



BAB II

TINJAUAN PUSTAKA

2.1 Teori Yang Berhubungan Dengan Sistem Secara Umum

Teori yang berhubungan dengan sistem meliputi sistem dan data.

2.1.1 Sistem

Menurut Pratama (2014:7), Sistem dapat didefinisikan sebagai sekumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan suatu tugas bersama-sama.

Sedangkan menurut Pratiwi (2016:4), Sistem adalah kumpulan dari obyek-obyek seperti orang, *resources*, konsep dan prosedur yang ditujukan untuk melakukan fungsi tertentu atau memenuhi suatu tujuan. Kemudian sistem juga merupakan kumpulan dari komponen yang berinteraksi bersama-sama secara kolektif untuk melaksanakan tugas.

Berdasarkan uraian tersebut maka dapat disimpulkan bahwa pengertian sistem adalah sekumpulan komponen atau elemen yang dapat diartikan sebagai suatu kumpulan atau obyek-obyek yang terhubung satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan.

2.1.2 Data

Menurut Husda dan Wangdra (2017:13), Data merupakan sesuatu yang belum mempunyai arti bagi penerimanya dan masih memerlukan adanya suatu pengolahan. Data bisa berujud suatu keadaan, gambar, suara, huruf, angka, matematika, bahasa ataupun symbol-simbol lainnya yang bisa kita gunakan sebagai bahan untuk melihat lingkungan, obyek, kejadian ataupun suatu konsep.

Sedangkan menurut Sutabri (2012:1), Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata.

Berdasarkan pengertian di atas dapat disimpulkan bahwa data merupakan catatan atau gambaran fakta berupa gambar, angka, huruf,



simbol-simbol maupun gagasan yang dapat membuktikan suatu kejadian yang nyata.

2.2 Teori yang Berhubungan dengan Penelitian

2.2.1 Peminatan

Menurut Badaruddin (2014:9), Peminatan adalah program kurikuler yang disediakan untuk mengakomodasikan pilihan minat, bakat dan/atau kemampuan peserta didik/konseli dengan orientasi pemusatan, perluasan, dan/atau pendalaman mata pelajaran dan/atau muatan kejuruan.

2.2.2 Keputusan

Menurut Pratiwi (2016:2), Keputusan merupakan hasil pemikiran berupa pemilihan satu diantara beberapa alternatif yang dapat digunakan untuk memecahkan masalah yang dihadapi.

Jenis-jenis keputusan dibedakan menjadi tiga macam yaitu keputusan terstruktur, keputusan tidak terstruktur, dan keputusan semi terstruktur (Pratiwi, 2016:5-6).

1. Keputusan terstruktur

Keputusan-keputusan yang berkaitan dengan persoalan yang telah diketahui sebelumnya. Proses pengambilan keputusan seperti ini biasanya didasarkan atas teknik-teknik tertentu dan sudah dibuat standarnya. Kategori keputusan ini juga dapat dikatakan suatu proses jawaban secara otomatis pada kebijakan yang sudah ditentukan sebelumnya. Secara alamiah hampir semua masalah rutin dan berulang memiliki parameter-parameter persoalan yang telah diketahui dan terdefinisi dengan baik, sehingga jawaban atau proses pengambilan keputusan pun bersifat rutin dan terjadwal.

2. Keputusan tak terstruktur

Keputusan-keputusan yang berkaitan dengan berbagai persoalan baru. Keputusan tidak terstruktur biasanya juga berkaitan dengan persoalan yang cukup pelik, karena banyak parameter yang tidak diketahui atau



belum diketahui. Oleh karena itu, untuk mengambil keputusan ini biasanya intuisi serta pengalaman seorang pelaku organisasi akan sangat membantu.

Keputusan tak terstruktur, adalah “*fuzzy*”, permasalahan kompleks dimana tak ada solusi yang mengikutinya. Masalah yang tak terstruktur adalah tak adanya 3 fase proses yang terstruktur. Keputusan tidak terstruktur (*unstructured decision*) bukan merupakan keputusan yang berulang dan rutin. Contohnya adalah memilih sampul depan sebuah majalah, mengontrak manajemen tingkat senior, dan memilih proyek penelitian awal yang akan dilakukan. Tidak ada kerangka atau model yang dapat memecahkan masalah sejenis ini. Bahkan, dibutuhkan banyak sekali pertimbangan dan intuisi. Walaupun demikian, keputusan tidak terstruktur dapat didukung oleh bantuan dari keputusan yang diambil berdasar hasil komputer, yang berfungsi untuk memfasilitasi pengumpulan informasi dari berbagai sumber. Contohnya adalah keputusan untuk pengembangan teknologi baru, pengembangan jenis usaha baru, keputusan untuk bergabung dengan perusahaan lain, perekrutan eksekutif.

3. Keputusan semi terstruktur

Terdapat beberapa keputusan terstruktur, tetapi tak semua dari fase-fase yang ada. Keputusan semi terstruktur (*semistructured decision*) ditandai dengan dengan peraturan-peraturan yang tidak lengkap untuk mengambil keputusan, dan adanya kebutuhan untuk membuat penilaian serta pertimbangan subjektif sebagai pelengkap analisis data yang formal. Menetapkan anggaran pemasaran untuk suatu produk baru adalah contoh dari keputusan semi terstruktur. Walaupun keputusan seperti ini biasanya tidak dapat secara penuh diotomatisasikan, namun sering didukung dari komputer (*computer-based decision*). Contoh keputusan jenis adalah investasi keuangan, pengevaluasian kredit, penjadwalan produksi, pemberian dana rehabilitasi sekolah, dan pengendalian persediaan.



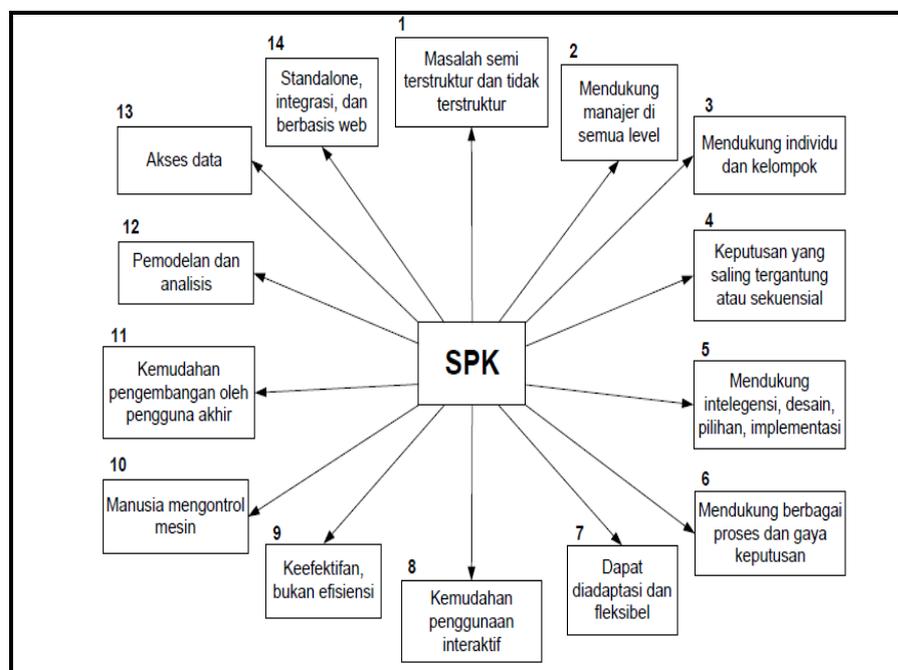
2.2.3 Sistem Pendukung Keputusan

Menurut Pratiwi dalam McLeod (Pratiwi, 2016:4), Sistem pendukung keputusan merupakan sistem penghasil informasi yang ditujukan pada suatu masalah yang harus dibuat oleh manajer, sistem pendukung keputusan merupakan suatu sistem informasi yang ditujukan untuk membantu manajemen dalam memecahkan masalah yang dihadapinya.

Definisi selengkapnya adalah sistem penghasil informasi spesifik yang ditujukan untuk memecahkan suatu masalah tertentu yang harus dipecahkan oleh manajer pada berbagai tingkatan. Sistem pendukung keputusan juga merupakan suatu sistem informasi berbasis komputer yang menghasilkan berbagai alternatif keputusan untuk membantu manajemen dalam menangani berbagai permasalahan yang terstruktur ataupun tidak terstruktur dengan menggunakan data atau model.

A. Karakteristik dan kapabilitas DSS

Karakteristik dan kapabilitas kunci dari DSS adalah (ditunjukkan pada gambar 2.1) (Turban, E., dkk, 2005:140-143):



Gambar 2.1 Karakteristik dan kapabilitas kunci dari DSS

Sumber: Turban, E., dkk (2005:142)



1. Dukungan untuk pengambil keputusan, terutama pada situasi semistruktur dan tak terstruktur, dengan menyertakan penilaian manusia dan informasi terkomputerisasi. Masalah-masalah tersebut tidak dapat dipecahkan (atau tidak dapat dipecahkan dengan konvenien) oleh sistem komputer lain atay oleh metode atau alat kuantitatif standar.
2. Dukungan untuk semua level manajerial, dari eksekutif puncak sampai manajer lini.
3. Dukungan untuk individu dan kelompok. Misalnya yang kurang terstruktur sering memerlukan keterlibatan individu dari departemen dan tingkat organisasional yang berbeda atau bahkan dari organisasi lain. DSS mendukung tim virtual melalui alat-alat Web kolaboratif.
4. Dukungan untuk keputusan independen dan atau sekuensial. Keputusan dapat dibuat satu kali, beberapa kali, atau berulang (dalam interval yang sama).
5. Dukungan di semua fase proses proses pengambilan keputusan: inteligensi, desain, pilihan dan implementasi.
6. Dukungan di berbagai proses dan gaya pengambilan keputusan.
7. Adaptivitas sepanjang waktu. Pengambil keputusan seharusnya reaktif, dapat menghadapi perubahan kondisi secara cepat, dan dapat mengadaptasikan DSS untuk memenuhi perubahan tersebut. DSS bersifat fleksibel dan karena itu pengguna dapat menambahkan, menghapus, menggabungkan, mengubah, atau menyusun kembali elemen-elemen dasr. DSS juga fleksibel dalam hal dapat dimodifikasi untuk memecahkan masalah lain yang sejenis.
8. Pengguna merasa seperti rumah. Ramah-pengguna, kapabilitas grafis yang sangat kuat, dan antarmuka manusia mesin interaktif dengan satu bahasa alami dapat sangat meningkatkan keefektifan DSS. Kebanyakan aplikasi DSS yang baru menggunakan antarmuka berbasis-Web.



9. Peningkatan terhadap keefektifan pengambilan keputusan (akurasi, timeliness, kualitas) ketimbang pada efisiensinya (biaya pengambilan keputusan). Ketika DSS disebar, pengambilan keputusan sering membutuhkan waktu lebih lama, namun keputusan lebih baik.
10. Kontrol penuh oleh pengambil keputusan terhadap semua langkah proses pengambilan keputusan dalam langkah memecahkan suatu masalah. DSS secara khusus menekankan untuk mendukung pengambilan keputusan, bukannya menggantikan.
11. Pengguna akhir dapat mengembangkan dan memodifikasi sendiri sistem sederhana. Sistem yang lebih besar dapat dibangun dengan bantuan ahli sistem informasi. Perangkat lunak OLAP dalam kaitannya dengan data warehouse membolehkan pengguna untuk membangun DSS yang cukup besar dan kompleks.
12. Biasanya model-model digunakan untuk menganalisis situasi pengambilan keputusan. Kapabilitas pemodelan memungkinkan eksperimen dengan berbagai strategi yang berbeda dibawah konfigurasi yang berbeda.
13. Akses disediakan untuk berbagai sumber data, format, dan tipe, mulai dari sistem informasi geografis (GIS) sampai sistem berorientasi-objek.
14. Dapat dilakukan sebagai alat standalone yang digunakan oleh seorang pengambil keputusan pada satu lokasi dan di beberapa organisasi sepanjang rantai persediaan. Dapat diintegrasikan dengan DSS lain dan atau aplikasi lain, dan dapat didistribusikan secara internal dan eksternal dengan menggunakan networking dan teknologi Web.



B. Komponen-komponen Sistem Pendukung Keputusan

Menurut Latif (2018:4), Komponen-komponen Sistem Pendukung Keputusan terdiri dari:

1. *Data Management*

Data Management. Termasuk database, yang mengandung data yang relevan untuk berbagai situasi dan diatur oleh software yang disebut *Database Management Systems (DBMS)*.

2. *Model Management*

Model Management. Melibatkan model finansial, statistikal, *management science*, atau berbagai model kuantitatif lainnya, sehingga dapat memberikan ke sistem suatu kemampuan analitis, dan manajemen *software* yang diperlukan.

3. *Communication (dialog subsystem)*

Communication (dialog subsystem). User dapat berkomunikasi dan memberikan perintah pada DSS melalui sub sistem ini. Ini berarti menyediakan antarmuka.

4. *Knowledge Management*

Knowledge Management. Subsistem optional ini dapat mendukung sub sistem lain atau bertindak sebagai komponen yang berdiri sendiri.

C. Fase Pengambilan Keputusan

Menurut Simon dalam Latif (2018:5-6), Ada tiga fase dalam proses Pengambilan Keputusan diantaranya sebagai berikut

1. *Intellegence*

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



2. Design

Tahap ini merupakan proses menemukan mengembangkan dan menganalisis alternatif tindakan yang bisa dilakukan. Tahap ini meliputi menguji kelayakan solusi.

3. Choice

Pada tahap ini dilakukan proses pemilihan diantara berbagai alternatif tindakan yang mungkin dijalankan. Hasil pemilihan tersebut kemudian diimplementasikan dalam proses pengambilan keputusan.

2.3 Metode *Simple Additive Weighting*

Menurut Pratiwi (2016:136-137), Metode *Simple Additive Weighting* (SAW) adalah metode penjumlahan bobot dari kinerja setiap objek-objek yang berbeda dan memiliki kesempatan yang sama pada semua kriteria yang dimiliki. Metode *Simple Additive Weighting* (SAW) memerlukan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat dibandingkan dengan semua rating alternatif yang ada.

Metode SAW dapat membantu dalam pengambilan keputusan suatu kasus, akan tetapi perhitungan dengan menggunakan metode SAW ini hanya yang menghasilkan nilai terbesar yang akan terpilih sebagai alternatif yang terbaik. Perhitungan akan sesuai dengan metode ini apabila alternatif yang terpilih memenuhi kriteria yang telah dibutuhkan dalam perhitungan lebih singkat.

$$r_{ij} = \begin{cases} \frac{X_{ij}}{\text{Max}_i X_{ij}} & \rightarrow \text{Jika } j \text{ adalah attribute keuntungan (benefit)} \\ \frac{\text{Min}_i X_{ij}}{X_{ij}} & \rightarrow \text{Jika } j \text{ adalah atribut biaya (cost)} \end{cases}$$

Keterangan :

r_{ij} = nilai *rating* kinerja ternormalisasi

x_{ij} = nilai atribut yang dimiliki dari setiap kriteria

$\text{Max}_i X_{ij}$ = nilai terbesar dari setiap kriteria i



Mini X_{ij} = nilai terkecil dari setiap kriteria i

benefit = jika nilai terbesar adalah terbaik

cost = jika nilai terkecil adalah terbaik

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

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

Keterangan :

V_i = ranking untuk setiap alternatif

w_j = nilai bobot dari setiap kinerja

r_{ij} = nilai rating kinerja ternormalisasi

Nilai V_i yang lebih besar mengindikasikan bahwa alternatif A_i yang terpilih.

Berikut adalah langkah-langkah penyelesaian *Simple Additive Weighting* (SAW):

1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu C_i .
2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
3. Membuat matriks keputusan berdasarkan kriteria (C_i), kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks ternormalisasi R .
4. Hasil akhir diperoleh dari proses perankingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan vektor bobot sehingga diperoleh nilai terbesar yang diperoleh sebagai alternatif terbaik (A_i) sebagai solusi.



2.4 Teori yang Berhubungan Teknik Analisa yang Digunakan

2.4.1 Use Case

Use Case atau Diagram *Use Case* merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat dan digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi serta siapa saja yang berhak menggunakan fungsi-fungsi itu.

Berikut adalah simbol-simbol *use case* menurut Rosa dan Shalahuddin (2015:156):

Tabel 2.1 Simbol *Use Case* Diagram

No	Simbol	Nama	Keterangan
1.		Aktor/ <i>Actor</i>	Aktor adalah pengguna sistem. aktor tidak terbatas hanya manusia saja, jika sebuah sistem berkomunikasi dengan aplikasi lain dan membutuhkan input atau memberikan output, maka aplikasi tersebut juga bisa dianggap sebagai aktor.
2.		<i>Use case</i>	<i>Use case</i> digambarkan sebagai lingkaran elips dengan nama <i>use case</i> dituliskan didalam elips tersebut, digunakan sebagai unit-unit yang saling bertukar pesan antar unit / aktor.
3.		<i>Association</i> / Asosiasi	Asosiasi digunakan untuk menghubungkan <i>actor</i> dengan <i>use case</i> . Asosiasi digambarkan dengan sebuah garis yang menghubungkan antara <i>Actor</i> dengan <i>Use Case</i> .
4.		<i>Generalizatio</i> <i>n</i> /	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan



		Generalisasi	struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
5.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
6.		<i>Extend /</i> Ekstensi	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan

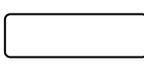
(Sumber: Rosa dan Shalahuddin, 2015:156)

2.4.2 Activity Diagram

Menurut Rosa dan Shalahuddin (2015:161), “Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”.

Berikut adalah simbol-simbol yang ada pada diagram activity :

Tabel 2.2 Simbol *Activity Diagram*

No	Simbol	Nama	Keterangan
1.		<i>Start state</i>	Titik awal atau permulaan
2.		<i>End state</i>	Titik akhir atau akhir dari aktivitas
3.		<i>Activity</i>	<i>Activity</i> atau aktivitas yang dilakukan oleh <i>actor</i>
4.		<i>Decision</i>	Pilihan untuk mengambil keputusan
5.		<i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

(Sumber: Rosa dan Shalahuddin, 2015:162)

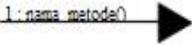


2.4.3 Sequence Diagram

Menurut Rossa dan Salahuddin (2015:165), “*Sequence diagram* atau diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirim dan diterima antar objek”.

Adapun simbol-simbol yang digunakan dalam *sequence diagram* adalah sebagai berikut:

Tabel 2.4 Simbol *Sequence Diagram*

No.	Gambar	Nama	Keterangan
1.		Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
2.		Lifeline	Menyatakan kehidupan suatu objek.
3.		Waktu Aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
4.		Pesan tipe call	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri.

(Sumber: Rosa dan Shalahuddin, 2015:165)

2.4.4 Class Diagram

Menurut Rosa dan Shalahuddin (2015:141), diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Berikut simbol-simbol yang ada pada *class diagram* :

Tabel 2.3 Simbol *Class diagram*

No.	Gambar	Nama	Keterangan
1.		<i>Class/Kelas</i>	Kelas pada struktur system
2.		<i>Interface/Antarmuka</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.		<i>Association/Asosiasi</i>	Relasi antarkelas dengan makna umum, asosiasi biasanya disertai dengan <i>multiplicity</i> .
4.		<i>Directed association/Asosiasi</i> berarah	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.		Generalisasi	Relasi antar kelas dengan makna generalisasi- spesialisasi (umum khusus).
6.		<i>Dependency/Kebergantungan</i>	Relasi antar kelas dengan makna kebergantungan antar kelas
7.		<i>Aggregation/ Agregasi</i>	Relasi antar kelas dengan makna semua-bagian.

(Sumber: Rosa dan Shalahuddin, 2015:146)

2.5 Teori Pendukung Lainnya

2.5.1 Database

Menurut Puspitosari (2013:3), Database adalah kumpulan data yang terintegrasi dan diatur sedemikian rupa sehingga data tersebut dapat dimanipulasi, diambil, dan dicari secara cepat.

Sedangkan menurut Hidayatullah dan Kawistara (2014:147), Basis data (*database*) dapat didefinisikan sebagai himpunan kelompok data saling



berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.

Berdasarkan pengertian di atas dapat disimpulkan bahwa database adalah kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi.

2.5.2 HTML (*Hypertext Markup Language*)

Menurut Raharjo (2016:449), HTML adalah singkatan dari *Hyper Text Markup Language* (HTML) merupakan file teks yang ditulis menggunakan aturan-aturan kode tertentu kemudian disajikan ke user melalui suatu aplikasi *web browser*. Informasi yang tampil di *web* selalu dibuat menggunakan kode HTML oleh karena itu dokumen HTML sering disebut juga sebagai *website* (halaman *web*). Untuk membuat dokumen HTML kita tidak bergantung pada aplikasi tertentu, karena dokumen HTML dapat dibuat menggunakan aplikasi Text Editor apapun bisa notepad (untuk lingkungan *MS Windows*) Emacs atau Vi Editor (untuk lingkungan *Linux*).

Sedangkan menurut Hidayatullah dan Kawistara (2014:180), *Hypertext Markup Language* (HTML) adalah bahasa standard yang digunakan untuk menampilkan halaman *web*. Yang bisa dilakukan HTML yaitu :

1. Mengatur tampilan dari halaman *web* dan isinya.
2. Membuat tabel dalam halamn *web*.
3. Mempublikasikan halaman *web* secara *online*.
4. Membuat form yang bisa digunakan untuk menangani registrasi dan transaksi via *web*.
5. Menambahkan objek-objek seperti citra, audio, video, animasi, *java applet* dalam halaman *web*.
6. Menampilkan area gambar (*canvas*) di *browser*.

Berdasarkan uraian di atas dapat disimpulkan bahwa HTML adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah penjelajah



web Internet dan pemformatan hiperteks sederhana agar dapat menghasilkan tampilan wujud yang terintegrasi yang dapat diakses secara online.

2.5.3 MySQL

Menurut Enterprise (2017:217), MySQL (My Structure Query Language) atau yang biasa dibaca “mai-se-kuel” adalah sebuah program pembuat database yang bersifat open source (terbuka).

Sedangkan menurut Hidayatullah dan Kawistara (2015:180), *MySQL* adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan oleh para pemrogram aplikasi web. Kelebihan dari *MySQL* adalah gratis, handal, selalu di-*update* dan banyak form yang memfasilitasi para pengguna jika memiliki kendala. *MySQL* juga menjadi DBMS yang sering dibundling dengan web *server* sehingga proses instalasinya jadi lebih mudah.

Berdasarkan pengertian di atas dapat disimpulkan bahwa *MySql* merupakan database engine atau server database yang bersifat open source yang digunakan oleh para pemrogram aplikasi web.

2.5.4 PHP (*Hypertext Preprocessor*)

Menurut Raharjo (2016:38), PHP adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi web. Ketika dipanggil dari *web browser*, program yang ditulis dengan PHP akan di-*parsing* di dalam web server oleh *interpreter* PHP dan diterjemah kedalam dokumen HTML, yang selanjutnya akan ditampilkan kembali ke *web browser*. Karena pemrosesan program PHP dilakukan di lingkungan *web server*, PHP dikatakan sebagai bahasa sisi *server* (*server-side*).

Sedangkan menurut Hidayatullah dan Kawistara (2014:231), PHP *Hypertext Preprocessor* atau disingkat dengan PHP ini adalah suatu bahasa *scripting* khususnya digunakan untuk *web development*. Karena sifatnya yang *server side scripting*, maka untuk menjalankan PHP harus menggunakan *web server*.

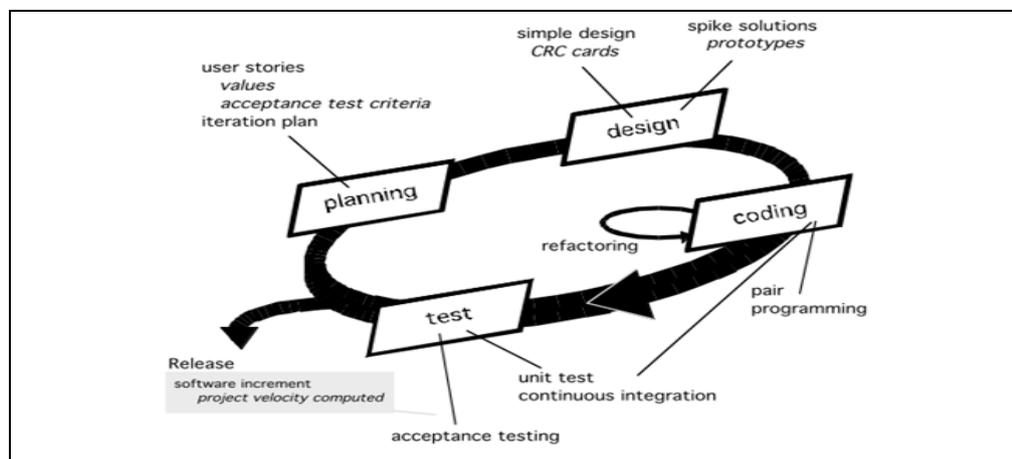


Berdasarkan pengertian di atas dapat disimpulkan bahwa PHP adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML yang biasanya dipakai untuk memprogram situs web dinamis.

2.6 Metode Pengembangan *Extreme Programming (XP)*

Menurut Suryantara (2017), Metode pengembangan yang penulis pakai adalah metodologi *Extreme Programming (XP)*. *Extreme Programming (XP)* merupakan salah satu metodologi rekayasa perangkat lunak yang banyak digunakan untuk mengembangkan aplikasi oleh para developer. XP sangat cocok untuk pengembangan proyek yang memerlukan adaptasi cepat dalam perubahan-perubahan yang terjadi selama pengembangan aplikasi. Kent Back mengakui dan menegaskan bahwa XP tidak selalu cocok untuk setiap proyek pengembangan perangkat lunak. Namun, XP memiliki kelebihan yaitu sesuai untuk proyek dynamic requirements atau proyek yang memiliki berbagai requirements yang tidak jelas dari klien.

Tahapan pengembangan perangkat lunak dengan XP meliputi :



Gambar 2.2 : Proses *Extreme Programming (XP)*

Sumber : Suryantara (2017)

1. Planning / Perencanaan. Tahap ini dimulai dengan pemahaman konteks bisnis dari aplikasi, mendefinisikan keluaran (output), fitur yang ada pada aplikasi, fungsi dari aplikasi yang dibuat, penentuan waktu dan biaya pengembangan aplikasi.



2. Design/Perancangan. Tahap ini menekankan pada desain aplikasi secara sederhana. Alat untuk mendesain pada tahap ini dapat menggunakan kartu CRC (*Class Responsibility Collaborator*) . CRC digunakan untuk pemetaan (membangun) kelas-kelas yang akan digunakan pada diagram use case, diagram kelas, dan diagram objek.
3. Coding/Pengkodean. Hal Utama dalam pengembangan aplikasi dengan XP adalah pair programming (dalam membuat program melibatkan 2 atau lebih programmer).
4. Testing/Pengujian. Tahap ini memfokuskan pada pengujian fitur-fitur yang ada pada aplikasi sehingga tidak ada kesalahan (error) dan aplikasi yang dibuat sesuai dengan proses bisnis pada pelanggan.

2.7 Metode Pengujian *Black Box*

Menurut Pressman (2012:597), Metode pengujian digunakan untuk mengetahui fungsi yang telah ditentukan bahwa suatu sistem telah dirancang dapat menunjukkan bahwa masing-masing fungsi sepenuhnya beroperasi. Pengujian kotak hitam (*black box*), juga disebut pengujian perilaku, berfokus pada persyaratan fungsional perangkat lunak.

Artinya, teknik pengujian kotak hitam memungkinkan untuk membuat beberapa kumpulan kondisi masukan yang sepenuhnya akan melakukan semua kebutuhan fungsional untuk program. Pengujian kotak hitam (*black box*) bukan teknik alternatif untuk kotak putih (*white box*).

Pengujian kotak hitam (*black box*) berupaya untuk menemukan kesalahan dalam kategori berikut: (1) fungsi yang salah atau hilang, (2) kesalahan dalam struktur data atau akses basis eksternal, (4) kesalahan perilaku atau kinerja, dan (5) kesalahan inisialisasi dan penghentian.

2.8 Referensi Jurnal

Menurut penelitian yang dilakukan oleh Melisa Elistri, Jusuf Wahyudi dan Reno Supradi pada tahun 2014 yang berjudul “Penerapan Metode SAW dalam Sistem Pendukung Keputusan Pemilihan Jurusan pada Sekolah



Menengah Atas Negeri 8 Seluma”, penelitian ini bertujuan untuk membantu dalam pembuatan keputusan pemilihan jurusan pada siswa SMA Negeri 8 Seluma. Sistem ini dirancang dengan menggunakan bahasa pemrograman *Visual Basic 6.0* dan dapat dijalankan pada komputer tunggal dibawah sistem operasi *windows*.

Berdasarkan jurnal penelitian oleh Gigih Adi Prabowo dan Beta Noranita pada tahun 2014 dalam jurnal yang berjudul “Sistem Pendukung Keputusan Penentuan Peminatan Peserta Didik Menggunakan Metode *Weighted Product* Berbasis Web (Studi Kasus : SMA Negeri 1 Purwodadi Grobogan”, Sistem Pendukung Keputusan Penentuan Peminatan Peserta Didik Menggunakan Metode *Weighted Product* dapat dijadikan metode penentuan peminatan sekolah karena dari hasil pengujian dengan membandingkan peminatan secara manual menghasilkan tingkat akurasi sebesar 85%, sehingga dapat membantu pihak sekolah dalam menentukan peminatan dengan lebih efektif dan efisien. Untuk mendukung penelitian ini, peneliti menggunakan bahasa pemrograman *PHP*.

Pada penelitian yang dilakukan oleh Rohmat Taufiq dan Ikhsan Shahlin Mustofa pada tahun 2017 dalam jurnal yang berjudul “Perancangan Sistem Pendukung Keputusan Kejurusan Menggunakan Metode *Simple Addictive Weighting* (SAW) di SMA Negeri 15 Tangerang”, tujuan dari penelitian ini adalah untuk membantu sekolah dalam pengambilan keputusan untuk menentukan siswa di tepatkan di kelas yang sesuai dengan kemampuan akademik siswa atau nilai siswa. Pada aplikasi ini, peneliti menggunakan metode *Waterfall*.

Lain halnya penelitian yang dilakukan oleh Adi Supriyatna pada tahun 2018 dalam jurnal yang berjudul “Metode *Extreme Programming* pada Pembangunan Web Aplikasi Seleksi Peserta Pelatihan Kerja”, penelitian ini bertujuan untuk menciptakan sebuah aplikasi berbasis web dengan metode *Extreme Programming* (XP) yang bermanfaat bagi masyarakat untuk memudahkan dalam proses pendaftaran dan pelaksanaan ujian seleksi menjadi peserta pelatihan kerja. *Extreme Programming* (XP) merupakan sebuah



proses rekayasa perangkat lunak yang cenderung menggunakan pendekatan berorientasi objek dan sasaran dari metode ini adalah tim yang dibentuk dalam skala kecil sampai medium serta metode ini juga sesuai jika tim yang dihadapkan dengan *requirement* yang tidak jelas maupun terjadi perubahan-perubahan *requirement* yang sangat cepat. Sistem Informasi ini dirancang dengan menggunakan bahasa pemrograman *PHP*, *CSS* dan *Jquery* dengan menggunakan *database MySQL*.

Adapun penelitian yang dilakukan oleh Agus Subhan Akbar pada tahun 2017 dalam jurnal yang berjudul “Rancang Bangun Sistem Informasi Administrasi Hotel dengan Metode *Extreme Programming*”, tujuan dari penelitian yaitu untuk mengembangkan sistem informasi administrasi hotel untuk memudahkan pencatatan operasional hotel, laporan keuangan dan laporan lainnya untuk mendukung pengambil keputusan dalam membuat keputusan terkait peningkatan pelayanan dan pendapatan hotel. Metode yang digunakan yaitu *Extreme Programming (XP)* yang merupakan salah satu metode yang termasuk dalam *Agile Methodology*. Pada metode *Extreme Programming (XP)* ada iterasi yang bisa dilakukan berulang kali sesuai dengan kebutuhan yang ada. Pada aplikasi ini, peneliti menggunakan bahasa pemrograman *NetBeans 7.4* dengan *database MySQL*.