**BAB II**

**LANDASAN TEORI**

* 1. **Penerapan**

Menurut Kamus Besar Bahasa Indonesia (KBBI), yang saya dapatkan dari <http://kbbi.web.id/terap-2>, pengertian penerapan adalah proses, cara, perbuatan menerapkan, sedangkan menurut beberapa ahli, penerapan adalah suatu perbuatan mempraktekkan suatu teori, metode, dan hal lain untuk mencapai tujuan tertentu dan untuk suatu kepentingan yang diinginkan oleh suatu kelompok atau golongan yang telah terencana dan tersusun sebelumnya. Menurut (Idris, 2019) Penerapan merupakan pelaksanaan sebuah hasil kerja yang dilakukan baik oleh individu maupun kelompok untuk tercapainya sebuah tujuan.

* 1. ***Decision Support System* (DSS)**
     1. **Pengertian *System***

*System* atau sistem merupakan suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau melakukan sasaran yang tertentu (Hutahaean, 2015).

Sedangkan menurut (Prof. Dr. Sri Mulyani & Sistematika, 2017), menyatakan bahwa sistem bisa diartikan sebagai sekumpulan subsistem, komponen ataupun elemen yang saling bekerja sama dengan tujuan yang sama untuk menghasilkan *output* yang sudah ditentukan sebelumnya.

Berdasarkan uraian di atas, dapat disimpulkan bahwa sistem adalah satu kesatuan yang saling berhubungan, dimana elemen-elemen di dalamnya memiliki peranan masing-masing untuk melakukan proses pengolahan suatu *input* hingga menghasilkan *output* yang diinginkan.

Sistem memiliki unsur yang saling melengkapi untuk mencapai tujuan tertentu. Unsur yang terdapat pada sistem itu yang disebut dengan subsistem. Subsistem ini saling berinteraksi melalui komunikasi agar dapat bekerja dalam mencapai tujuan. Adapun Sistem merupakan seperangkat dari sejumlah elemen yang saling berintegrasi untuk mencapai tujuan tertentu. Elemen yang mewakili pada sistem adalah *input, processing,* dan *Output.* Komponen pada sistem dapat digambarkan pada Gambar 2.1:

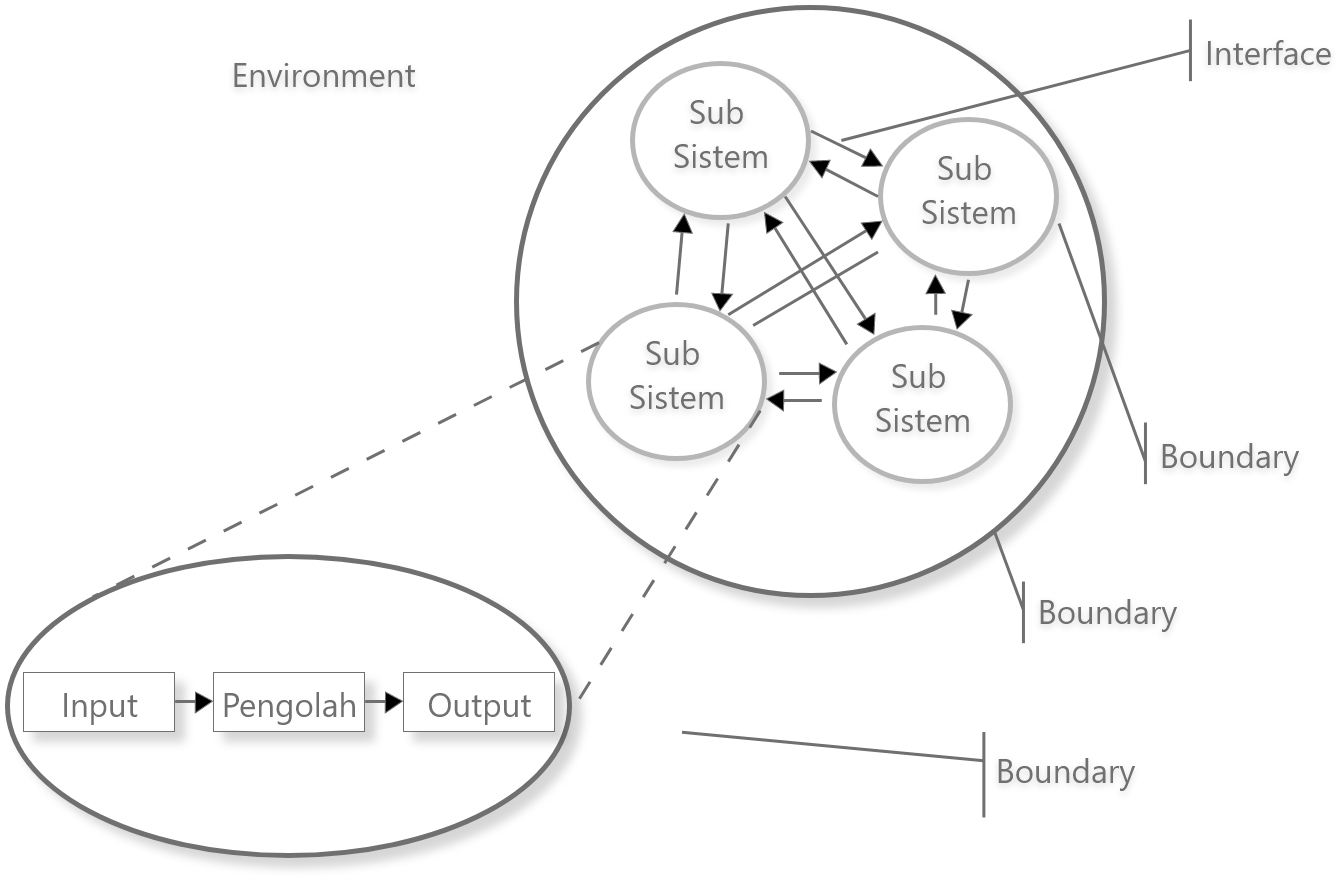
**Gambar 2.1.** Komponen Sistem

(Sumber: Hutahaean, 2015)

Dapat dijelaskan dari komponen sistem di atas adalah sebagai berikut:

1. *Input* yaitu bagian yang akan dimasukkan kedalam sistem.
2. *Processing* yaitu suatu pengolahan karena mengubah sebuah masukan menjadi *output*.
3. *Output* yaitu hasil dari proses pengolahan dari *input* atau hasil akhir dari *input*.
   * 1. **Karakteristik Sistem**

Suatu sistem mempunyai ciri-ciri karakteristik yang terdapat pada sekumpulan elemen yang harus dipahami dalam megidentifikasi pembuatan sistem (Hutahaean, 2015).



**Gambar 2.2.** Karakteristik Sistem

(Sumber: Hutahaean, 2015)

Adapun karakteristik sistem yang dimaksud adalah sebagai berikut:

1. Komponen

Sistem terdiri dari sejumlah komponen yang saling berinteraksi dan bekerja sama untuk membentuk satu kesatuan. Komponen sistem dapat berupa sub sistem atau bagian-bagian dari sistem.

1. Batasan sistem (*boundary*)

Daerah yang membatasi antara suatu sistem dengan sistem lainnya atau dengan lingkungan luar dinamakan dengan batasan sistem. Batasan sistem ini memungkinkan sistem dipandang sebagai satu kesatuan dan juga menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

1. Lingkungan luar sistem (*environment*)

Apapun yang berada di luar batas dari sistem dan mempengaruhi sistem tersebut dinamakan dengan lingkungan luar sistem. Lingkungan luar yang bersifat menguntungkan wajib dipelihara dan yang merugikan harus dikendalikan agar tidak mengganggu kelangsungan sistem.

1. Penghubung sistem (*interface*)

Media penghubung diperlukan untuk mengalirkan sumber-sumber daya dari sub sistem ke sub sistem lainnya dinamakan dengan penghubung sistem.

1. Masukkan sistem (*input*)

Energi yang dimasukkan ke dalam sistem dinamakan dengan masukan sistem (*input*) dapat berupa perawatan dan masukan sinyal. Perawatan ini berfungsi agar sistem dapat beroperasi dan masukan sinyal adalah energi yang diproses untuk menghasilkan keluaran (*output*).

1. Keluaran sistem (*output*)

Hasil dari energi yang telah diolah dan diklasifikasikan menjadi keluaran yang berguna dinamakan dengan keluaran sistem (*output*). Informasi merupakan contoh keluaran sistem.

1. Pengolah sistem

Untuk mengolah masukan menjadi keluaran diperlukan suatu pengolah yang dinamakan dengan pengolah sistem.

1. Sasaran sistem

Sistem pasti memiliki tujuan atau sasaran yang sangat menentukan input yang dibutuhkan oleh sistem dan keluaran yang dihasilkan.

* + 1. **Konsep *Decision Support System* (DSS)**

*Decision Support System* (DSS) atau Sistem Penunjang Keputusan (SPK) pada awal tahun dikenalkan oleh Michael S. Scott Morton 1970-an, DSS juga dikenal dengan istilah *Management Decision System*. Proses pengambilan keputusan tersebut diolah dengan berupa data dan informasi yang memiliki tujuan untuk menghasilkan berbagai alternatif keputusan yang dapat diambil. DSS dirancang untuk mendukung seluruh tahap pengambilan keputusan dari identifikasi masalah, memilih data yang relevan, menentukan pendekatan yang digunakan dalam proses pengambilan keputusan, evaluasi pemilihan alternatif (Limbong et al., 2020).

* + 1. **Pengertian DSS**

DSS dapat dikatakan sebagai sistem pendukung keputusan yang merupakan sistem berbasis perangkat lunak interaktif yang dimaksudkan untuk membantu pembuat keputusan mengumpulkan informasi yang berguna dari kombinasi data mentah, dokumen, dan pengetahuan pribadi, atau model bisnis untuk mengidentifikasi dan menyelesaikan masalah dan membuat keputusan. Sistem ini digunakan untuk membantu membuat keputusan dalam situasi semi-terstruktur dan tidak terstruktur, dimana tidak ada yang tahu secara umum bagaimana keputusan dibuat (Zhang et al., 2014; Simarmata et al., 2020) dalam (Limbong et al., 2020).

* + 1. **Tahapan DSS**

Menurut (simangnsong dan sinaga 2019) diacu dalam (Limbong et al., 2020), Ada empat tahap yang saling berhubungan dan saling berurutan dalam mengabil keputusan. Empat proses tersebut adalah:

1. *Intelligence*

Kecerdasan dapat didefinisikan dalam banyak pemahaman: pemahaman logika, kecerdasan diri, pembelajaran, pengetahuan empsional, penalaran, perencanaan, kreatifitas, pemikiran kritis, dan pemecah masalah. Secara umum ini dapat digambarkan sebagai kemampuan untuk mempersipsikan sebuah informasi, dan mempertahankannya sebagai pengetahuan yang diterapkan.

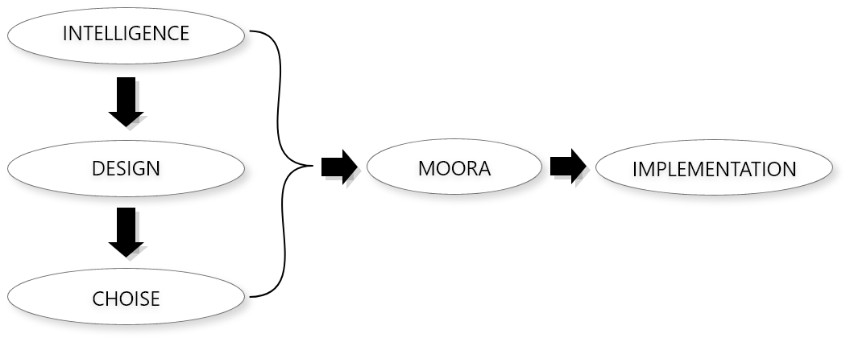
1. *Design*

Desain adalah rencana untuk spesifikasi untuk kontruksi objek atau sistem untuk implementasi suatu kegiatan atau proses, atau hasil dari rencana atau speifikasi itu dalam bentuk *prototype*, produk dan proses, kata kerja mendesain mengekspresikan proses pengembangan suatu desain.

1. *Choice*

Tahap ini dilakukan untuk menentukan sebuah pilihan dari berbagai aspek pencarian, evaluasi dan penyelesaian yang dibuat sesuai model yang telah dirancang. Penyelesaian dengan menerapkan sebuh model adalah nilai spesifik dari alternatif yang terpilih.

1. *Implementation*

Implementasi diterapkan pada teknologi untuk menggambarkan interaksi unsur-unsur dalam bahasa pemrograman. Penerapan digunakan untuk mengenali dan menggunakan elemen kode atau sumber daya pemrograman yang ditulis kedalam program. Model simon menjelaskan alur dari sebuah sistem dengan memanfaatkan adanya informasi yang ada, adapun penerapan model DSS adalah:

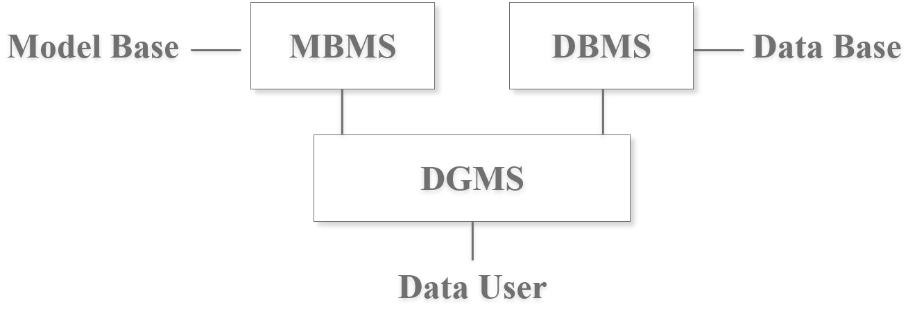
**Gambar 2.3.** *Fase* Pengambilan Keputusan

(Sumber: Limbong et al., 2020)

Dalam ke empat tahapan di atas menjelaskan teknik dalam penerapan sistem pendukung keputusan, berdasarkan alur di atas mempunyai peranan untuk menghasilkan sebuah keputusan yang tepat. *Intelligence* merupakan hasil dari *fase* kontribus dari sistem. *Choice* merupakan *fase* yang digunakan untuk membuat sebuah pemilihan sebuah kriteria yang tepat sebagai pendukung dalam mengambil sebuah keputusan. *Design* merupakan *fase* hasil dari kontribusi dari *intelegence* dan *design*. Pengambilan yang dibuat dalam penentuan keputusan yang terdiri dari beberapa tindakan yang dijadikan sebagai alternatif dalam mencapai beberapa tujuan sesuai dengan yang sudah diterapkan.

* + 1. **Komponen DSS**

Menurut (Limbong et al., 2020), komponen-komponen dari *Decision Support System* sebagai berikut:

* + - * 1. Sistem manajemen basis data/ *Database management system* (DBMS). DBMS berfungsi sebagai bank data untuk DSS. Ini menyimpan sejumlah data yang relevan dengan kelas masalah yang telah diarancang DSS dan menyediakan struktur data logis (sebagai lawan struktur data fisik) yang digunakan pengguna untuk berinteraksi. DBMS memisahkan pengguna dari aspek fisik struktur dan pemrosesan basis data. Ini juga harus mampu memberi informasi kepada pengguna tentang jenis data yang tersedia dan cara mendapatkan akses ke data tersebut.
        2. *Model Base Management System* (MBMS). Peran MBMS *analog* dengan peran DBMS. Fungsi utamanya adalah memberikan independensi antara model tertentu yang digunakan dalam DSS dari aplikasi yang menggunakannya. Tujuan MBMS adalah mengubah data dari DBMS menjadi informasi yang berguna dalam pengambilan keputusan. MBMS juga harus mampu membantu pengguna dalam membangun model.
        3. *Dialog Generation and Management Sysytem* (DGMS). Produk utama dan interaksi dengan DSS adalah wawasan. Karena penggunanya seringkali orang yang tidak terlatih dengan komputer, DSS perlu dilengkapi dengan antarmuka yang intuitif dan mudah digunakan.

**Gambar 2.4.** Arsitektur Decision Support System

(Sumber: Limbong et al., 2020)

* + 1. **Karakteristik DSS**

Karakteristik dan kemampuan dari DSS menurut (Limbong et al., 2020), yaitu sebagai berikut:

1. DSS mengabungkan data dan model menjadi satu bagian. DSS dirancang untuk membantu manajer dalam proses pengambilan keputusan masalah semi-struktural.
2. DSS memberi dukungan untuk pertimbangan manajer daripada dimaksudkan untuk menggantikan fungsi manajer.
3. DSS memiliki fasilitas interaktif yang dapat memfasilitasi hubungan antara manusia dan komputer.
4. Bekerja dengan menggabungkan model dan teknis analisis dengan memasukan data yang ada dan berfungsi untuk mencari informasi.
5. Berfungsi untuk membantu proses penilaian pribadi pengambilan keputusan menjadi dasar pengambilan keputusan.
6. DSS merupakan sistem terkomputerisasi yang mengumpulkan dan menganalisis data, mensintesisnya untuk menghasilkan laporan informasi yang komprehensif.
7. DSS berbeda dari aplikasi operasi biasa, yang fungsinya hanya mengumpulkan data.
8. DSS memungkinkan pengambilan keputusan yang lebih terinformasi, penyelesaian masalah tepat waktu, dan peningkatan efisiensi dalam menangani masalah atau operasi, perencanaan, dan bahkan manajemen.
   * 1. **Jenis DSS**

Klafikasi DSS berbasis data menurut Power dalam (Limbong et al., 2020) dibagi menjadi dua jenis, yaitu:

1. DSS Berbasis Model

DSS berbasis model merupakan sistem mandiri dan tidak terhubung dengan sistem informasi perusahaan utama lainnya. Model yang kuat bersama dengan antar muka pengguna yang baik membuatnya mudah digunakan, membantu untuk melakukan analisis *what-if* dan hal lain yang serupa, membuat model simulasi, melakukan perencanaan, penjadwalan, dan membuat laporan dalam bentuk statistik.

1. DSS Berbasis Data

DSS berbasis data dapat menganalisis sejumlah data besar dari berbagai sumber yang disimpan di bank data. Bank data adalah *database* yang dapat menyimpan data sekarang dan data masa lalu yang diekstarksi dari berbagai sistem operasional, dan menyediakan alat pelaporan dan permintaan tertentu. Teknik utama sebagian besar digunakan dalam DSS yaitu untuk menganalisis data seperti *Online Analytical Processing* (OLAP) dan Data Mining (Rumondor dan Irawati, 2019) dalam (Limbong et al., 2020).

* + 1. **Keuntungan DSS**

DSS melaksanakan proses pengambilan keputusan secara terstruktur dan membantu pembuat keputusan mengidentifikasi solusi terbaik yang dapat diterima untuk masalah tertentu, DSS dapat lebih halus daripada sistem lainnya, berikut keuntungan dari DSS seperti:

1. Hemat waktu
2. Akurasi data meningkat
3. Strategi yang signifikan
4. Cepat dan terarah
5. Pengurangan biaya pengambil keputusan
   * 1. **Kekurangan DSS**

Salah satu kelemahan dari DSS adalah bahwa mereka dapat menghentikan pengguna dari berfikir dan dapat meningkatkan bias kognitif.

1. Pengguna dapat menerima informasi yang berlebihan, dan
2. Program DSS tanpa data yang cukup dapat membuat keputusan yang buruk.
   1. **Metode *Multi-Objective Optimization on the basis of Ratio Analysis* (MOORA)**
      1. **Pengertian Metode MOORA**

Menurut (Poningsih et al., 2020), *Metode Multi-Objective Optimization on the basis of Ratio Analysis* (MOORA) merupakan multiobjektif sistem mengoptimalkan dua atau lebih atribut yang saling bertentangan secara bersamaan. MOORA diperkenalkan oleh Brauers dan Zavadskas pada tahun 2006. Pada awalnya metode ini diperkenalkan oleh Brauers pada tahun 2004 sebagai “*Multi-Objective Optimation*” yang dapat digunakan untuk memecahkan berbagai masalah pengambilan keputusan yang rumit pada lingkungan pabrik. Metode ini diterapkan untuk memecahkan masalah ekonomi, manajerial, dan konstruksi pada sebuah perusahaan maupun proyek dengan perhitungan matematika yang kompleks.

Metode MOORA memiliki tingkat fleksibilitas dan kemudahan untuk dipahami dalam memisahkan bagian subjektif dari suatu proses evaluasi kedalam kriteria bobot keputusan dengan beberapa atribut pengambilan keputusan.

Metode ini memiliki tingkat selektifitas yang baik karena dapat menentukan tujuan dari kriteria yang bertentangan. Di mana kriteria dapat bernilai menguntungkan (*benefit*) atau yang tidak menguntungkan (*cost*).

* + 1. **Keunggulan Metode MOORA**

Metode MOORA sangat sederhana, stabil, dan kuat, bahkan metode ini tidak membutuhkan seorang ahli dibidang matematika untuk menggunakannya serta membutuhkan perhitungan matematis yang sederhana. Metode MOORA memiliki hasil yang lebih akurat dan tepat sasaran dalam membantu pengambilan keputusan. (Ramadani, Sihombing dan Parlina, 2019) dalam (Poningsih et al., 2020).

* + 1. **Langkah Penyelesaian Metode** **MOORA**

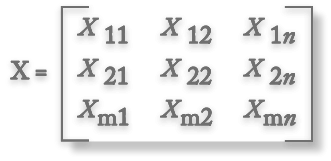
Metode MOORA terdiri dari lima langkah penyelesain menurut (Nofriansyah & Defit, 2017) adalah sebagai berikut:

1. Menginput nilai kriteria

Menentukan nilai krtiteria pada suatu alternatif dimana nilai tersebut nantinya akan diproses dan hasilnya akan menjadi sebuah keputusan.

1. Merubah nilai kriteria menjadi matriks keputusan.

Matriks keputusan berfungsi sebagai pengukuran kinerja dari alternatif I th pada atribut J th, (*m*) adalah alternatif dan (*n*) adalah jumlah atribut dan kemudian sistem rasio dikembangkan dimana setiap kinerja dari sebuah alternatif pada sebuah atribut dibandingkan dengan penyebut yang merupakan wakil untuk semua alternatif dari atribut tersebut. Berikut adalah perubahan nilai kriteria menjadi sebuah matriks kaputusan:

 Rumus …………… (2.1)

Keterangan:

*X* = nilai kriteria masing-masing kriteria yang direpresentasikan

sebagai matriks

1. Normalisasi matriks

Normalisasi bertujuan untuk menyatukan setiap elemen matriks sehingga elemen pada matriks memiliki nilai yang seragam. Normalisasi pada MOORA dapat dihitung menggunakan persamaan sebagai berikut.

 Rumus …………… (2.2)

Keterangan:

*Xij* = urutan ke (*i*) dari alternatif pada kriteria ke (*j*)

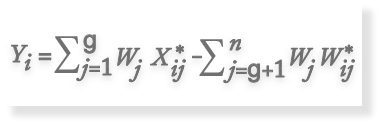
2 = menunjukan banyaknya jumlah alternatif ke-1 sampai ke

(*m*) yang dikuadratkan.

1. Mengurangi nilai maximax dan minimax.

Untuk menandakan bahwa sebuah atribut lebih penting itu bisa dikalikan dengan bobot yang sesuai (koefisien signifikasi). Atribut

bobot dipertimbangkan perhitungan menggunakan persamaan sebagai berikut:

 Rumus …………… (2.3)

Keterangan:

*Yi* = nilai dari penilaian normalisasi alternatif (*i*) terhadap

semua atribut.

*Xij* = urutan ke (*i*) dari alternatif pada kriteria ke (*j*)

*Wj* = nilai bobot setiap kriteria

*g*  = nilai kriteria yang akan dimaksimalkan, (n-g) adalah

nilai dari kriteria yang diminimalkan

*j*  = nilai kriteria

1. Menentukan rangking dari hasil perhitungan MOORA.

Nilai (*Yi*) dapat menjadi positif atau negatif tergantung dari total maksimal (attribut yang menguntungkan) dalam matriks keputusan. Sebuah urutan peringkat dari (*Yi*) menunjukkan pilihan terahir. Dengan demikian alternative terbaik memiliki nilai (*Yi*) tertinggi sedangkan alternative terburuk memiliki nilai (*Yi*) terendah.

* 1. **Penerimaan Siswa Baru (PSB)**

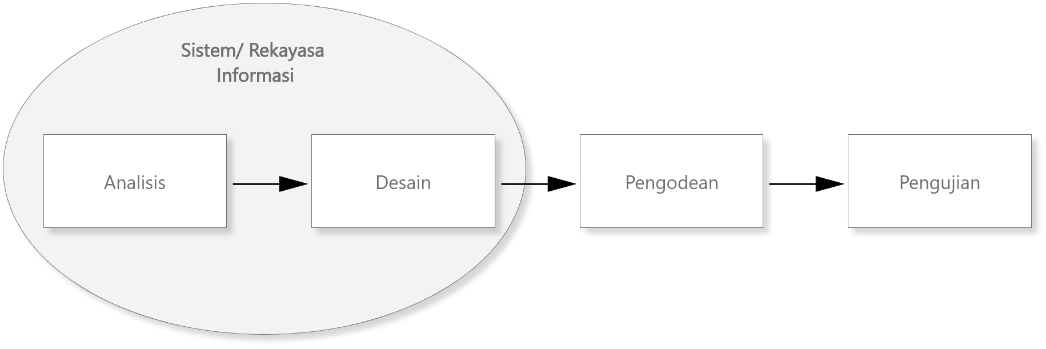
Penerimaan merupakan proses penyambutan, tindakan atau sikap terhadap seseorang, siswa merupakan pelajar pada akademi atau perguruan tinggi, sedangkan baru merupakan sesuatu yang tidak ada sebelumnya. Jadi penerimaan siswa baru atau biasa disebut (PSB) adalah kegiatan menerima dan memilih calon peserta pendidikan dan pelatihan di sekolah. PSB merupakan kegiatan rutin tahunan yang harus dilalui peserta didik didalam penyaringan objek-objek pendidikan oleh semua sekolah di bawah pengawasan dan koordinasi Dinas Pendidikan (Santoso & Sembiring, 2021).

* 1. **Alat Bantu Analisa Dan Perancangan Sistem**
     1. ***System Development Lyfe Cycle* (SDLC)**

SDLC menurut (Sukamto & Shalahuddin, 2018) adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya.

* + 1. ***Waterfall***

Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau turut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*).

****Berikut adalah tahapan metode *waterfall* dapat dilihat pada Gambar 2.5:

**Gambar 2.5.** Ilustrasi Model *Waterfall*

(Sumber: Sukamto & Shalahuddin, 2018)

1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Kebutuhan perangkat lunak pada tahap ini perlu di dokumentasikan.

1. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak, representasi antar muka, dan prosedur pengodean. Tahap ini memenuhi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat di implementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu di dokumentasikan.

1. Pembuatan Kode Program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

1. Pengujian

Penguijan fokus pada perangkat lunak secara dari segi *logic* dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan *output* yang dihasilkan sesuai dengan yang diinginkan.

1. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkatlunak baru.

* + 1. ***Flowmap***

Menurut (Hidayat, 2017). *Flowmap* atau bagan alir adalah alur kegiatan penyajian dokumen yang berhubungan antara suatu entitas lainnya dengan tujuan untuk menghasilkan sebuah informasi. Adapun symbol yang digunakan dalam *flowmap* terlihat pada Tabel 2.1:

**Tabel 2.1.** Simbol-Simbol *Flowmap*

|  |  |  |
| --- | --- | --- |
| No. | Nama dan Simbol | Keterangan |
| 1. | Terminator | Permulaan/ akhir program |
| 2. | Garis alir | Arah aliran program |
| 3. | Proses manual | Proses pengolahan data secara manual |
| 4. | *Direct acces storage* | Menyimpan data |
| 5. | Proses | Proses pengolahan data |
| 6. | *Input/ output* data | Proses *input/ output* data, parameter, informasi |
| 7. | *Decision* | Perbandingan pernyataan atau pilihan |
| 8. | *Document* | Simbol untuk data yang berbentuk informasi |

(Sumber : James dalam (Hidayat, 2017)).

* + 1. ***Flowchart***

Menurut(Liswati, 2020)*, Flowchart* dalam bahasa Indonesia sering disebut dengan diagram alir. *Flowchart* adalah bagan atau gambar yang memperlihatkan urutan prosedur dan hubungan antarproses disertai intruksinya. Bagan tersebut biasanya dinyatakan dengan simbol. Simbol menggambarkan proses tertentu, sedangkan hubungan antarproses digambarkan dengan garis penghubung. Berikut ini beberapa jenis *flowchart*:

1. *System Flowchart* (Bagan alir sistem)

Bagan alir sistem dapat didefinisikan sebagai bagan yang menunjukan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan-urutan dan prosedur-prosedur yang ada di dalam sistem. Bagan alir sistem menunjukan apa yang dikerjakan di sistem.

1. *Document Flowchart* (Bagan alir dokumen)

Bagan alir dokumen atau disebut juga bagan alir formulir (*form flowchart*) atau *paperwork flowchart* merupakan bagan alir yang menunjukan arus dari laporan dan formulir termasuk tembusan-tembusannya.

1. *Schematic Flowchart* (Bagan alir skematik)

Bagan alir skematik merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem. Perbedaannya adalah bagan alir skematik selain menggunakan simbol-simbol bagan alir sistem, juga menggunakan gambar-gambar komputer dan peralatan lainnya yang digunakan.

1. *Program Flowchart* (Bagan alir program)

Bagan alir program merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Bagan alir program dibuat dari derivikasi bagan alir program.

1. *Process Flowchart* (Bagan alir proses)

Bagan alir proses merupakan bagan alir yang banyak digunakan di teknik industri. Bagan alir ini juga berguna bagi analis sistem untuk menggambarkan proses dalam suatu prosedur.

Simbol-simbol pada *flowchart* menurut (Sari, 2017), dapat dilihat pada Tabel 2.2:

**Tabel 2.2.** Simbol-Simbol *Flowchart*

|  |  |  |
| --- | --- | --- |
| No. | Simbol | Keterangan |
| 1. | *Input/Output* | Mempresentasikan i*nput* data atau *output* data yang diproses atau Informasi |
| 2. | Proses | Mempresentasikan operasi |
| 3. | Penghubung | Keluar ke atau masuk dari bagan lain *flowchart* khususnya halaman yang sama |
| 4. | Anak Panah | Mempresentasikan alur kerja |
| 5. | Penjelasan | Untuk komentar tambahan |
| 6. | Keputusan | Keputusan dalam program |
| 7. | *Predefined Process* | Rincian operasi berada di tempat lain |
| 8. | Preparation | Pemberian harga awal |
| 9. | Terminal *Points* | Awal/ akhir *flowchart* |
| 10. | *Punched Card* | *Input/ output* yang menggunakan kartu berlubang |
| 11. | Dokumen | I/O dalam format yang dicetak |
| 12. | *Magnetic Tape* | I/O yang menggunakan pita magnetik |
| 13. | *Magnetic Disk* | I/O yang menggunakan disk magnetik |
| 14. | *Magnetic Drum* | I/O yang menggunakan drum magnetik |
| 15. | *On-line Storage* | I/O yang menggunakan penyimpanan akses langsung |
| 16. | *Punched Tape* | I/O yang menggunakan pita kertas berlubang |
| 17. | Manual *Input* | *Input* yang dimasukan secara langsung dari *keyboard* |
| 18. | *Display* | *Output* yang ditampilkan pada terminal |
| 19. | Manual *Operation* | Operasi Manual |
| 20. | *Communication Link* | Tranmisi data melalui *channel* komunikasi, seperti telepon |
| 21. | *Off-line Storage* | Penyimpanan yang tidak dapat diakses oleh komputer secara langsung |

(Sumber : Sari, 2017)

* + 1. ***Entity Relationship Diagram* (ERD)**

ERD digunakan untuk pemodelan basis data relasional. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow’s Foot, dan beberapa notasi lain (Sukamto & Shalahuddin, 2018). Berikut simbol-simbol ERD yang digunakan dengan notasi Chen pada Tabel 2.3:

**Tabel 2.3.** Simbol-Simbol ERD

|  |  |  |
| --- | --- | --- |
| No. | Nama dan Simbol | Keterangan |
| 1. | Entitas/*entity* | Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya adar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama *table*. |
| 2. | Atribut | *Field* atau kolom data yang butuh disimpan dalam suatu entitas |
| 3. | Atribut kunci primer | *Field* atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses *record* yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama) |
| 4. | Atribut multi nilai/ *multivalue* | *Field* atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu |
| 5. | Relasi | Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja |
| 6. | Asosiasi/ *association* | Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki *multiplicity* kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan *one to many* menghubungkan entitas A dan entitas B |

(Sumber: Sukamto & Shalahuddin, 2018)

ERD ini diperlukan agar dapat menggambarkan hubungan antar entitas dengan jelas, dapat menggambarkan batasan jumlah entitas dan partisipasi antar entitas, mudah dimengerti pemakai dan mudah disajikan oleh perancang *database*.

* + 1. ***Conceptual Data Model* (CDM)**

CDM adalah konsep yang berkaitan dengan pandangan pemakai terhadap data yang disimpan dalam basis data (Sukamto & Shalahuddin, 2018). Pada CDM tidak terdapat tipe data yang menggambarkan hubungan antara tabel untuk keperluan implementasi ke basis data. CDM hasil penjabaran dari ERD, berikut simbol-simbol CDM ditampilkan pada Tabel 2.4 dan aturan konversi ERD menjadi CDM pada Tabel 2.4:

**Tabel 2.4.** Simbol-simbol CDM

|  |  |  |
| --- | --- | --- |
| No. | Nama dan Simbol | Keterangan |
| 1. | Entitas/ tabel | entitas atau tabel yang menyimpan data dalam basis data |
| 2. | Relasi | relasi antar tabel yang terdiri atas nama relasi dan *multiplicity* |

(Sumber: Sukamto & Shalahuddin, 2018)

**Tabel 2.5.** Aturan Konversi ERD menjadi CDM

|  |  |  |
| --- | --- | --- |
| No. | ERD | CDM |
| 1. | relasi dengan kardinalitas *many to many* | Menjadi sebuah tabel sendiri dengan kunci primer (*primary key*) adalah atribut yang menjadi kunci primer di kedua entitas yang direlasikannya |
| 2. | relasi dengan kardinalitas *one to many* | Kunci primer entitas yang memiliki hubungan *one* akan dijadikan kunci primer di entitas yang memiliki hubungan many dengan kata lain, relasi tidak menjadi tabel sendiri |
| 3. | relasi dengan kardinalitas *one to one* | Kunci primer salah satu entitas akan dijadikan kunci asing (*forigen key*) pada tabel yang lain dan kunci asing itu dijadikan kunci primer juga, dengan kata lain, relasi tidak menjadi tabel sendiri. |

(Sumber: Sukamto & Shalahuddin, 2018)

* + 1. ***Physical Data Model* (PDM)**

PDM adalah model yang menggunakan sejumlah tabel untuk menggambarkan data serta hubungan antara data. Pada setiap tabel mempunyai kolom yang memiliki nama unik dan tipe data, PDM sudah berbentuk fisik perancangan basis data yang siap diimplementasikan ke dalam DBMS sehingga nama tabel sudah merupakan nama asli tabel yang akan diimplementasikan ke dalam DBMS. Berikut simbol-simbol PDM dapat dilihat pada Tabel 2.6:

**Tabel 2.6.** Simbol-simbol PDM

|  |  |  |
| --- | --- | --- |
| No. | Nama dan Simbol | Keterangan |
| 1. | Tabel | Tabel yang menyimpan data dalam basis data |
| 2. | Relasi | Relasi antar tabel yang terdiri dari persamaan antara *primary key* (kunci primer) tabel yang diacu dengan kunci yang menjadi referensi acuan di tabel lain. |

(Sumber: Sukamto & Shalahuddin, 2018)

* + 1. ***Data Flow Diagram* (DFD)**

Menurut (Sukamto & Shalahuddin, 2018), DFD adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*).

Notasi DFD memiliki dua pendapat, yaitu menurut Gane/Sarson dan Yourdon/De Marco. Notasi yang dimaksud dapat dilihat pada Tabel 2.7:

**Tabel 2.7.** Simbol-Simbol DFD Gane/Sarson danYourdon De Marco

|  |  |  |  |
| --- | --- | --- | --- |
| No | Gane/Sarson | Yourdon/De Marco | Keterangan |
| 1. | Proses |  | Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah  yang harusnya menjadi fungsi atau prosedur di dalam kode program  Catatan:  Nama yang diberikan pada sebuah proses biasanya berupa kata kerja |
| 2. | Penyimpanan Data |  | File atau basis data atau penyimpanan (*storage*); pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel pada basis data (*Entity Relationalship Diagram* (ERD), *Conceptual Data Model* (CDM), *Physical Data Model* (PDM))  Catatan:  Nama yang diberikan pada sebuah penyimpanan biasanya kata benda |
| 3. | Entitas Luar |  | Entitas luar (*external entity*) atau masukan (*input*) atau keluaran (*output*) atau orang yang memakai/berinteraksi dengan perangkat lunak ynag dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan  Catatan:  Nama yang digunakan pada masukan (*input*) atau keluaran (*output*) berupa kata benda |
| 4. | Aliran Data |  | Aliran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (*input*) atau keluaran (*output*)  Catatan:  Nama yang digunakan pada aliran dara biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data siswa” atau tanpa kata data misalnya “siswa” |

(Sumber: Sukamto & Shalahuddin, 2018)

Rosa, Shalahudin (2018:72) menjelaskan tentang tahapan-tahapan perancangan dengan menggunakan DFD yaitu:

1. Membuat DFD Level 0 atau sering disebut juga *Context Diagram*

DFD Level 0 menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. DFD Level 0 digunakan untuk menggambarkan interaksi antara sistem yang akan dikembangkan dengan entitas luar.

1. Membuat DFD Level 1

DFD Level 1 digunakan untuk menggambarkan modul-modul yang ada dalam sistem yang akan dikembangkan. DFD Level 1 merupakan hasil *breakdown* DFD Level 0 yang sebelumnya sudah dibuat.

1. Membuat DFD Level 2

Modul-modul pada DFD Level 1 dapat di-*breakdown* menjadi DFD Level 2. Modul mana saja yang harus di-*breakdown* lebih detail tergantung pada kedetailan modul tersebut. Apabila modul tersebut sudah cukup detail dan runci maka modul tersebut sudah tidak perlu di-*breakdown* lagi. Untuk sebuah sistem, jumlah DFD Level 2 sama dengan jumlah modul pada DFD Level 1 yang di-*breakdown*.

1. Membuat DFD Level 3 dan seterusnya

DFD Level 3,4,5, dan seterusnya merupakan *breakdown* dari modul pada DFD Level di-atasnya. *Breakdown* pada level 3,4,5, dan seterusnya aturannya sama persis dengan DFD Level 1 atau 2.

* 1. ***Internet***

Menurut (Hidayatullah & Kawistara, 2017) “Internet adalah jaringan global yang menghubugkan komputer-komputer di seluruh dunia. Dengan internet, sebuah komputer bisa mengakses data yang terdapat pada komputer lain di benua yang berbeda. Dengan internet, sebuah toko online bisa tetap terbuka selama 24 jam sehari dan 7 hari seminggu tanpa henti. Dengan intenet, kejadian penting yang terjadi di suatu negara bisa segera diketahui oleh orang lain di negara yang berbeda”.

* 1. ***Website***

Menurut (Abdulloh, 2018) mengemukan bahwa, “*website* adalah sekumpulan halaman yang berisi informasi data digital baik teks, gambar, animasi, suara dan video atau gabungan dari semuanya yang disediakan melalui jalur koneksi internet sehingga dapat diakses dan dilihat oleh semua orang diseluruh dunia”.

Jenis *website* dibagi menjadi 3 yaitu *website* statis, dinamis dan interaktif (Abdulloh, 2018).

1. *Website* Statis yaitu jenis *website* yang isinya tidak diperbarui secara berkala, sehingga isinya dari waktu ke waktu akan selalu tetap.
2. *Website* Dinamis yaitu jenis *website* yang isinya terus diperbaharui secara berkala oleh pengelola atau pemilik *website*.
3. *Website* Interaktif pada dasarnya termasuk dalam kategori website dinamis, dimana isi informasinya selalu diperbaharui dari waktu ke waktu.
   1. ***Web Server***

Menurut (Hidayatullah & Kawistara, 2017) “*Web Server* adalah tempat dimana anda menyimpan aplikasi *web* Anda kemudian mengaksesnya melalui Internet. Setiap perubahan, kecil maupun besar, anda *upload* ke *web server* baru setelah itu periksa apakah perubahan itu sudah selesai dengan yang anda inginkan atau belum. Selain itu dibutuhkannya *web server* ini adalah karena untuk *server side script seperti* PHP, pemeriksaan baru akan tampil jika menggunakan *web server*”.

Sedangkan menurut (Abdulloh, 2018) mengatakan bahwa “*Web server* merupakan perangkat lunak yang berfungsi untuk menerima permintaan (*request*) melalui *protocol* HTTP atau HTTPS dari *client* kemudian mengirimkan kembali dalam bentuk halaman – halaman *web*.

Dari pengertian di atas dapat disimpulkan bahwa *web server* adalah perangkat lunak yang berfungsi menerima layanan HTTP atau HTTPS melalui *web browser* dan menampilkan hasil berupa halaman – halaman *web*.

* 1. ***Web Browser***

Menurut (Hidayatullah & Kawistara, 2017) *Web browser* adalah program untuk menampilkan halaman web yang dapat dilihat oleh web cilients. Sedangkan menurut (Abdulloh, 2018) *Web browser* digunakan untuk menampilkan dan mengetes hasil program.

Maka dapat disimpulkan bahwa *web browser* merupakan perangkat lunak untuk menanpilkan halaman *website* yang dapat di lihat oleh *clients*.

* 1. ***Hypertext Transfer Protocol* (HTTP)**

Menurut (Hidayatullah & Kawistara, 2017) “HTTP adalah protocol agar cilent dan server bisa berkomunikasi dengan gaya request-response. HTTP menentukan bagaimana format pesan dan bagaimana cara pengirimannya, serta bagaimana web server dan browser beraksi dan bereaksi terhadap berbagai perintah”.

* 1. **Bahasa Pemograman**

Bahasa pemrograman digunakan sebagai instruksi manusia kepada komputer dalam membuat *website* untuk mengelola data sesuai dengan apa yang diharapkan. Berikut ini merupakan bahasa pemrograman yang digunakan penulis untuk membuat perangkat lunak berbasis *web*.

* + 1. ***Hypertxt Markup Language* (Html)**

Menurut (Abdulloh, 2018), “HTML merupakan singkatan dari *Hypertxt Markup Language* yaitu bahasa standar *web* yang dikelola penggunaanya oleh W3C (*World Wide Web Consortium*) berupa tag-tag yang menyusun setiap elemen dari *website*”.

Sedangkan menurut (Hidayatullah & Kawistara, 2017), “HTML adalah bahasa standard yang digunakan untuk menampilkan halaman *web*”. Dan yang bisa dilakukan dengan HTML yaitu:

1. Mengatur tampilan dari halaman *web* dan isinya.
2. Membuat tabel dalam halaman *web*.
3. Mempublikasikan halaman *web* secara *online*.
4. Membuat *form* yang bisa digunakan untuk menangani registrasi dan transakasi *via web*.
5. Menampilkan area gambar (*canvas*) di *browser*.

Pengertian di atas dapat disimpulkan bahwa HTML adalah sebuah file dokumen yang berisikan tag, beberapa elemen dan atribut untuk menampilkan halaman pada *web browser*.

* + 1. ***PHP:Hypertext Preprocessor***

Menurut (Hidayatullah & Kawistara, 2017) dalam bukunya yang berjudul Pemrograman *Web* mengemukakan bahwa “PHP adalah suatu bahasa *scripting* khususnya digunakan untuk *web development*”. PHP memiliki sifat *server side scripting* sehingga untuk menjalankan PHP harus menggunakan *web server*.

Sedangkan menurut (Abdulloh, 2018), “PHP *Hypertext Preprocessor* yaitu bahasa pemrograman *web* yang dapat disisipkan dalam skrip HTML dan bekerja di sisi *server*.”

Maka dapat disimpulkan bahwa PHP merupakan bahasa pemrograman

*scripting* yang terpasang pada HTML yang bertujuan untuk merancang halaman *website* yang sepenuhnya dijalankan oleh *server*.

* + 1. **Jquery**

Menurut (Abdulloh, 2018) menjelaskan bahwa “JQuery merupakan salah satu dari banyak javascript library yang fungsi JavaScript sudah siap pakai. Sehingga mempermudah dan mempercepat dalam membuat kode JavaScript. Dengan menggunakan JQuery skrip yang panjang dapat diangkat menjadi beberapa baris kode saja”.

Sedangkan menurut (Hidayatullah & Kawistara, 2017) dalam bukunya yang berjudul Pemrograman *Web* mengatakan bahwa “JQuery adalah Library Javascript yang akan menjadikan *web* anda lebih bagus dalam hal *user interface*, lebih stabil, dan dapat mempercepat waktu dan kinerja anda dalam membuat *web* karena anda hanya perlu memanggil fungsinya saja tanpa harus membuatnya dari awal”.

Pengertian di atas dapat disimpulkan bahwa JQuery adalah kumpulan fungsi-fungsi Javascript dalam library yang sudah dibentuk menjadi suatu *framework*, sehingga tampilan *web* menarik.

* + 1. **Javascript**

Javascript merupakan bahasa pemrograman *web* yang pemrosesannya dilakukan di sisi *client*. Karena berjalan di sisi *client*. Javascript dapat dijalankan hanya dengan menggunakan *browser* (Abdulloh, 2018).

* + 1. ***Cascading Style Sheet* (CSS)**

Menurut (Abdulloh, 2018) CSS adalah singkatan dari *Cascading Style Sheet* yaitu dokumen *web* yang berfungsi mengatur elemen HTML dengan berbagai properti yang tersedia sehingga dapat tampil dengan berbagai gaya yang diinginkan. Sebagian orang menganggap CSS bukan termasuk salah satu bahasa pemrograman karena memang strukturnya yang sederhana, hanya berupa kumpulan – kumpulan aturan yang mengatur *style* elemen HTML.

Sedangkan menurut (Hidayatullah & Kawistara, 2017). “CSS berfungsi untuk membuat tampilan lebih rapih dan menarik”.

Ada 4 cara untuk memasang kode CSS ke dalam HTML/ Halaman *Web* (Hidayatullah & Kawistara, 2017) yaitu :

1. *Inline style sheet*

Cara ini adalah menuliskan langsung *script* CSS ke dalam *tag* HTMLyang diinginkan. Cara ini sebaiknya hanya digunakan jika ingin memformat suatu elemen satu kali saja.

1. *Internal style sheet*

Ini dilakukan dengan cara memasang kode CSS berada pada file HTML bersangkutan namun diletakan khusus pada suatu tempat yaitu diantara *tag <head>* dan *</head>.*

1. Me-*link* ke *external* CSS

Kode CSS *external* ditulis dalam suatu *file* terpisah yang disimpan dengan ekstensi .css. Kemudian harus memanggil file CSS tersebut pada halaman *web* yang memerlukannya.

1. *Import* CSS file
2. Meng-*import* CSS kedalam suatu *file* CSS yang lain mernggunakan *tag import*.

Pengertian di atas dapat disimpulkan bahwa CSS adalah kumpulan aturan yang dikembangkan untuk mempercantik halaman *web*.

* 1. ***Software* Pendukung** 
     1. ***Visual Studio Code* (VS Code)**

VS Code adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi *multiplatform* yang mendukung bahasa pemrograman Javascript, Typescript, dan Node.js, serta bahasa pemrograman lainya dengan bantuan plugin di *marketplace* VS Code (Ummy Gusti Salamah & Indonesia, 2021).

**Gambar 2.6.** Logo *Visual Studio Code*

(Sumber: Ummy Gusti Salamah & Indonesia, 2021)

* + 1. ***Power Designer***

*Power Designer* juga dimaksudkan untuk pembuatan arsitektur informasi dan arsitektur interprise. Tool pemodelan yang dikeluarkan oleh *Sybase* untuk membangun sebuah sistem informasi yang cepat, terstruktur dan efektif (Mohamad et al., 2020). Beberapa pendukung pemodelan *Power Designer* sebagai berikut:

1. Requirement Management
2. Bussiness Process
3. Data modelling
4. XML Modelling
5. Information
   * 1. ***Adobe Experience Design CC***

Adobe XD atau juga dikenal dengan sebutan *Adobe Experience Design CC*. Adobe XD merupakan perangkat lunak desain yang berfokus pada desain *user interface* (UI) dan *user experience* (UX), memiliki fitur tambahan seperti plugin untuk mempermudah pengguna dalam berbagai kebutuhan. Adobe XD merupakan aplikasi desktop untuk designer dalam merancang desain *user interface*, *prototype user flows* dan interaksi, dan dapat juga untuk meng-*export* aset gambar untuk pengembangan aplikasi dan situs *web* (Schwarz, 2017).

**Gambar 2.7.** Logo Adobe XD

(Sumber: https://www.adobe.com/)

* + 1. **MySQL**

Menurut (Hidayatullah & Kawistara, 2017). “MySQL adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan oleh para pemrogram aplikasi *web*. Contoh DBMS lainnya adalah: PostgreSQL (*freeware*), SQL Server, My Access dari Microsoft, DB2 dari IBM, Oracle dan Oracle Corp, Dbase, FoxPro”.

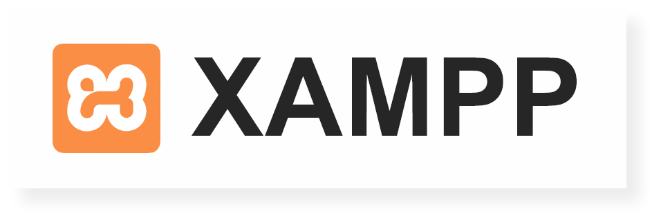
**Gambar 2.8** Logo MySQL

(Sumber: <https://www.mysql.com/>)

* + 1. **XAMPP**

Xampp adalah perangkat lunak (*free software*) bebas, yang mendukung untuk banyak sistem operasi, yang merupakan komplilasi dari beberapa program. Fungsi Xampp sendiri adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri beberapa program antara lain: Apache HTTP *Server*, MySQL *database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl (Haqi, 2019).

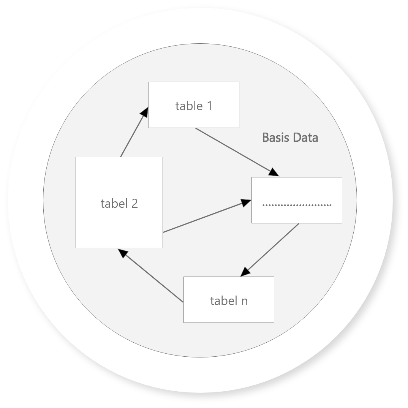
Menurut (Hidayatullah & Kawistara, 2017) mengemukakan bahwa kata Xampp sendiri berasal dari:

1. X yang berarti *cross platform* karena XAMPP bisa dijalankan di Windows, l Linux, Mac dan lain sebagainya.
2. A yang berarti Apache sebagai *web server*-nya
3. M yang berarti MySQL sebagai *Database Management System* (DBMS)-nya.
4. PP yang berarti PHP dan Perl sebagai bahasa yang didukungnya.

**Gambar 2.9** Logo XAMPP

(Sumber: https://www.apachefriends.org/)

* 1. **Basis Data**
     1. **Pengertian Basis Data**

Sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan atau media untuk menyimpan data agar dapat diakses dengan mudah dan cepat (Sukamto & Shalahuddin, 2018). Implementasi basis data relasional dengan tabel-tabel yang saling memiliki relasi pada Gambar 2.10:

**Gambar 2.10.** Basis Data Relasional

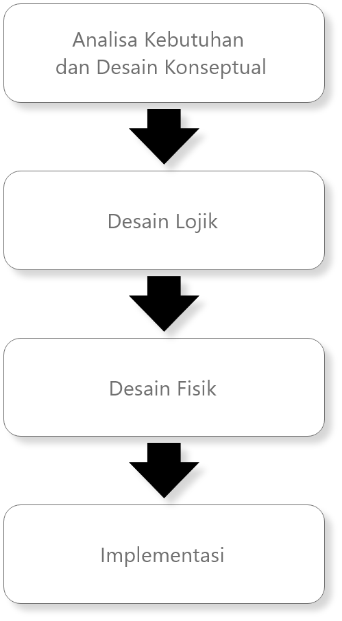
(Sumber: Sukamto & Shalahuddin, 2018)

* + 1. **Database Management System (DBMS)**

DBMS menurut (Sukamto & Shalahuddin, 2018)**.** Suatu sitem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data. Suatu sistem aplikasi bisa disebut DBMS jika memenuhi persyaratan sebagai berikut:

1. Menyediakan fasilitas untuk mengelola akses data
2. Mampu menangani integritas data
3. Mampu menangani akses data, dan
4. Mampu menangani *backup* data
   * 1. **Alur Hidup Basis Data**

Menurut (Sukamto & Shalahuddin, 2018). Dalam membuat perencanaan basis data juga memiliki alur hidup atau *Database Life Cycle* (DBLC). *Fase-fase* DBLC dapat dilihat pada Gambar 2.11:

**Gambar 2.11.** Alur Hidup Basis Data

(Sumber: Sukamto & Shalahuddin, 2018)

Pada Gambar 2.11 alur hidup basis data terdiri atas lima *fase* penggabungan, dimulai dari analisis kebutuhan dan desain konseptual, diakhiri dengan implementasi.

1. Analisa kebutuhan/ *requirement analysis* hal-hal yang harus dilakukan pada tahap ini:
2. Didefinisikan dengan mewawancarai produsen dan pemakai data, data apa sajakah yang butuh untuk disimpan dan terkait dengan aplikasi komputer yang dikembangkan
3. Membuat kontrak spesifikasi basis data
4. ERD sebagai bagian dari desain konseptual
5. Desain lojik basis data/ *logycal database design*, pada tahap ini harus dibuat rancangan lojik basis data, dan CDM.
6. Dessain fisik basis data/ *physical database design*, pada tahapan ini harus dibuat rancangan fisik basisi data, dan PDM.
7. Implementasi
   * + 1. Membuat Query SQL
       2. Aplikasi ke DBMS atau *file*
   1. **Struktur Navigasi**

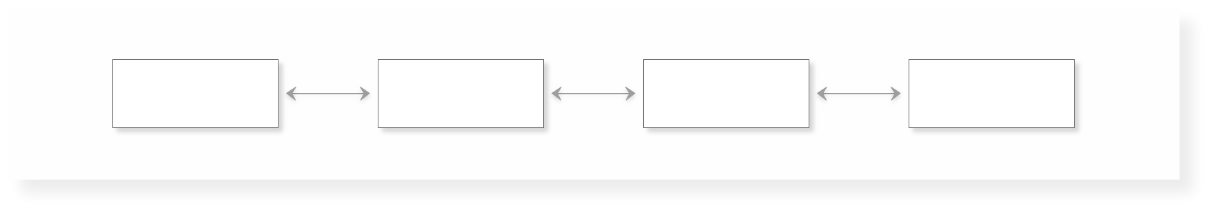
Untuk mendukung pembuatan aplikasi *web* ini, hal pertama yang akan dilakukan adalah menentukan struktur navigasi. Struktur navigasi dapat dikatakan sebagai penggambaran dari hubungan atau rantai dari seluruh elemen yang akan digunakan dalam aplikasi.

Struktur navigasi struktur navigasi adalah gabungan dari struktur referensi informasi situs *web* dan mekanisme *link* yang mendukung pengunjung untuk melakukan penjelajahan situs. Struktur navigasi menghubungkan halaman *web* yang satu dengan yang lainnya agar memudahkan kita dalam menjelajah sebuah situs *web* (Binanto, 2010).

Ada empat struktur dasar yang digunakan yaitu:

1. Struktur navigasi *linier*

Pengguna akan melakukan navigasi secara berurutan, dari *frame* atau *byte* informasi yang satu ke yang lainya.

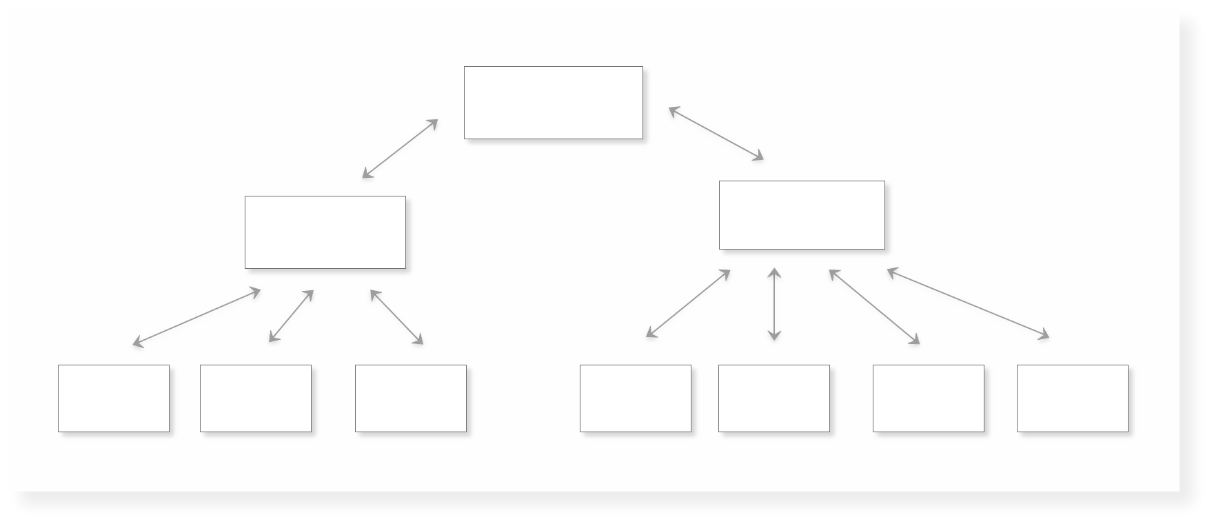
****

**Gambar 2.12.** Struktur Navigasi Linier

(Sumber: Binanto, 2010)

1. Struktur navigasi *hierarkis*

Struktur dasar ini disebut juga struktur “*liniear* dengan percabangan”, karena pengguna melakukan navigasi disepanjang cabang pohon terstruktur yang terbentuk oleh logika isi.

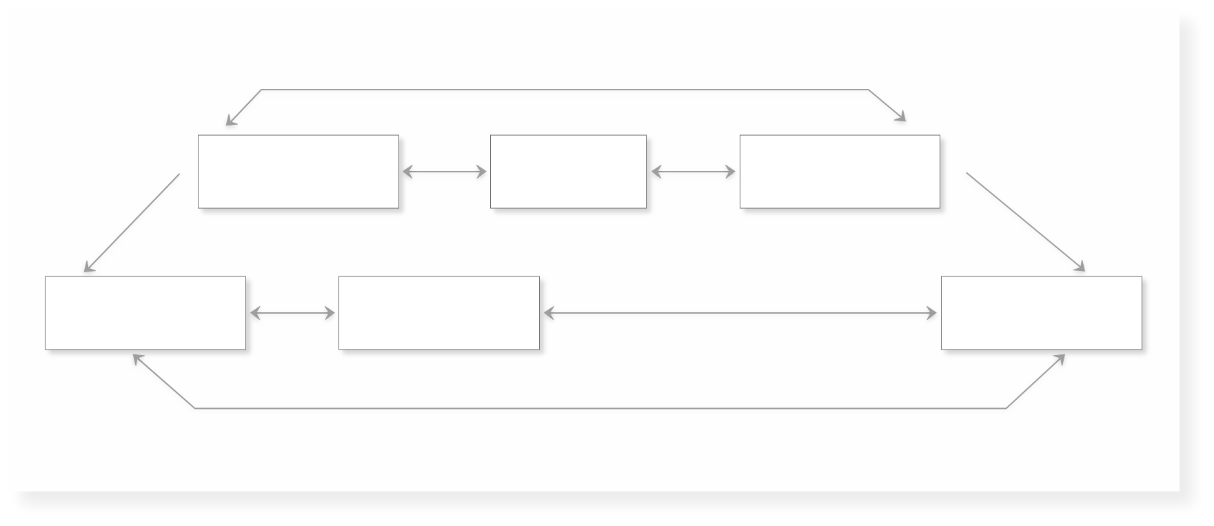
****

**Gambar 2.13.** Struktur Navigasi Hirarki

(Sumber: Binanto, 2010)

1. Struktur navigasi *nonlinier*

Pengguna akan lebih melakukan navigasi dengan bebas melalui isi proyek dengan tidak terikat dengan jalur yang sudah ditentukan sebelumnya.

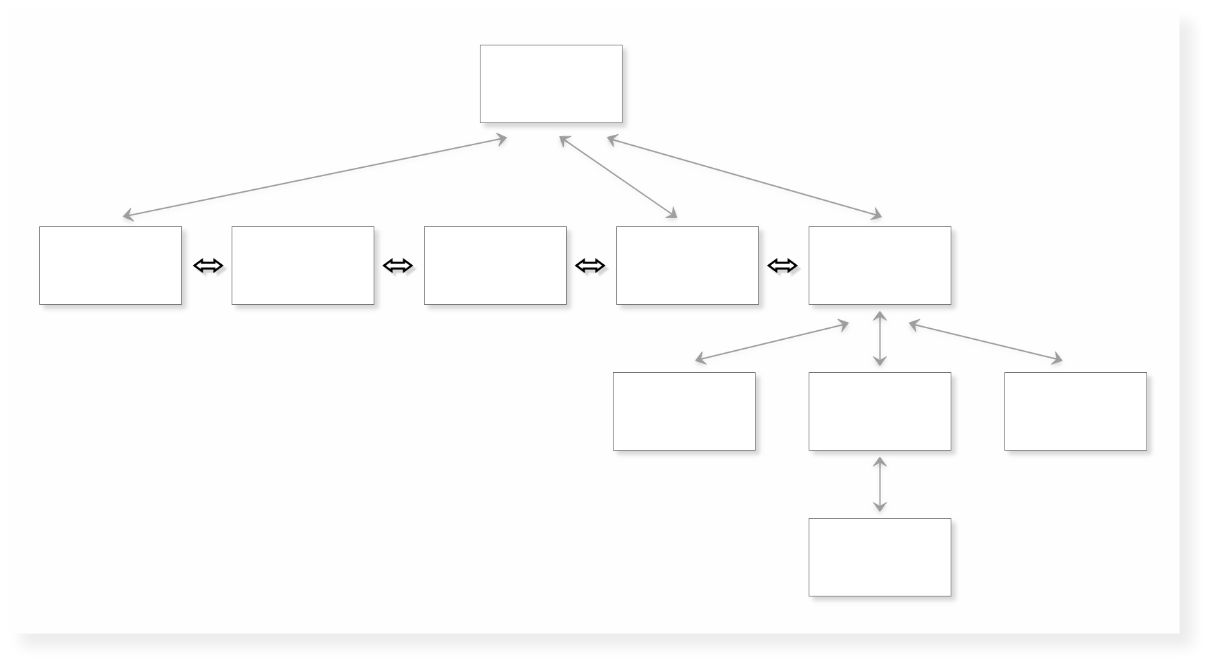
****

**Gambar 2.14.** Struktur Navigasi *Nonlinier*

(Sumber: Binanto, 2010)

1. Struktur navigasi komposit

Pengguna akan melakukan navigasi dengan bebas (secara non-liniear), tetapi terkadang dibatasi persentasi *linear* film atau informasi penting dan pada data yang paling terorganisasi secara logis pada suatu hirarki.

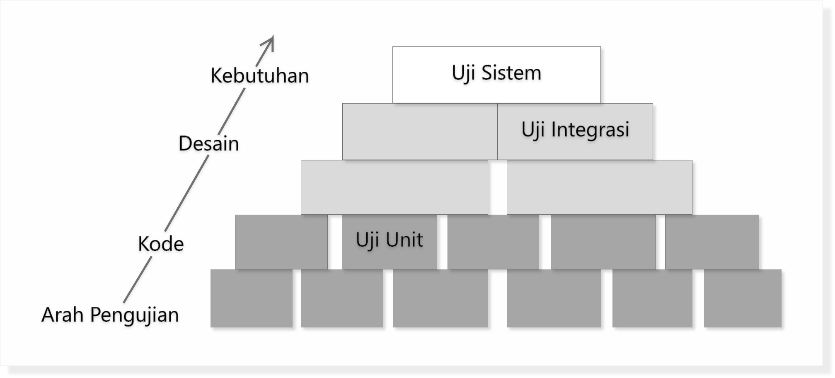


**Gambar 2.15.** Struktur Navigasi Komposit

(Sumber: Binanto, 2010)

* 1. **Pengujian Perangkat Lunak**

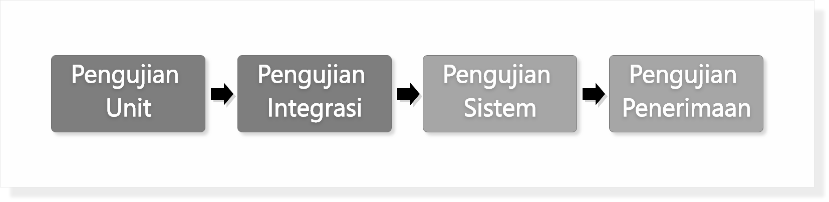
Menurut (Sukamto & Shalahuddin, 2018). Pengujian perangkat lunak adalah sebuah elemen, topik yang memiliki cakupan luas dan sering dikaitkan dengan verifikasi (*verification*) dan validasi (*validation*) (V&V). Verifikasi mengacu pada kumpulan aktifitas perangkat lunak dengan menerapkan fungsi yang spesifik, validasi mengacu pada sekumpulan aktifitas yang berbeda yang menjamin bahwa perangkat lunak yang dibangun dapat ditelususri sesuai dengan kebutuhan.

Tahap pengujian untuk verifikasi dilakukan mulai dari lingkup yang kecil naik ke lingkup yang besar seperti pada Gambar 2.16:

**Gambar 2.16.** Hiraki Pengujian Sistem

(Sumber: Sukamto & Shalahuddin, 2018)

Gambar di atas menunjukan pengujian perangkat lunak di awali dengan pengujian unit, setelah pengujian unit selesai dilanjutkan pengujian integrasi, dan yang terakhir pengujian sistem. Pengujian ini biasanya juga dilengkapi dengan pengujian penerima, dimana pengujian ini dilakukan untuk mengetahui tingkat kepuasan pelanggan terhadap perangkat lunak yang dibuat. Tahapan pengujian keseluruhan dapat dilihat pada Gambar 2.17:



**Gambar 2.17.** Pengujian Perangkat Lunak

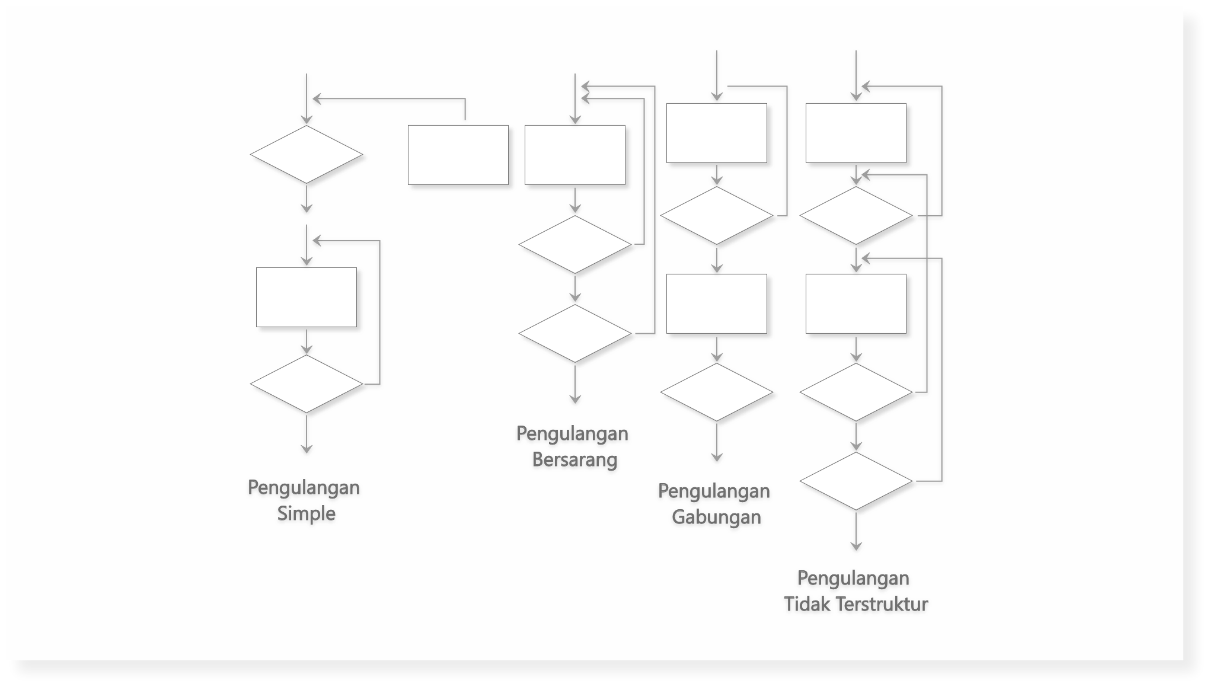
(Sumber: Sukamto & Shalahuddin, 2018)

Tahap pengujian untuk validasi memiliki beberapa pendekatan sebagai berikut:

1. Pengujian kotak hitam (*Black-Box Testing*)

Menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Apakah fungsi-fungsi masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

1. Pengujian kotak putih (*White-Box Testing*)

Menguji perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi-fungsi, masukan dan keluaran yang sesuai dengan spesifikasi kebutuhan. Pengujian kotak putih dilakukan dengan memeriksa lojik dari kode program, dapat dilihat ilustrasi pengujian kotak putih pada gambar 2.18:

**Gambar 2.18.** Ilustrasi Pengujian Kotak Putih

(Sumber: Sukamto & Shalahuddin, 2018)

* 1. **Tinjauan Studi**

Pada tinjauan studi ini merupakan penelitian yang ada sebelumnya dan digunakan sebagai acuan atau rujukan dalam penelitian yang akan dilakukan, terdapat persamaan dari objek penelitian yaitu sama-sama banyak meneliti tentang sistem penunjang keputusan seputar penerimaan mahasiswa atau siswa. Serta menghasilkan keputusan berdasarkan perhitungan pada metode MOORA yang diterapkan dalam sebuah sistem yaitu tampak pada Tabel 2.8:

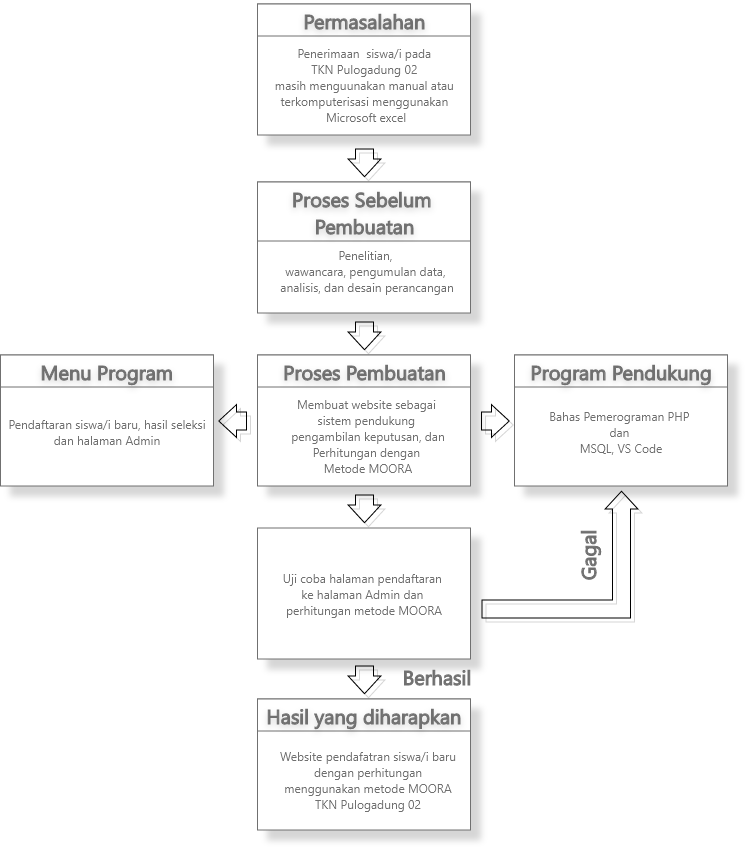
**Tabel 2.8.** Penelitian Tinjauan Studi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | Judul Penelitian | Tahun | Penulis | Metode | Tujuan Penelitian |
| 1. | Sistem Pendukung Keputusan Penerimaan Siswa Baru Menggunakan Metode Moora (*Multi-Objective Optimization On The Basis Of Ratio Analysis*) | 2021 | Andi Arisman, Fricles Ariwisanto Sianturi | *Multi-Objective Optimization On The Basis Of Ratio Analysis* (MOORA) | Menerapkan sebuah sistem pendukung keputusan untuk memudahkan Guru dalam penyeleksian penerimaan siswa baru yang mendaftar |
| 2. | Perbandingan Metode Moora Dan Topsis Dalam Penentuan Penerimaan Siswa Baru Dengan Pembobotan Roc Menggunakan Gui Matlab (Studi Kasus : Madrasah Aliyah Negeri Asahan Tahun Ajaran 2018/2019) | 2019 | Rafida Zahro Hasibuan, Alan Prahutama, Dwi Ispriyanti | *Multi-Objective Optimization on the basis of Ratio Analysis* (MOORA) dan *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS) | Menghasilkan sistem pengolahan data dengan uji analisis sensitivitas untuk mengetahui metode terbaik dan membantu pihak sekolah dalam melakukan penyeleksian calon siswa baru dengan menerapkan sistem penunjang keputusan |
| 3. | Sistem Pendukung Keputusan Penerimaan Santri Baru Untuk Menentukan Kelas Diniyah Menggunakan Metode Moora (*Multi Objective Optimization On The Basis Of Ratio Analysis*) | 2019 | Muhammad Faazada Roini | *Multi-Objective Optimization On The Basis Of Ratio Analysis* (MOORA) | Membuat sistem pendukung keputusan, agar proses penyeleksian santri baru akan lebh akurat serta dapat menggantikan sistem lama |
| 4. | Sistem Perangkingan Mahasiswa Baru Jalur Penerimaan Pmdk Dengan Metode Moora Dan *Analytic Hierarchy Process* | 2019 | Hilmi Keigi Naafi, Ragil Saputra | *Multi-Objective Optimization On The Basis Of Ratio Analysis* (MOORA) dan *Analytic Hierarchy Process* (AHP) | Mengkombinasika sistem penunjang keputusan dengan dua metode dengan hasil  otomatisasi dalam proses seleksi, dan perhitungan nilai ranking |
| 5. | Perancangan Sistem Pendukung Keputusan Penerimaan Siswa Baru Pada Smk Swa Bina Karya Medan Menggunakan Metode Moora | 2019 | Lubis, Jonatan Budi Erikson | *Multi-Objective Optimization On The Basis Of Ratio Analysis* (MOORA) | Merancang suatu sistem pendukung keputusan sesuai dengan kriteria, agar  mempercepat dan mempermudah pihak sekolah dalam proses pendataan dan proses seleksi siswa baru |

Setelah mempelajari dan mengkaji beberapa penelitian sebelumnya mengenai sistem penunjang keputusan dengan metode MOORA, peneliti membuat perancangan Sistem pendukung pengambilan keputusan berdasarkan penelitian sebelumnya, peneliti lebih memadukan antara pengambilan keputusan untuk seleksi penerimaan siswa/i baru dengan dengan hasil akhir nilai ranking, dimana nilai hasil ranking yang paling tinggi akan diterima masuk oleh instansi pendidikan.

* 1. **Kerangka Pikir Masalah**

Kerangka pikir merupakan pemaparan kerangka pikir atau pentahapan pemikiran secara kronologis dalam menjawab atau menyelesaikan suatu pokok permasalahan penelitian berdasarkan pemahaman teori dan konsep-konsep. Pemaparan ini dilakukan dalam bentuk bagan alir yang sederhana seperti pada gambar 2.19:

**Gambar 2.19** Kerangka Pikir Masalah

Kerangka pemikiran di atas dapat dijelaskan sebagai berikut:

* 1. Proses penyeleksian beasiswa untuk penerimaan siswa/i baru masih menggunakan sistem manual atau sudah terkomputerisasi menggunakan *microsoft excel.*
  2. Melakukan kegiatan penelitian sesuai metedologi yang akan digunakan.
  3. Untuk membuat *website* DSS digunakan bahasa pemrograman PHP dan MYSQL dengan *tool* pendukung VS Code.
  4. Berdasarkan permasalahan dan metode pengembangan maka dibuat DSS berbasis website disertai perhitungan (*proses*) data dengan perhitungan metode MOORA.
  5. Halaman menu program terdapat beberapa menu yang nantinya akan menghasilkan halaman untuk pendaftaran siswa/i, hasil seleksi dari perhitungan metode MOORA dan halaman admin akan menampilkan perhitungan-perhitungan metode MOORA.
  6. Hasil yang diharapkan yaitu berupa sistem pendukung pengambilan keputusan yang ditujukan untuk penerimaan siswa/i baru, dimana akan memudahkan panitia dalam mengambil keputusan dan mengurangi kesalahan-kesalahan yang ada dalam proses keputusan serta seleksi dengan hasil nilai perhitungan metode MOORA.