikan itu dimaksudkan untuk meningkatkan pengurangan biaya operasional pengambilan keputusan yang diambil. Jika nilai kecocokan setiap kriteria itu semakin kecil nilainya semakin baik, maka kriteria tersebut dikatakan kriteria biaya.

$$V_i = \sum_{j=1}^n w_j r_{ij}$$

Nilai V_i yang lebih besar mengindikasikan bahwa alternatif A_i lebih terpilih. (Kusumadewi dan Purnomo, 2013).

Hasil akhir diperoleh dari setiap proses perankingan yaitu penjumlahan dari perkalian matriks ternormalisasi dengan bobot preferensi sehingga diperoleh nilai V_i yang lebih besar mengindikasikan bahwa alternatif A_i merupakan alternatif terbaik.

2.2.1 Konsep Perhitungan Metode Simple Additive Weighting (SAW)

Terdapat beberapa langkah dalam menggunakan metode SAW untuk memecahkan masalah, langkah-langkahnya adalah sebagai berikut (Kusumadewi dan Purnomo, 2013) :

1. Menentukan alternatif, yaitu A_i .
2. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu C_i
3. Memberikan nilai bobot pada setiap kriteria.
4. Menentukan bobot preferensi atau tingkat kepentingan (W_j) setiap kriteria.
5. Membuat tabel rating kecocokan dari setiap alternatif pada setiap kriteria.
6. Membuat matriks keputusan berdasarkan kriteria (C_i).
7. Melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis kriteria, (kriteria keuntungan ataupun kriteria biaya) sehingga diperoleh matriks ternormalisasi R .
8. Hasil akhir diperoleh dari proses perankingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan vektor bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik (A_i) sebagai solusi.

2.3 Diagram Konteks

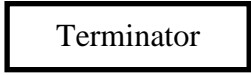


Diagram Konteks merupakan tingkatan tertinggi dalam diagram aliran data dan hanya memuat satu proses, menunjukkan sistem secara keseluruhan. Proses tersebut diberi nomor nol. Semua entitas eksternal yang ditunjukkan pada diagram

konteks berikut aliran data-aliran data utama menuju dan dari sistem. Diagram tersebut tidak memuat penyimpanan data dan tampak sederhana untuk diciptakan, begitu entitas-entitas eksternal serta aliran data-aliran daa menuju dan dari sistem diketahui penganalisis dari wawancara dengan *user* dan sebagai hasil analisis dokumen (Pressman, 2015).

Diagram Konteks menggarisbawahi sejumlah karakteristik penting dari suatu sistem (Pressman, 2015):

1. Kelompok pemakai, pihak yang akan memberikan data ke sistem
2. Data, apa saja yang diterima/dihasilkan sistem dari/ke dunia luar
3. Penyimpanan data, tempat sistem harus memberi informasi atau laporan
4. Batasan, yang membedakan antara sistem dan lingkungan

Tabel 2.1. Simbol Diagram Konteks

Simbol Diagram Konteks	Keterangan
 Terminator	Pihak-pihak yang berada di luar sistem, tetapi secara langsung berhubungan dengan sistem dalam hal memberi data atau menerima informasi
 Process	Didalam diagram konteks, berisi mengenai sistem yang akan dibuat
Data Flow 	Berisi data atau informasi yang mengalir dari satu pihak ke sistem dan sebaliknya

2.4 DFD (*Data Flow Diagram*)

Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. DFD ini sering disebut juga dengan nama *Bubble chart*, *Bubble diagram*, model proses, diagram alur kerja, atau model fungsi (Pressman, 2015).

Pada saat informasi mengalir melalui perangkat lunak, dimodifikasi oleh suatu deretan transformasi. Diagram alir data/ DFD (*data flow diagram*) adalah sebuah


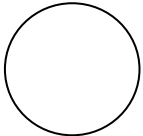
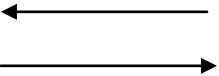

teknis grafis yang menggambarkan aliran informasi dan transformasi yang diaplikasikan pada saat data bergerak dari *input* menjadi *output*.

Berikut ini adalah aturan-aturan pembuatan (DFD) :

1. Didalam *Data Flow Diagram* (DFD) tidak boleh menghubungkan antara *external entity* dengan *entity* lainnya secara langsung.
2. Didalam *Data Flow Diagram* (DFD) tidak boleh menghubungkan *data store* yang satu dengan *data store* yang lainnya secara langsung.
3. Didalam *Data Flow Diagram* (DFD) tidak boleh menghubungkan *data store* dengan *external entity* secara langsung.
4. Setiap proses harus memiliki data *flow* yang masuk dan juga data *flow* yang keluar.

Berikut adalah simbol-simbol DFD :

Tabel 2.2 Simbol-simbol *Data Flow Diagram* (DFD)

Simbol	Keterangan
	<i>External Entity</i> Simbol ini digunakan untuk menggambarkan asal atau tujuan data
	Proses Simbol ini menunjukan proses pengolahan atau tranformasi data
	<i>Data Flow</i> Simbol ini digunakan untuk menggambarkan aliran data yang berjalan
	<i>Data Store</i> Simbol ini menggambarkan <i>data Flow</i> yang sudah disimpan atau diarsipkan

Jenis-jenis DFD (*Data Flow Diagram*) :

1. *Context Diagram* (CD)

Jenis pertama *Context Diagram*, adalah *data flow diagram* tingkat atas (DFD *Top Level*), yaitu diagram yang paling tidak detail, dari sebuah sistem

informasi yang menggambarkan aliran-aliran data ke dalam dan ke luar sistem dan ke dalam dan ke luar entitas-entitas eksternal. (CD menggambarkan sistem dalam satu lingkaran dan hubungan dengan entitas luar. Lingkaran tersebut menggambarkan keseluruhan proses dalam sistem).

Beberapa hal yang harus diperhatikan dalam menggambar CD :

- a. Terminologi sistem
- b. Batas Sistem adalah batas antara “daerah kepentingan sistem”
- c. Lingkungan Sistem adalah segala sesuatu yang berhubungan atau mempengaruhi sistem tersebut
- d. *Interface* adalah aliran yang menghubungkan sebuah sistem dengan lingkungan sistem tersebut

2. Diagram Level n / *Data Flow Diagram Levelled*

Dalam diagram n DFD dapat digunakan untuk menggambarkan diagram fisik maupun diagram diagram logis. Dimana Diagram Level n merupakan hasil pengembangan dari *Context Diagram* ke dalam komponen yang lebih detail tersebut disebut dengan *top-down partitioning*. Jika kita melakukan pengembangan dengan benar, kita akan mendapatkan DFD-DFD yang seimbang.

a. DFD Fisik

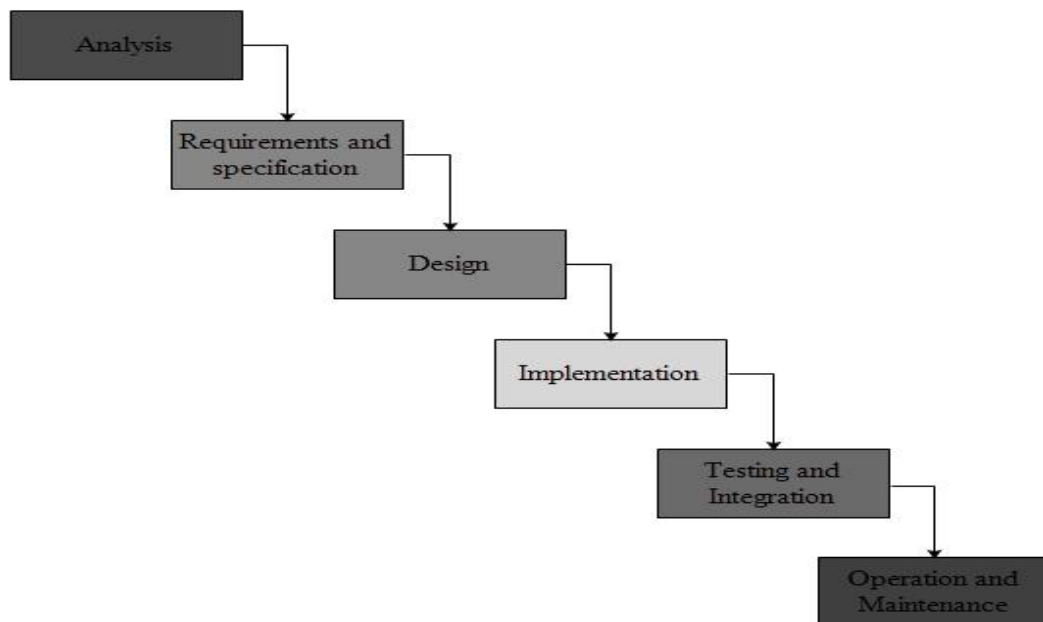
Adalah representasi grafik dari sebuah sistem yang menunjukkan entitas-entitas internal dan eksternal dari sistem tersebut, dan aliran-aliran data ke dalam dan keluar dari entitas-entitas tersebut.

b. DFD Logis

Adalah representasi grafik dari sebuah sistem yang menunjukkan proses-proses dalam sistem tersebut dan aliran-aliran data ke dalam dan ke luar dari proses-proses tersebut. Kita menggunakan DFD logis untuk membuat dokumentasi sebuah sistem informasi karena DFD logis dapat mewakili logika tersebut, yaitu apa yang dilakukan oleh sistem tersebut, tanpa perlu menspesifikasi dimana, bagaimana, dan oleh siapa proses-proses dalam sistem tersebut dilakukan.

2.5 Metode Pengembangan Perangkat Lunak Menggunakan Metode *Waterfall*

Menurut Tata Sutabri (2012) “*Classic Life Cycle*” atau model *Waterfall* merupakan model yang paling banyak dipakai di dalam Software Engineering. Model ini melakukan pendekatan secara sistematis dan urut mulai dari kebutuhan sistem lalu menuju ke tahap analisis, design, coding, testing/*verification*, dan maintenance. Disebut dengan waterfall karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.



Gambar 2.3. Model Air Terjun/*waterfall*

Sumber : Tata Sutabri (2012)

Tahapan Metode Waterfall Tata Sutabri, sebagai berikut.

1. *System/Information Engineering and Modeling*

Permodelan ini diawali dengan mencari kebutuhan dari keseluruhan sistem yang akan di aplikasikan kedalam bentuk *software*. Hal ini sangat penting, mengingat *software* harus dapat berinteraksi dengan elemen-elemen yang lain seperti *hardware*, *database*, dan sebagainya. Tahap ini sering disebut dengan *Project Definition*.

2. *Software Requirements Analysis*

Proses pencarian kebutuhan diintensikan dan difokuskan pada software. Untuk mengetahui sifat dari program yang akan dibuat, maka para software engineer harus mengerti tentang domain informasi dari software, misalnya fungsi yang dibutuhkan, user interface, dengan sebagainya. Dari 2 aktivitas tersebut (pencarian kebutuhan sistem dan software) harus didokumentasikan dan ditunjukkan kepada pelanggan.

3. *Design*

Proses ini digunakan untuk mengubah kebutuhan-kebutuhan di atas menjadi representasi ke dalam bentuk "*blueprint*" software sebelum coding dimulai. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya seperti 2 aktivitas sebelumnya, maka proses ini juga harus didokumentasikan sebagai konfigurasi dari *software*.

4. *Coding*

Desain yang telah dibuat kemudian diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu kedalam bahasa pemrograman melalui proses *coding*. Tahap ini merupakan implementasi dari tahap *design* yang secara teknis nantinya dikerjakan oleh programmer

5. *Testing/Verification*

Sesuatu yang dibuat haruslah diujicobakan. Demikian juga dengan software. Semua fungsi *software* harus diujicobakan, agar *software* bebas dari *error*, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.

6. *Maintenance*

Pemeliharaan suatu *software* diperlukan, termasuk didalamnya adalah pengembangan, karena software yang dibuat tidak selamanya hanya seperti itu. Ketika dijalankan mungkin saja masih ada *error* kecil yang tidak ditemukan sebelumnya, atau ada penambahan fitur-fitur yang belum ada pada *software* tersebut. Pengembangan diperlukan ketika adanya perubahan dari eksternal perusahaan seperti ketika ada pergantian sistem operasi, atau perangkat lainnya.

2.6 Basis Data

Database adalah kumpulan file-file yang mempunyai kaitan antara satu file dengan file yang lain sehingga membentuk satu bangunan data untuk menginformasikan satu perusahaan, instansi dalam batasan tertentu. (Indrajani, 2015)

Istilah-istilah yang digunakan dalam basis data:

- 1) *File* : merupakan kumpulan dari atribut *record-record* sejenis yang mempunyai panjang elemen yang sama, atribut yang sama namun berbeda-beda dalam data *value*-nya.
- 2) *Record* : merupakan kumpulan dari elemen-elemen yang saling berhubungan atau berkaitan menginformasikan tentang *entry* secara lengkap.
- 3) *Field* : merupakan sekumpulan tanda-tanda yang berbentuk kesatuan tersendiri, merupakan bagian terkecil dari *record* dan bentuknya unik dijadikan *field* kunci yang dapat mewakili *record*-nya.
- 4) *Entity* : merupakan tempat kejadian atau konsep yang informasikan direkam.

2.7 MySQL

Menurut Nugroho (2013), “MySQL adalah software atau program Database Server”. Sedangkan SQL adalah bahasa pemrogramannya, bahasa permintaan (query) dalam database server termasuk dalam MySQL itu sendiri. SQL juga dipakai dalam software database server lain, seperti SQL Server, Oracle, PostgreSQL dan lainnya.

Keistimewaan yang dimiliki MySQL adalah sebagai berikut :

1. *Portability*

MySQL dapat berjalan stabil pada berbagai sistem operasi diantaranya seperti Windows, Linux, FreeBSD dan masih banyak lagi.

2. *Open Source*

MySQL didistribusikan secara *open source* (gratis).

3. *Multiuser*

MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. *Performance Tuning*

MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan memproses lebih banyak SQL persatuan waktu.

5. *Column Types*

MySQL memiliki tipe kolom yang sangat kompleks.

6. *Command dan Functions*

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *SELECT* dan *WHERE* dalam *query*.

7. *Security*

MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta password terenkripsi.

8. *Scalability dan Limits*

MySQL mampu menangani *database* dalam skala besar, dengan jumlah record lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu, batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. *Connectivity*

MySQL dapat melakukan koneksi dengan *client* menggunakan protocol TCP/IP, Unix socket (Unix), atau Named Pipes (NT).

10. Localisation

MySQL dapat mendeteksi pesan kesalahan (*error code*) pada *client* dengan menggunakan lebih dari dua puluh bahasa.

11. Interface

MySQL memiliki *Interface* (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

12. Client dan Tools

MySQL dilengkapi dengan berbagai tool yang dapat digunakan untuk administrasi *database*, dan pada setiap tool yang ada pada petunjuk *online*.

13. Struktur Tabel

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan *database* lainnya semacam PostgreSQL ataupun Oracle.

2.8 HTML (Hyper Text Markup Language)

HTML merupakan kepanjangan dari *Hyper Text Markup Language* adalah suatu bahasa yang digunakan untuk membuat halaman-halaman hypertext (hypertext page) pada internet. Dengan konsep hypertext ini, untuk membaca suatu dokumen anda tidak harus melakukannya secara urut, baris demi baris, atau halaman demi halaman. Tetapi anda tidak dapat dengan mudah melompat dari satu topik ke topik lainnya yang anda sukai, seperti halnya jika anda melakukan pada online Help dari suatu aplikasi Windows. HTML dirancang untuk digunakan tanpa tergantung pada suatu platform tertentu (platform independent) (Nugroho, 2013).

Nugroho (2013), menjelaskan bahwa “HTML (Hyper Text Markup Language) untuk desain tampilan, yaitu untuk mengatur teks, tabel dan juga membuat form”.

2.9 PHP (Hypertext Preprocessor)

Nugroho (2013), menjelaskan bahwa “PHP (Hypertext Preprocessor) itu bahasa pemrograman berbasis Web. Jadi, PHP itu adalah bahasa program yang digunakan untuk membuat aplikasi berbasis web (website, blog, atau aplikasi web)”.

Skrip PHP berkedudukan sebagai tag dalam bahasa HTML. Sebagaimana diketahui, HTML adalah bahasa standar untuk membuat halaman-halaman web. Berikut adalah contoh kode HTML (disimpan dengan ekstensi .htm atau .html).