

BAB II

LANDASAN TEORI

2.1 Pemilihan Mahasiswa Berprestasi

Penggunaan istilah pemilihan Mahamasiswa Berprestasi (Pilmapres) dimulai tahun 2004 yang kemudian pada tahun 2017 dimulai dengan akromin Pilmapres dan sejak tahun itu dapat diselenggarakan setiap tahun dengan kualitas yang terus ditingkatkan. Pilmapres dinilai telah memberikan dampak positif pada budaya berprestasi dan menghargai prestasi serta karya mahamasiswa di kalangan perguruan tinggi dan secara langsung atau tidak langsung dapat mengangkat martabat mahamasiswa dan perguruan tingginya.

(http://pilmapres.ristekdikti.go.id/file/pendukung/2017/PEDOMAN_PILMAPRES_SARJANA_2017.pdf)

Menurut petunjuk pelaksanaan pemilihan mahamasiswa berprestasi tahun 2020 yang di publikasikan oleh Kementerian Pendidikan dan Kebudayaan. Dilaksanakannya pemilihan mahamasiswa berprestasi untuk menciptakan mahamasiswa unggul yang memiliki keseimbangan antara tiga kecerdasan yaitu intelektual, emosional dan spiritual.

a. Persyaratan Umum

- 1) Terdaftar pada PD-Dikti dan aktif sebagai mahamasiswa program Sarjana dan Diploma maksimal semester VI yang dibuktikan dengan Kartu Tanda Mahamasiswa (KTM) yang masih berlaku.

- 2) Berusia tidak lebih dari 22 tahun pada tanggal 1 Januari 2021 yang di buktikan dengan Kartu Tanda Penduduk (KTP) atau Kartu Izin Tinggal Terbatas (KITAS).
- 3) Belum pernah menjadi pemenang Pilmapres Tingkat Nasional.
- 4) Memiliki Indeks Prestasi Kumulatif (IPK) minimal 3,00.
- 5) Menunjukkan Surat Pengantar dari pimpinan perguruan tinggi bidang kemahasiswaan (Wakil Rektur/Ketua/Direktur) yang menyatakan bahwa yang bersangkutan merupakan wakil resmi mahasiswa berprestasi program sarjana/diploma dari perguruan tingginya. Setiap perguruan tinggi hanya diwakili oleh 1 (satu) orang peserta Program Sarjana dan/atau 1 (satu) orang peserta Program Diploma.

b. Persyaratan Khusus

Persyaratan yang perlu dipenuhi oleh peserta pemilihan mahasiswa berprestasi yang akan dinilai tim juri sesuai dengan prestasi yang dimiliki yaitu:

- 1) Prestasi yang diperoleh melalui kompetisi
 - a. Ilmiah/Penalaran/Akademik
 - b. Seni-Budaya
 - c. Olah Raga
 - d. Teknologi & Sains, serta Inovasi
 - e. Keagamaan
 - f. Kewirausahaan

2) Prestasi yang bukan diperoleh melalui kompetisi

a) Berperan sebagai :

1. Pemakalah/penyaji/narasumber/pemandu, moderator dalam forum ilmiah/seminar/FGD/konferensi/lokakarya/pelatihan.
2. Wirausahawan (entrepreneur)
3. Pelatih/wasit/juri/coach/adjudicator
4. Pemberdaya masyarakat
5. Ketua/kordinasi kepanitiaan dalam kegiatan tingkat provinsi/nasional/regional/internasional

b) Menghasilkan :

1. Temuan Inovatif
2. Karya yang telah mendapatkan HaKI
3. Literatur berupa buku, artiker, karya tulis, cerpen, novel, lagu/hasil seni yang di publikasikan/diterbitkan
4. Produk di bidang seni/olahraga/teknologi

c) Memperoleh :

1. HaKI
2. Anugerah

3) Capaian Karier di Organisasi Kepemudaan/Kemahamasiswaan

- a) Organisasi kemahamasiswaan/lembaga kemahamasiswaan : Badan Eksekutif Mahamasiswa, Senat Mahamasiswa, Dewan Perwakilan Mahamasiswa,

Majelis Permusyawaratan Mahasiswa, Himpunan Mahasiswa

- b) Unit Kegiatan Mahasiswa
- c) Badan Semi Otonom
- d) Organisasi Profesi Mahasiswa
- e) Organisasi Sosial Masyarakat

4) Penulisan gagasan kreatif

Yang dimaksud Gagasan kreatif adalah karya ilmiah (10 s.d 15 halaman) berisi uraian yang dilandasi penalaran logis dan data akurat dengan menyelesaikan masalah pembangunan yang berkaitan dengan Sustainable Development Goals (SDGs) dan Revolusi Industri 4.0 dan untuk program Sarjana dapat berupa konsep/strategi sedangkan untuk diploma harus berupa uraian produk yang dihasilkan.

- 5) Memiliki kemampuan bahasa Indonesia dan bahasa Inggris yang baik serta wawasan kebangsaan. (Kemampuan Bahasa Inggris akan dipakai untuk komunikasi dalam beberapa kegiatan dan diskusi)

c. Komponen Penilaian

- 1. Indeks Prestasi Kumulatif
- 2. Bahasa Inggris

Konten, Akurasi, kefasihan dan pengucapan, pemahaman dan respon, performa keseluruhan

- 3. Penulisan Gagasan Kreatif

Cara penulisan, kelengkapan isi, kualitas gagasan.

4. Penyajian Gagasan Kreatif

Penyajian dan tanya jawab.

5. Pencapaian Prestasi/Kemampuan yang diunggulkan

Nilai Pencapaian Prestasi/Kemampuan Unggul dan Nilai penghargaan/pengakuan.

6. Capaian Karier dalam berorganisasi

Nilai Kepemimpinan yang di jabat.

7. Kepribadian

a. Komunikasi

b. Kerjasama

c. Kemampuan berpikir kritis

d. Kreativitas

e. Inisiatif

f. Kepercayaan diri

g. Kepedulian pada kesulitan peserta lain

h. Kepemimpinan

d. Bobot Penilaian

1. Indeks Prestasi Kumulatif

2. Nilai Pencapaian Prestasi/Kemampuan Unggul

3. Nilai Penghargaan/Pengakuan

4. Nilai Kepemimpinan

5. Penilaian Penulisan Gagasan Kreatif

6. Rubrik Penilaian Penyajian Gagasan Kreatif melalui Media Poster

7. Rubrik Penilaian Berkomunikasi dalam Bahasa Inggris

2.2 Sistem Informasi

Sistem didefinisikan sebagai kumpulan prosedur yang saling berkaitan dan saling berhubungan untuk melakukan suatu tugas bersama-sama. Secara garis besar, sebuah sistem informasi terdiri atas tiga komponen tersebut mencakup software, hardware dan brainware. (Pratama,2014)

Dalam sistem informasi diperlukannya klasifikasi alur informasi, hal ini disebabkan keanekaragaman kebutuhan akan suatu informasi oleh pengguna informasi. Kriteria dari sistem informasi antara lain, fleksibel, efektif dan efisien.

2.3 Sistem Pendukung Keputusan

Konsep Sistem Pendukung Keputusan (SPK) pertama kali diungkapkan pada awal tahun 1970-an oleh Michael S. Scott Morton dengan istilah Management Decision System. Sistem tersebut adalah suatu sistem yang berbasis komputer yang ditunjukkan untuk membantu pengambil keputusan dengan memanfaatkan data dan model tertentu untuk memecahkan berbagai persoalan yang tidak terstruktur (Dewanto, 2015).

2.4 Metode Penelitian

Menurut Sugiyono (2018:2) metode penelitian pada dasarnya merupakan cara ilmiah untuk mendapatkan data dengan tujuan dan kegunaan tertentu. Cara ilmiah berarti kegiatan penelitian yang didasarkan pada ciri-ciri keilmuan, yaitu rasional, empiris, dan sistematis. Rasional berarti kegiatan penelitian yang dilakukan dengan cara masuk akal, sehingga terjangkau oleh penalaran manusia. Empiris berarti cara-cara yang dilakukan itu dapat diamati oleh indra manusia, sehingga orang lain dapat mengamati dan mengetahui

cara yang digunakan. Sistematis artinya proses yang digunakan dalam penelitian yang menggunakan langkah-langkah tertentu yang bersifat logis.

2.4.1 VlseKriterijumska Optimizacija I Kompromisno Rejenje (Vikor)

Metode VIKOR (*Vise Kriterijumska Optimizacija I Kompromisno Resenje*) didasarkan pada perankingan dan pemilihan dari sekumpulan alternatif yang ada. Metode ini dapat digunakan untuk meranking sebanyak kriteria baik itu kualitatif maupun kuantitatif. Metode vikor memiliki kelebihan dalam mengkompromi alternatif yang ada, serta dapat menyelesaikan pengambilan keputusan bersifat diskret pada kriteria yang bertentangan dan *non commensurable*, yaitu perbedaan unit antar kriteria.

2.4.1.1 Matriks Keputusan (F)

Dari data yang didapat dijadikan data untuk matriks keputusan F. Pada Langkah ini setiap kriteria dan alternative disusun ke dalam bentuk matriks F; A_j menyatakan alternatif ke $i=1,2,3,\dots,m$; dan C_{xn} menyatakan kriteria ke $j=1,2,3,\dots, n$.

$$F = \begin{matrix} & \begin{matrix} C_{x1} & C_{x2} & \dots & C_{xn} \end{matrix} \\ \begin{matrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{matrix} & \begin{bmatrix} a_{11} & x_{12} & \dots & x_{1n} \\ a_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix} \end{matrix}$$

Keterangan

- x_{ij} : Respon alternatif i pada kriteria j
- i : 1,2,3, ..., m adalah nomor urutan alternatif
- j : 1,2,3, ..., n adalah nomor urutan atribut atau kriteria

- d. A_i : Alternatif ke -i
- e. C_j : Kriteria ke -j
- f. F : Matriks Keputusan

2.4.1.2 Bobot Kriteria (W)

Menentukan bobot kriteria yang diperoleh dari pengguna system sesuai dengan kebutuhan atau kriteria yang diinginkan. Rumusan umum untuk bobot kriteria adalah berlaku persamaan.

$$\sum_{j=1}^n w_j = 1$$

Keterangan

- a. W_j : Bobot kriteria j
- b. j : 1,2,3, ..., n adalah nomor urutan atribut atau kriteria

2.4.1.3 Matriks Normalisasi (N)

Membuat matriks normalisasi dengan menentukan nilai positif dan nilai negative sebagai solusi ideal dari setiap kriteria. Matrik F tersebut kemudian di normalisasikan dengan persamaan berikut :

$$N_{ij} = \frac{(f^+ - f_{ij})}{(f_j^+ - f_j^-)}$$

Keterangan

- a. f_{ij} : Fungsi respon alternative i pada kriteria j
- b. f^+ : nilai terbaik/positif dalam satu kriteria j

- c. f_j^- : nilai terjelek/negatif dalam satu kriteria j
- d. $i : 1,2,3, \dots, m$ adalah nomor urutan alternatif
- e. $j : 1,2,3, \dots, n$ adalah nomor urutan atribut atau kriteria
- f. N : Matriks ternormalisasi

Penentuan nilai data terbaik/positif (f_j^+) dan terburuk/negatif (f_j^-) atau dengan istilah cost dan benefit dalam satu variable penelitian higher-the-better (HB) atau lower-the-better (LB) (Kusdiantoro 2012).

Nilai (f_j^+) dan (f_j^-) tersebut dinyatakan sebagai berikut.

$$f_j^+ = \max(f_{1j}, f_{2j}, f_{3j}, \dots, f_{mj})$$

$$f_j^- = \min(f_{1j}, f_{2j}, f_{3j}, \dots, f_{mj})$$

Keterangan

- a. f_j^+ : nilai terbaik/positif dalam satu kriteria j
- b. f_j^- : nilai terjelek/negatif dalam satu kriteria j
- c. $i : 1,2,3, \dots, m$ adalah nomor urutan alternatif
- d. $j : 1,2,3, \dots, n$ adalah nomor urutan atribut atau kriteria.

2.4.1.4 Normalisasi Bobot (F^*)

Menentukan nilai terbobot dari data ternormalisasi untuk setiap alternatif dan kriteria. Melakukan perkalian antara nilai data yang telah dinormalisasi (N) dengan nilai bobot kriteria (W) yang telah ditentukan dengan perhitungan sebagai berikut :

$$F_{ij}^* = w_j \cdot N_{ij}$$

Keterangan

- a. F_{ij}^* : Nilai data ternormalisasi yang sudah terbobot untuk alternative i pada kriteria j
- b. W_j : Nilai bobot pada kriteria j
- c. N_{ij} : Nilai data ternormalisasi untuk alternatif I pada kriteria j
- d. i : 1,2,3, ..., m adalah nomor urutan alternative
- e. j : 1,2,3, ..., n adalah nomor urutan atribut atau kriteria

2.4.1.5 Utility Measure (S) dan Regret Measures (R)

Utility measures (S) dan Regret measures (R) dari setiap alternative dihitung menggunakan rumus sebagai berikut :

$$S_i = \sum_{j=1}^n w_j \frac{(f_j^+ - f_{ij})}{(f_j^+ - f_j^-)}$$

S_i merupakan jarak Manhattan (Manhattan distance) yang terbobot dan dinormalisasi.

$$R_i = \max_j \left[w_j \frac{(f_j^+ - f_{ij})}{(f_j^+ - f_j^-)} \right]$$

R_i merupakan jarak Chebyshev (Chebyshev distance) yang terbobot dan dinormalisasi.

S_i (maximum grup utility) dan R_i (minimum individual regret of the opponent), keduanya menyatakan utility measures yang diukur dari titik terjauh dan titik terdekat dari solusi ideal,

sedangkan w_j adalah bobot yang diberikan pada setiap kriteria ke- j .

2.4.1.6 Indeks Vikor (Q)

Setiap alternative i dihitung indeks vikornya menggunakan rumus sebagai berikut :

$$Q_i = v \left[\frac{S_i - S^-}{S^+ - S^-} \right] + (1 - v) \left[\frac{R_i - R^-}{R^+ - R^-} \right]$$

Dimana $S^- = \min(S_i)$, $S^+ = \max(S_i)$, $R^- = \min(R_i)$, $R^+ = \max(R_i)$ dan v merupakan bobot berkisar antara 0-1 (umumnya bernilai 0,5). Nilai v adalah nilai bobot strategy of the maximum group utility, sedangkan $1-v$ adalah bobot dari individual regret semakin kecil nilai indeks Vikor (Q_i) maka semakin baik pula solusi alternative tersebut.

2.4.1.7 Perangkingan Alternatif

Setelah Q_i dihitung, maka akan terdapat 3 macam perangkingan yaitu S_i , R_i , Q_i . Solusi kompromi dapat dilihat pada perangkingan Q_i .

Pengurutan perangkingan ditentukan dari nilai yang paling rendah dengan solusi kompromi sebagai ideal dilihat dari perangkingan Q_i dengan nilai terendah. Karena nilai S_i merupakan solusi yang diukur dari titik terjauh solusi ideal, sedangkan nilai R_i merupakan solusi yang diukur dari titik terdekat solusi ideal.

Solusi Kompromi ditentukan dari alternative yang memiliki peringkat terbaik dengan mengukur indeks Vikor yang minimum, apabila 2 kondisi berikut terpenuhi.

A. Acceptable Advantage

$$Q_{(A_2)} - Q_{(A_1)} \geq DQ$$

$$DQ = \frac{1}{(m-1)}$$

Dimana m adalah banyaknya alternative, alternative A1 adalah peringkat pertama dan A2 adalah peringkat kedua dari perangkian Qi

B. Acceptable Stability in Decision Making

Alternatif A1 juga harus menjadi peringkat terbaik dalam perangkian. Solusi kompromi ini stabil dalam proses pengambilan keputusan, yang dapat menjad voting by majority rule (saat $v > 0,5$) atau by concensus ($v = 0,5$) atau with veto ($v < 0,5$)

C. Solusi Kompromi

Jika salah satu kondisi tidak memuaskan, maka solusi kompromi dapat diajukan sebagai berikut :

1. Memilih alternative A1 dan A2 , jika hanya kondisi 2 tidak memuaskan, atau
2. Memilih alternative A1, A2, ..., Am jika kondisi 1 tidak memuaskan.

Am merupakan alternative yang ditentukan dengan menggunakan persamaan

$$Q_{(A_m)} - Q_{(A_i)} < DQ$$

Dimana m maksimum adalah alternatif yang posisinya berapa pada kondisi saling berdekatan.

2.5 Alat Bantu Pemodelan Sistem

2.5.1 *Unified Modelling Language (UML)*

Menurut Windu Gata, Grace dalam jurnal Ade Handini (2016:108). UML (*Unified Modeling Language*) adalah Bahasa spesifikasi standar yang digunakan untuk mendokumentasikan, menspesifikasika dan membangun perangkat lunak.

UML menawarkan sebuah standar untuk merancang model sebuah system. Ada beberapa diagram dalam UML (*Unified Modelling Language*) antara lain:

1) *Use Case Diagram*

Ini menggabarkan fungsionalitas yang diharapkan dari sebuah system. Use case merepresentasikan sebuah interaksi antara actor dengan system. Adapun symbol dari use case diagram antara lain :

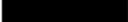
Tabel 2. 1 Simbol *Usecase* Diagram

No	Simbol	Keterangan
1	Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
2	Aktor / <i>Actor</i>  Nama Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3	Asosiasi / <i>Association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4	Ekstensi / <i>Extend</i> <<extend>> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.
5	Generalisasi / <i>Generalization</i> 	Hubungan generalisasi dan spesialisasi (umum- khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6	Menggunakan / <i>Include / uses</i> <<include>> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>usecase</i> tambahan dijalankan.

2) Activity Diagram

Menggambarkan berbagai alur aktifitas dalam system yang sedang dirancang, bagaimana masing-masing alur berawal dan decision yang mungkin terjadi dan bagaimana alur berakhir. Adapun symbol dari Activity Diagram antara lain:

Tabel 2. 2 Simbol Activity Diagram

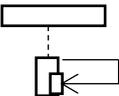
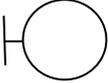
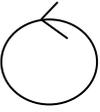
No	Gambar	Keterangan
1.	 <i>Activity</i>	Aktivitas yang dilakukan sistem, aktifitas biasanya diawali dengan kata kerja
2.	 <i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
3.	 Status Awal	Status awal aktiviatas sistem, sebuah diagram aktivitas memiliki status awal.
4.	 Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memilki sebuah status akhir.
5.	 <i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

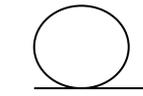
3) Sequence Diagram

Sequence Diagram menggambarkan interaksi antar objek didalam dan sekitar sister (termasu pengguna dan display) berupa message yang digambarkan terhadap waktu, *Sequence* Diagram terdiri atas dimensi vertical (waktu) dan dimensi horizontal (objek-objek yang terkait). Setiap pesan yang dikirimkan bisa memberi respon (return)

relative pada scenario yang dirancang di Use Case Diagram. Interaksi yang terjadi bisa bersifat instansiasi sebuah objek maupun static metod dari sebuah class. Berikut ini adalah symbol-symbol dari *sequence diagram* :

Tabel 2. 3 Simbol *Sequence Diagram*

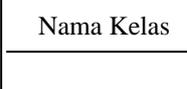
No	Gambar	Keterangan
1.	 Aktor	Aktor adalah pengguna sistem, pengguna dapat berarti manusia, mesin atau sistem lain atau subsistem dari model apapun yang berinteraksi dengan sistem dari <i>boundary</i> sistem
2.	 <i>Lifeline</i>	Peserta individu dalam interaksi (yaitu jalur hidup yang tidak dapat memiliki multiplisitas)
3.	 <i>Message</i>	Menunjukkan aliran informasi atau kendali transaksi antar elemen
4.	 <i>Sel-Message</i>	Mencerminkan proses baru atau metode pemanggilan operasi <i>lifeline</i> . Ini adalah spesifikasi pesan biasanya dalam <i>sequence diagram</i>
5.	 <i>Boundary</i>	<i>Boundary</i> adalah objek stereotip yang memodelkan batasan sistem. Biasanya layar <i>user interface</i>
6.	 <i>Control</i>	<i>Control</i> adalah objek stereotip yang mengontrol atau mengatur entitas

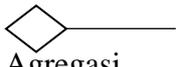
7.	 <i>Entity</i>	Merupakan table pada <i>database</i> yang merupakan model penyimpanan data yang menangkap dan menyimpan informasi dalam sistem
----	--	--

4) *Class Diagram*

Class ini adalah sebuah spesifikasi yang jika diinstansi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek serta menggambarkan keadaan suatumsistem sekaligus menawarkan layanan untuk memanipulasi keadaan.

Tabel 2. 4 Simbol *Class Diagram*

No	Gambar	Keterangan
1.	 Nama Kelas	Kelas pada stuktur sistem
2.		Properti dari suatu kelas yang melambangkan batas nilai
3.	 <i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4.	 <i>Directed Association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain. Biasanya disertai <i>multiplicity</i>
5.	 <i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas
6.	 Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)

7.	 Agregasi	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)
----	---	--

2.5.2 SDLC

Menurut Rosa dan Shalahudin (2015:26), SDLC atau sering disebut system development life cycle adalah proses mengembangkan atau mengubah suatu system Perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan system perangkat lunak sebelumnya.

Menurut Pressman (2015) didalam SDLC terdapat beberapa odel diantaranya adalah model waterfall, terkadang disebut sebagai siklus hidup klasik, menunjukkan sistematis, pendekatan sekuensial untuk penyebaran perangkat lunak yang dimulai dengan spesifikasi permintaan pelanggan dan berlangsung melalui perencanaan, pemodelan, contruction dan deployment yang berakhir pada dukungan yang berkelanjutan dari tersesainya software.

Fase-fase dalam model waterfall menurut referensi pressman seperti terlihat pada gambar

1. *System Engineering*

Langkah ini merupakan analisis terhadap kebutuhan software, dan tahap untuk mengadakan pengumpulan data dengan melakukan pertemuan dengan costumer, maupun mengumpulkan data tambahan baik yang ada di jurnal, artikel, maupun dari internet.

2. *Analysis*

Proses *planning* merupakan lanjutan dari proses *communication* (*analysis requirement*). Tahapan ini akan menghasilkan dokumen *user requirement* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan user dalam pembuatan software, termasuk rencana yang akan dilakukan.

3. *Design*

Proses modeling ini akan menerjemahkan syarat kebutuhan ke sebuah perancangan software yang dapat diperkirakan sebelum dibuat coding. Proses ini berfokus pada rancangan struktur data, arsitektur software, representasi interface, dan detail (*algoritma*) *procedural*. Tahapan ini akan menghasilkan dokumen yang disebut *software requirement*.

4. *Coding*

Construction merupakan proses membuat kode. *Coding* atau pengkodean merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh computer. Programmer akan menerjemahkan transaksi yang diminta oleh user. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan software, artinya penggunaan computer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan testing terhadap system yang telah dibuat tadi. Tujuan testing adalah menemukan kesalahan-kesalahan terhadap system tersebut untuk kemudian diperbaiki.

5. *Testing*

Setelah perangkat lunak telah dianggap sesuai dengan requirement customer, perangkat lunak diuji untuk mempertahankan hasil system dan sesuai dengan perancangan system.

2.6 Alat Bantu Pembuatan Program

2.6.1 HTML

Hypertext Markup Language (HTML) adalah bahasa yang digunakan untuk menulis halaman *web*. *HTML* merupakan pengembangan dari standar pemformatan dokumen teks yaitu *Standard Generalized Markup Language (SGML)* (Kadir, 2014). *HTML* sebenarnya adalah dokumen *ASCII* atau teks biasa, yang dirancang untuk tidak tergantung pada suatu sistem operasi tertentu.

Mendesain *HTML* berarti melakukan suatu tindakan pemrograman. Namun *HTML* bukanlah sebuah bahasa pemrograman. Namun *HTML* hanyalah berisi perintah-perintah yang telah terstruktur berupa tag-tag penyusun. Menuliskan tag-tag *HTML* tidaklah sebatas hanya memasukkan perintah-perintah tertentu agar *HTML* kita dapat diakses oleh browser. Mendesain *HTML* adalah sebuah seni tersendiri. *Homepage* yang merupakan implementasi dari *HTML* adalah refleksi dari orang yang membuatnya. Untuk itu kita perlu mendesainnya dengan baik agar para pengunjung homepage yang kita buat merasa senang dan bermanfaat. Mendesain *HTML* dapat dilakukan dengan cara menggunakan *HTML Editor*, seperti notepad++, adobe dreamweaver dan lain-lain.

2.6.2 CSS

Cascading Style Sheets (CSS) adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa markup (Kadir, 2014). Penggunaan yang paling umum dari CSS adalah untuk memformat halaman web yang ditulis dengan *HTML* dan *XHTML*. Walaupun demikian, bahasanya sendiri dapat dipergunakan untuk semua jenis dokumen *XML* termasuk *SVG* dan *XUL*. Spesifikasi CSS diatur oleh *World Wide Web Consortium (W3C)*. CSS digunakan oleh penulis maupun pembaca halaman web untuk menentukan warna, jenis huruf, tata letak, dan berbagai aspek tampilan dokumen. CSS digunakan terutama untuk memisahkan antara isi dokumen (yang ditulis dengan *HTML* atau *bahasa markup* lainnya) dengan presentasi dokumen (yang ditulis dengan *CSS*). Pemisahan ini dapat meningkatkan aksesibilitas isi, memberikan lebih banyak keleluasaan dan kontrol terhadap tampilan, dan mengurangi kompleksitas serta pengulangan pada struktur isi.

2.6.3 PHP

Menurut Sidik (2014:4) PHP merupakan secara umum dikenal sebagai bahasa pemrograman *script-script* yang membuat dokumen HTML secara *on the fly* yang dieksekusi di *server web*, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat menggunakan editor teks atau editor HTML, dikenal juga sebagai bahasa pemrograman *server side*.

Madcoms (2016:148), *PHP* adalah bahasa *script* yang dapat ditanamkan atau disisipkan ke dalam HTML. *PHP* adalah bahasa

pemrograman script *server-side* yang didesain untuk pengembangan *web*.

Menurut Abdulloh (2016:3) “*PHP* singkatan dari *hypertext preprocessor* yang merupakan *server-side* programming, yaitu bahasa pemrograman yang diproses disisi *server*.”

Dari penjelasan di atas dapat disimpulkan bahwa *PHP* adalah bahasa pemrograman yang digunakan untuk membuat dan mendesain pengembangan dari sebuah server web.

2.6.4 MySQL

Menurut Nugroho (2014:26), “*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.

Menurut Buana (2014:2), “*MYSQL* Merupakan *database server* yang paling sering digunakan dalam pemograman *PHP*. *MYSQL* digunakan untuk menyimpan data dalam *database* dan memanipulasi data-data yang diperlukan. Manipulasi data tersebut berupa menambah, mengubah, dan menghapus data yang berada dalam *database*”.

Menurut Susanto (2014:73) “*Mysql* merupakan sistem manajemen database yang bersifat *relational*. Artinya, data yang dikelola dalam database akan diletakan pada beberapa *table* yang terpisah sehingga manipulasi data akan menjadi lebih cepat”.

2.6.5 XAMPP

Menurut Nugroho (2015:2), XAMPP adalah paket program *web* lengkap yang dapat Anda pakai untuk belajar pemrograman *web*, khususnya *PHP* dan *MYSQL*.

Menurut Susanto (2014:72) “XAMPP merupakan singkatan dari X (empat sistem operasi apapun), *apache*, *Mysql*, *PHP*, dan *Perl*. XAMPP adalah tool yang menyediakan paket perangkat lunak dalam satu buah paket”.

Menurut Buana (2014:4), XAMPP adalah perangkat lunak *open source* yang diunggah secara gratis dan bisa dijalankan di semua semua operasi seperti *windows*, *linux*, *solaris*, dan *mac*.

XAMPP berperan sebagai *server web* pada komputer anda. XAMPP juga dapat disebut sebuah *C-Panel server virtual*, yang dapat membantu anda melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus *online* atau terakses dengan *internet*. Bagian penting XAMPP yang biasa digunakan pada umumnya.

- a. XAMPP *Control Panel Application* berfungsi mengelola layanan (*service*) XAMPP. Seperti mengaktifkan layanan (*start*) dan menghentikan (*stop*) layanan dan juga mengontrol atau mengendalikan XAMPP dengan lebih efektif, mulai dari mengatur *setting website, database*.
- b. *htdocs* adalah folder tempat meletakkan berkas-berkas yang akan dijalankan yang sudah dibuat dan nantinya akan ditampilkan.

c. *PHPMAdmin* merupakan bagian untuk menyimpan dan mengelola *database* yang mendukung untuk keperluan tampilan sebuah *website*.

2.7 Basis Data

Menurut Connolly dan Begg (2010, p65), basis data adalah sekumpulan data yang berhubungan secara logikal dan deskripsi mengenai data itu sendiri yang dirancang untuk memenuhi kebutuhan informasi organisasi. Basis data adalah kumpulan terintegrasi dari elemen data yang secara logika saling berhubungan. Selain itu, Basis data adalah kumpulan file yang saling terkait.

