BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian mengenai pemilihan jurusan SMK bukanlah baru pertama kali ini dilakukan ,sudah ada penelitian terdahulu tentang penerapan metode *AHP* dalam penelitian tersebut. Penelitian terdahulu yang relevan dengan penelitian ini adalah sebagai berikut:

Tabel 2.1 Perbedaan Penilitian

).	Peneliti	Judul	Masalah	Hasil
1.	Wijayanto, S. (2019).	Sistem Pendukung Keputusan Penerimaan Peserta Didik Baru dengan Metode Analytical Hierarchy Process dan Simple Additive Weighting	1. Bagaimana meml pendukung penerimaan pes baru menggunak Analytical Process (AHP) b kondisi yang ad Putra Bangsa?	1. Perancangan dan pembuatan Sistem Pendukung Keputusan Penerimaan Peserta Didik Baru menggunakan metode pengambilan keputusan AHP (Analytical Hierarchy Process). Sedangkan metode pengembangan system menggunakan metode waterfall, yang terdiri dari beberapa tahap, yaitu analisis kebutuhan, perancangan sistem,

				desain sistem, coding dan testing, serta penerapan dan pengujian program. Sistem Pendukung Keputusan Penerimaan Peserta Didik Baru ini merupakan inovasi dari sistem penerimaan siswa baru di SMK Putra Bangsa.
	Julianti (2017)	Pemilihan Program Keahlian Menggunaka nMetode AHPPada SMK Citra Bangsa Bogor	2. Siswa di SMK Citra Bangsa Bogor diwajibkan memilih salah satu dari keempat program keahlian.	Berdasarkan hasil penelitian dapat disimpulkan bahwa sesuai dengan hasil perhitungan matematis metode AHP : 1. Sistem merekomendasi kan 33% siswa untuk memlih jurusan Akuntansi 2. Sistem merekomendasi kan 26% siswa untuk memlih jurusan Administrasi Perkantoran 3. Sistem merekomendasi kan 21% siswa untuk memlih jurusan pemasaran

				4. Sistem merekomendasikan 20% siswa untuk memlih jurusan teknik komputer.
	Ninik Tri Hartanti (2017)	Sistem Pendukung Keputusan Untuk Menentukan Program Keahlian Di SMK Syubbanul Wathon Magelang	3. Proses penentuan program keahlian pada SMK Syubbanul Wathon masih menggunakan cara manual.	1. Aplikasi berhasil mengimplementasikan algoritma K-means dan AHP utuk menentukan prioritas penentuan program keahlian SMK berdasarkan kriteria nilai Raport SMP, nilai UN SMP, nilai Placement Test dan Minat siswa di angket.
	Permana, Silvester Dian Handy (2017).	Sistem Penunjang Keputusan Pemilihan Sekolah Menengah Kejuruan Teknik Komputer	4. Banyaknya pilihan sekolah yang dapat membingungkan calon siswa dalam memilih sekolah.	Setelah dilakukan pengeolaan data mengguanakan metode MCDM maka diperoleh kesimpulan bahwa urutan prioritas dari paling tertinggi sampai yang rendah adalah:

		Dan Jaringan Yang Terfavorit Dengan Menggunakan Multi-Criteria Decision Making		1. SMK TI 2 dengan nilai bobot 44,8%. 2. SMK TI 1 dengan nilai bobot 29,3% 3. SMK TI 3 dengan nilai bobot 25,9%
	Lorang, S (2019)	Implementasi Metode AHP Dalam Pemilihan Jurusan SMK Studi Kasus : SMK Putra Tama	5. Proses seleksi pemilihan calon siswa di SMK Putra Tama masih secara manual,mulai dari registrasi pendataan data diri calon siswa hingga proses penentuan pemilihan calon siswa ke jurusan yang sesuai dengan kemampuan calon siswa.	1. Sistem yang dibangun mampu memberikan rekomendasi kepada siswa berupa terpilih di jurusan berdasarkan dari bobot kriteria penilaian berdasarkan metode AHP

2.1.1. Hasil Penelitian Tinjauan Pustaka

1. Perancangan dan pembuatan Sistem Pendukung Keputusan Penerimaan Peserta Didik Baru menggunakan metode pengambilan keputusan AHP (*Analytical Hierarchy Process*). Sedangkan metode pengembangan system menggunakan metode waterfall, yang terdiri dari beberapa tahap, yaitu analisis kebutuhan, perancangan sistem, desain sistem, coding dan testing, serta penerapan dan pengujian program. Sistem Pendukung Keputusan Penerimaan Peserta Didik Baru ini merupakan inovasi dari sistem penerimaan siswa baru di SMK Putra Bangsa.
2. Berdasarkan hasil penelitian dapat disimpulkan bahwa sesuai dengan hasil perhitungan matematis metode AHP :
    a) Sistem merekomendasi kan 33% siswa untuk memlih jurusan Akuntansi.
    b) Sistem merekomendasi kan 26% siswa untuk memlih jurusan Administrasi Perkantoran.
    c) Sistem merekomendasi kan 21% siswa untuk memlih jurusan pemasaran.
    d) Sistem merekomendasi kan 20% siswa untuk memlih jurusan teknik komputer.
3. Aplikasi berhasil mengimplementasik an algoritma K- means dan AHP utuk menentukan prioritas penentuan program keahlian SMK berdasarkan kriteria nilai Raport SMP, nilai UN SMP, nilai Placement Test dan Minat siswa di angket.
4. Setelah dilakukan pengeolaan data mengguanakan metode MCDM maka

diperoleh kesimpulan bahwa urutan prioritas dari paling tertinggi sampai yang rendah adalah:

1. SMK TI 2 dengan nilai bobot 44,8%.
2. SMK TI 1 dengan nilai bobot 29,3%
3. SMK TI 3 dengan nilai bobot 25,9%

5. Sistem yang dibangun mampu memberikan rekomendasi kepada siswa berupa terpilih di jurusan berdasarkan dari bobot kriteria penilaian berdasarkan metode AHP

Dari semua penelitian terdahulu maka penulis membuat perbedaan antara penelitian terdahulu dan penelitian yang akan di buat adalah sebagai berikut :

Sistem ini bertujuan untuk mempermudah pihak sekolah untuk mendukung dalam pengambilan keputusan pada penerimaan siswa baru yang mendaftar dan melakukan tes yang telah dilakukan oleh para calon siswa dan siswi baru.

2.2 Landasan Teori

2.2.1 Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (SPK) adalah 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 dan model. Kata berbasis komputer merupakan kata kunci, karena hampir tidak mungkin membangun SPK tanpa memanfaatkan komputer sebagai alat bantu, terutama untuk menyimpan data serta mengelola model (Sari, 2017).

Pada dasarnya Sistem Pendukung Keputusan atau dikenal juga dengan istilah *Decision Support System* (DSS) ini merupakan pengembangan lebih lanjut dari

sistem informasi manajemen terkomputerisasi yang dirancang sedemikian rupa sehingga bersifat interaktif dengan pemakainya. Sifat interaktif ini dimaksudkan untuk memudahkan integrasi antara berbagai komponen dalam proses pengambilan keputusan seperti prosedur, kebijakan, teknik analisis, serta pengalaman dan wawasan manajerial guna membentuk suatu kerangka keputusan yang bersifat fleksibel.

Keterbatasan Sistem Pendukung Keputusan

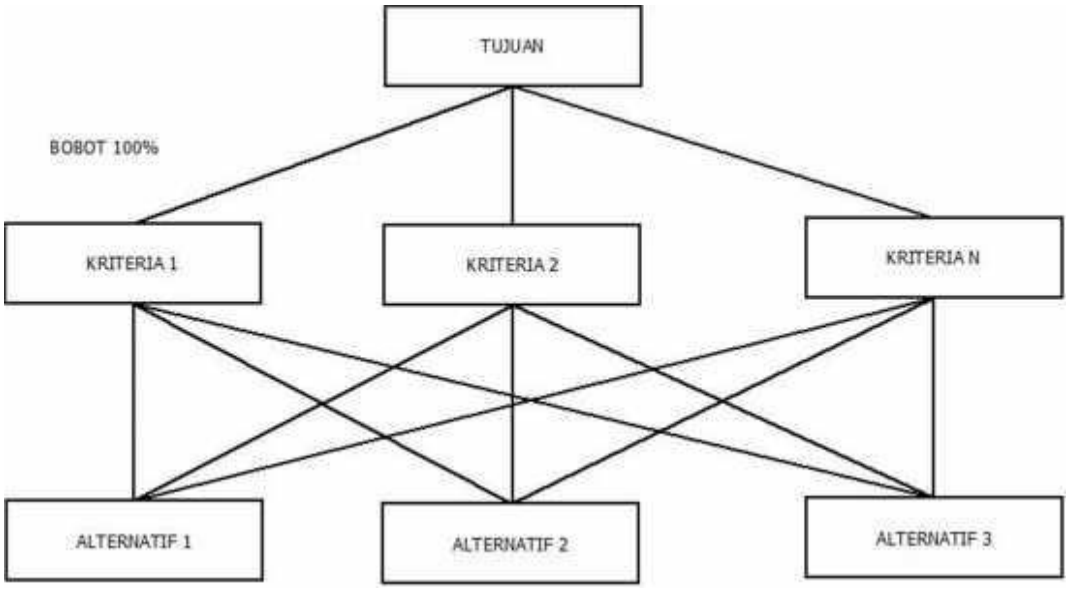
1. Ada beberapa kemampuan manajemen dan bakat manusia yang tidak dapat dimodelkan, sehingga model yang ada dalam sistem tidak semuanya mencerminkan persoalan sebenarnya.
2. Kemampuan suatu SPK terbatas pada pembendaharaan pengetahuan yang dimilikinya (pengetahuan dasar serta model dasar).
3. Proses-proses yang dapat dilakukan oleh SPK biasanya tergantung juga pada kemampuan perangkat lunak yang digunakannya.
4. SPK tidak memiliki kemampuan intuisi seperti yang dimiliki oleh manusia. Karena walau bagaimana pun canggihnya suatu SPK hanyalah suatu kumpulan perangkat keras, perangkat lunak dan sistem operasi yang tidak dilengkapi dengan kemampuan berpikir.

2.2.2 *Analytical Hierarchy Proses* (AHP)

Analytical Hierarchy Process merupakan suatu metode untuk memecahkan suatu masalah yang kompleks dan tidak terstruktur ke dalam suatu kelompok-kelompoknya, mengatur kelompok tersebut ke dalam suatu *hierarchy*, memasukkan nilai numerik sebagai pengganti persepsi manusia dalam melakukan perbandingan relatif dan akhirnya dengan suatu sintesa ditentukan elemen mana

yang mempunyai prioritas tinggi. AHP merupakan metode yang memperhatikan faktor- faktor subyektifitas seperti persepsi, preferensi, pengalaman dan intuisi. AHP adalah Sebuah hierarki fungsional dengan input utamanya persepsi manusia, keberadaan hirarki memungkinkan dipecahnya masalah kompleks atau tidak terstruktur dalam sub-sub masalah, lalu menyusunnya menjadi suatu bentuk hierarki (Kusrini, 2007). Tetapi perlu diingat bahwa sistem pendukung keputusan hanya untuk memberikan alternatif pilihan bukan untuk menentukan keputusan akhir.

Struktur sebuah model AHP adalah model dari sebuah pohon terbaik. Ada suatu tujuan tunggal di puncak pohon yang mewakili tujuan dari masalah pengambilan keputusan. Seratus persen bobot keputusan adalah di titik ini. Tepat dibawah tujuan adalah titik daun yang menunjukkkan kriteria, baik kualitatif maupun kuantitatif. Bobot Tujuan harus dibagi diantara titik-titik kriteria berdasarkan rating. Bobot dari tiap-tiap kriteria adalah 100 % dibagi dengan bobot titik-titik kriteria berdasarkan rating. Setiap alternatif dibandingkan dengan masing-masing kriteria.



Gambar 2.2 Hirarki AHP

Secara khusus, AHP sesuai untuk digunakan dalam pengambilan keputusan yang melibatkann perbandingan elemen keputusan yang sulit untuk dinilai secara kuantitatif. Hal ini berdasarkan asumsi bahwa reaksi natural manusia ketika menghadapi pengambilan keputusan yang kompleks adalah mengelompokkan elemen-elemen keputusan tersebut menurut karakteristiknya secara umum. Pengelompokan ini meliputi pembuatan hirarki(ranking) dari elemen-elemen keputusan kemudian melakukan perbandingan antara setiap pasangan dalam setiap kelompok, sebagai suatu matriks. Setelah itu akan diperoleh bobot dan rasio inkonsistensi untuk setiap elemen. Dengan demikian akan mudah untuk mengujikonsistensi data (Saaty, 2001).

Tabel 2.2. Skala Penilaian Perbandingan Pasangan Saaty (2001)

Intensitas Kepentingan	Keterangan
1	Kedua elemen sama pentingnya
3	Elemen yang satu sedikit lebih penting dari pada elemen yang lainnya

5	Elemen yang satu lebih penting dari pada elemen lainnya
7	Satu elemen jelas lebih mutlak penting dari pada elemen lainnya
9	Satu elemen mutlak penting dari pada elemen Lainnya
2,4,6,8	Nilai-nilai antara dua nilai pertimbangan yang Berdekatan
Kebalikan	Jika untuk aktivitas i mendapat satu angka dibanding dengan aktivitas j, maka j mempunyai nilai kebalikannya dibanding dengan i.

Pada keadaan nyata sering terjadi penyimpangan dari hubungan tersebut sehingga matriks menjadi tidak konsisten. Penyimpangan konsistensi dinyatakan dengan *Consistency Index* (CI) dengan persamaan (2.1).

$$CI = \frac{\lambda \max - n}{n-1}$$

λmax = eigen value maksimum

n = ukuran matriks

Kebalikan dari CI adalah *Indeks Random*(RI) *Indeks Random* (RI) merupakan nilai acak CI untuk suatu n. Nilai *Indeks Random* (RI) dapat dilihat pada tabel 2.3 Tabel 2.3 *Indeks Random* (RI)

N	1	2	3	4	5	6	7	8	9	10
RI	0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

Perbandingan antara CI dan RI suatu matriks didefinisikan sebagai :

ConsistencyRatio(CR) (2.2).

$$CR = \frac{CI}{RI}$$

Matriks perbandingan berpasangan untuk model AHP dapat diterima jika besarnya CR = 0.1.

Secara detil, terdapat tiga prinsip dasar AHP, yaitu (Saaty, 1994):

2.4 Dekomposisi (*Decomposition*)

Setelah persoalan didefinisikan, maka perlu dilakukan *decomposition*, yaitu memecah persoalan yang utuh menjadi unsur-unsurnya. Jika ingin mendapatkan hasil yang akurat, maka pemecahan terhadap unsur-unsurnya dilakukan hingga tidak memungkinkan dilakukan pemecahan lebih lanjut. Pemecahan tersebut akan menghasilkan beberapa tingkatan dari suatu persoalan.

3.4 Penilaian Komparasi (*Comparative Judgment*)

Prinsip ini membuat penilaian tentang kepentingan relatif dua elemen pada suatu tingkat tertentu yang berkaitan dengan tingkat di atasnya. Penilaian ini merupakan inti dari AHP karena berpengaruh terhadap prioritas elemen- elemen. Hasil penilaian ini tampak lebih baik bila disajikan dalam bentuk matriks perbandingan berpasangan (*pairwise comparison*).

4.4 Penentuan Prioritas (*Synthesis of Priority*)

Dari setiap matriks *pairwise comparison* dapat ditentukan nilai *eigenvector* untuk mendapatkan prioritas daerah (*local priority*). Oleh karena matriks *pairwise comparison* terdapat pada setiap tingkat, maka *global priority* dapat diperoleh dengan melakukan sintesa di antara prioritas daerah. Prosedur melakukan sintesa

berbeda menurut hierarki. Pengurutan elemen- elemen menurut kepentingan relatif melalui prosedur sintesa dinamakan *priority setting*.

Langkah-langkah metode AHP adalah:

1. Mendefinisikan masalah dan menentukan solusi yang diinginkan.
2. Membuat struktur hirarki yang diawali dengan tujuan umum, dilanjutkan dengan subtujuan - subtujuan, kriteria dan kemungkinan alternatif pada tingkatan kriteria yang paling bawah.
3. Membuat matriks perbandingan berpasangan yang menggambarkan kontribusi relatif atau pengaruh setiap elemen terhadap masing-masing tujuan kriteria yang setingkat di atasnya. Perbandingan bersadarkan pertimbangan dari pendukung keputusan dengan menilai tingkat kepentingan suatu elemen dibandingkan elemen lainnya.
4. Melakukan perbandingan berpasangan sehingga diperoleh *judgment* seluruhnya sebanyak n x [(n-1)/2] buah, dengan n adalah banyaknya elemen yang dibandingkan.
5. Menghitung nilai *eigen* dan menguji konsistensinya, jika tidak konsisten maka pengambilan data diulangi.
6. Mengulangi langkah 3, 4 dan 5 untuk seluruh tingkat hirarki.
7. Menghitung vektor eigen dari setiap matriks perbandingan berpasangan.

   Nilai vektor eigen merupakan bobot setiap elemen. Langkah ini untuk mensintesis *judgment* dalam penentuan prioritas elemen-elemen pada tingkat hirarki terendah sampai pencapaian tujuan.
8. Memeriksa konsistensi hirarki. Jika nilainya lebih dari 10% maka penilaian data *judgment* harus diperbaiki.

2.3 *Website*

Yang dimaksud dengan aplikasi *Web* atau aplikasi berbasis *Web* (*Web- based application*) menurut (Janner, 2016) adalah aplikasi yang dijalankan melalui browser. Aplikasi seperti ini pertama kali dibangun hanya dengan menggunakan bahasa yang disebut HTML (*Hyper Text Markup Language*) dan protokol yang digunakan dinamakan HTTP (*Hyper Text Transfer Protocol*). Namun, tentu saja hal seperti ini memiliki kelemahan. Semua perubahan harus dilakukan pada level aplikasi. Pada perkembangan berikutnya, sejumlah skrip dan objek dikembangkan untuk memperluas kemampuan HTML.

2.4 *Database*

(Suyanto, 2017) *Database* dapat juga diartikan sebagai program. Dalam lingkungan komputer mikro, yang dimaksud database adalah sebuah program yang memungkinkan pemakai membuat dan menyimpan informasi atau melihat suatu informasi tertentu bila diperlukan.

2.5 *MySQL*

Nugroho (2015). *MySql* adalah sofware atau program aplikasi *database*, yaitu *sofware* yang dapat dipakai untuk menyimpan data berupa informasi, teks dan juga angka.

2.6 PHP (Personal Home Page)

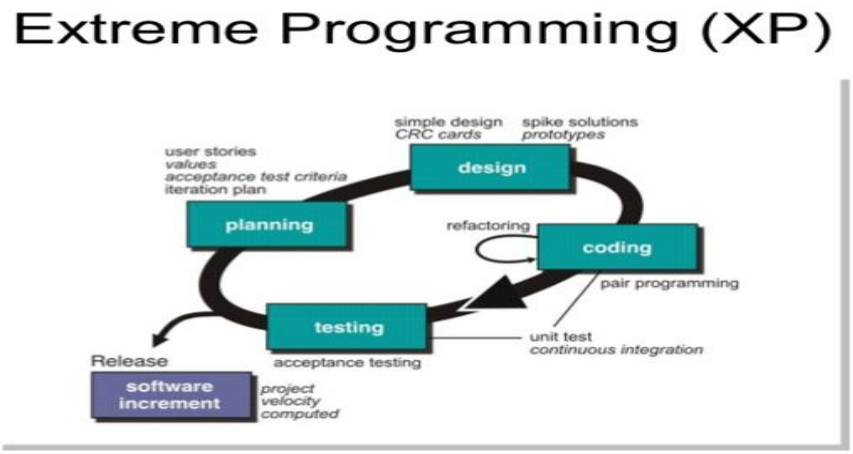
Betha (2016 : 4). PHP merupakan secara umum dikenal sebagai bahasa pmrograman scrip- scrip yang membuat dokumen HTML secara *on the fly* yang dieksekusi di sever web, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML. Dikenal juga sebagai bahasa pemrograman *server side*.

2.7 Xampp

Xampp merupakan paket PHP yang berbasis *Open Source* yang dikembangkan oleh sebuah komunitas *Open Source* (Nugroho, 2008:74). Penggunaan perangkat lunak *XAMPP* diawali dengan install paket *Xampp* pada halaman resmi http://www.apachefriends.org. Tersedia beberapa *update* yang dapat di *download* sesuai dengan *platform* komputer pengguna. Setelah penginstalan selesai maka pengguna dapat memulai pemrograman dengan membuka *XAMPP Control Panel* terlebih dahulu untuk mengaktifkan *service* yang disediakan seperti : *Apache, MySQL, FileZilla, Mercury* dan *Tomcat* dengan mengklik *Action* : *Start.*

2.8 Metode **Pengembangan Sistem**

Menurut Pressman (2016) *extreme programming* merupakan suatu pendekatan berorientasi objek dan sebagai pengembangan perangkat lunak cepat sedikit lebih rinci dengan tujuan memberikan ulasan secara ringkas. Pengembangan *extreme programming* dapat dilihat menggunakan suatu alur tahapan pengembangan yang dapat dilihat pada Gambar 2.1:



Gambar 2.2 *Extreme Programming*

Berdasarkan tahapan tersebut merupakan suatu paradigma yang diinginkan mencakup didalam seperangkat aturan dan praktik-praktik dalam empat konteks kegiatan kerangka kerja yaitu :

1. Perencanaan

   Customer dan XP *team* bekerja bersama untuk memustuskan bagaimana grup *story* untuk *release* berikutnya (*software increment* berikutnya) untuk dibangun oleh XP *team*. Jika komitmen telah dibuat, XP *team* akan membangun *story* dengan cara :

   Semua *story* segera diimplementasikan (dalam beberapa minggu)

   a. *Story* dengan value tertinggi akan dipindahkan dari jadwal dan diimplementasikan pertama.
   b. *Story* dengan resiko paling tinggi akan diimplementasikan terlebih dulu. Setelah *project* pertama di*release* dan di *delivery,* XP *team* memperhitungkan kecepatan *project*. Selama *development, customer,*dapat menambah *story,* merubah *value,* membagi *story* atau menghapusnya.

2. Perancangan

   Penggunaan rancangan bertujuan untuk membangun dan menggambarkan sistem yang akan dibuat guna mempermudah proses pengkodean. XP juga menggunakan *CRC card,* untuk mengenali dan mengatur *object oriented class* yang sesuai dengan *software increment*.

3. Pengkodean

   Sebelum membuat *code*¸lebih baik membuat unit *test* tiap *story* untuk dimasukkan dalam *software increment*. XP menyarankan agar dua orang

bekerja bersama pada satu computer *workstation* untuk membuat *code* dari satu *story* (*pair programing*), untuk menyediakan *real time problemsloving* dan jaminan *real time quality*. Setelah *pair programing* selesai, *code* diintegrasikan dengan kerja lainnya (*continuous integration*).

4. Pengujian

   Unit *test* yang telah dibuat harus diimplementasikan menggunakan suatu *framework* dan diatur ke dalam universal *testing suite*, integrasi dan validasi sistem dapat dilakukan setiap hari. *Customer test* (*acceptance test*) dilakukan oleh *customer* dan fokus pada keseluruhan fitur dan fungsional sistem. *Acceptance test* diperoleh dari *customer stories* yang telah diimplemetasikan sebagai bagian dari *software release*.

Kelebihan Model *Extreme Programming*:

Komunikasi dalam XP dibangun dengan melakukan pemrograman berpasangan (*pair programming*). *Developer* didampingi oleh pihak klien dalam melakukan *coding* dan *unit testing* sehingga klien bisa terlibat langsung dalam pemrograman sambil berkomunikasi dengan *developer*. Selain itu perkiraan beban tugas juga diperhitungkan.

1. Menekankan pada kesederhanaan dalam pengkodean: "*What is the simplest thing that could possibly work?*" Lebih baik melakukan hal yang sederhana dan mengembangkannya besok jika diperlukan.
2. Setiap *feed back* ditanggapi dengan melakukan *test, unit test* atau *system integration* dan jangan menunda karena biaya akan membengkak (uang, tenaga, waktu).

3. Banyak ide baru dan berani mencobanya, berani mengerjakan kembali dan setiap kali kesalahan ditemukan, langsung diperbaiki.

Kelemahan Model *Extreme Programming* :
Pengembangan exreme programming yang digunakan tentunya masih memiliki beberapa kelemahan atau kekurangan seperti berikut.

1. *Developer* harus selalu siap dengan perubahan karena perubahan akan selalu diterima.
2. Tidak bisa membuat kode yang detail di awal (*prinsip simplicity* dan juga anjuran untuk melakukan apa yang diperlukan hari itu juga).

2.9 Alat Pengembangan Sistem

Alat pengembangan sistem yang digunakan menggunakan pemodelan *Unified Modeling Language* (UML) *use case diagram* dan *activity diagram*.

2.9.1. Pengenalan Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. *UML* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak.

UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan *UML* tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya *UML* paling banyak digunakan pada metodologi berorientasi objek (Rosa A.S dan M. Shalahudin, 2014:133).

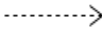
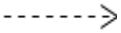
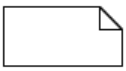
2.9.2. Diagram *UML*

Unified Modeling Language (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisi dan desain, serta menggambarkan arsitektur dalam pemograman berorientasi objek (Rosa A.S dan M. Shalahuddin, 2018). UML terdiri dari 13 macam diagram yang dikelompokan dalam tiga kategori. Pembagian kategori dan macam-macam *diagram* tersebut dapat dilihat pada gambar dibawah ini.

2.9.3. Use Case Diagram

Use case diagram merupakan titik awal yang baik dalam memahami dan menganalisis kebutuhan sistem pada saat perancangan. *Use case* diagram dapat digunakan untuk kebutuhan apa saja yang diperlukan dalam suatu sistem, sehingga sistem dapat digambarkan dengan jelas bagaimana proses dari sistem tersebut, bagaimana cara aktor menggunakan sistem, serta apa saja yang dapat dilakukan pada suatu sistem. Simbol - simbol pada *usecase* diagram dapat dilihat pada Tabel 2.1 dibawah ini.

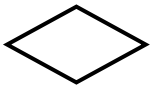
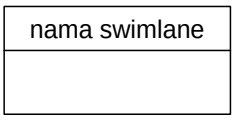
Tabel 0.1 Simbol *Use Case Diagram* (Rosa A.S dan M. Shalahuddin, 2018)

Simbol	Fungsi
	*Actor*
	*Dependency*
	*Generalization*
	*Include*
	*Extend*
	Association
	*Use Case*
	*Collaboration*
	*Note*

2.9.4. Activity Diagram

Activity diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan adalah *activity diagram* menggambarkan aktivitas sistem bukan apa yang dilakukan actor (Rosa, 2018). Simbol-simbol *activity* diagram dapat dilihat pada Tabel 2.3 dibawah ini.

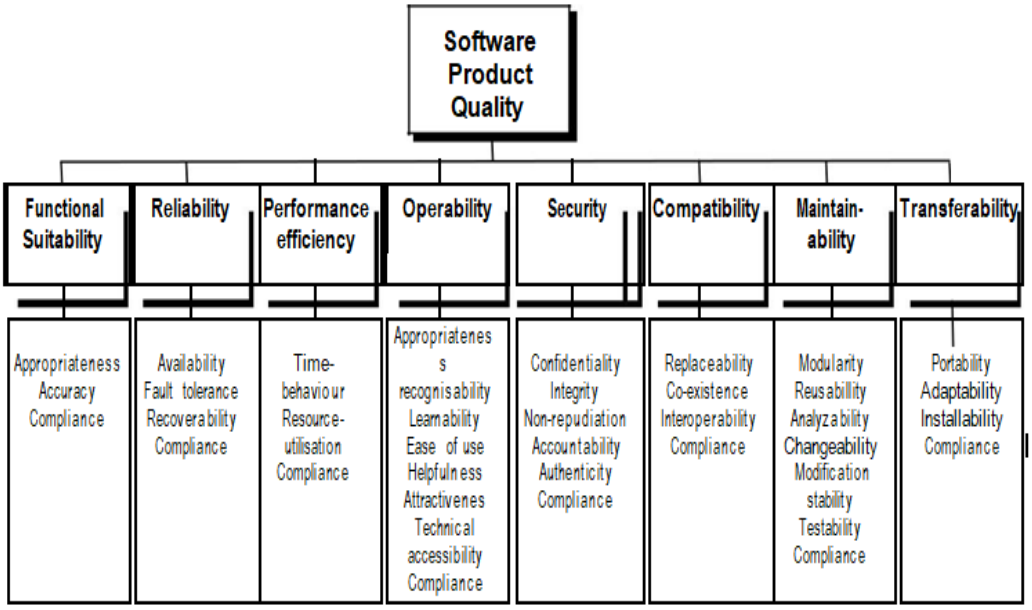
Tabel 0.3 Simbol-simbol *Activity Diagram*

Simbol	**Deskripsi**
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/*decision* 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan/*join*	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.10 Pengujian Sistem ISO 25010

Pada Penelitian ini, metode yang digunakan untuk menguji sistem ini adalah ISO 25010. Standar ISO /IEC 25010 pertama kali diperkenalkan pada tahun 1991 melalui pertanyaan tentang definisi kualitas perangkat lunak. ISO/IEC 25010 memperkenalkan tipe kualitas (*quality in use*) dimana mengikuti elemen yang telah

diketahui. ISO/IEC 25010 merupakan standar internasional yang diterbitkan oleh ISO untuk evaluasi kualitas perangkat lunak dan merupakan perkembangan dari ISO 9126. Model kualitas ISO 25010 mempunyai delapan ukuran kualitas yang ditetapkan oleh ISO/IEC 25010 yang dapat dilihat pada Gambar 2.2



. **Gambar 2.3** Karakteristik ISO 25010

1. *Functional Suitability*

Functional Suitability merupakan tingkat dimana produk perangkat lunak menyediakan fungsi yang memenuhi kebutuhan yang dinyatakan dan tersirat ketika perangkat lunak digunakan dalam kondisi tertentu. Subkarakteristik *Functional Suitability* meliputi *appropriateness, accuracy,* dan *compliance.* Berikut ini penjelasan untuk masing-masing subkarakteristik *Functional Suitability* yang dapat dilihat pada tabel 2.4.

Tabel 2.4 Penjelasan Subkarakteristik *Functional Suitability*

Karakteristik	Subkarakteristik	Penjelasan
Functional Suitability	*Appropriateness*	Sejauh mana produk perangkat lunak menyediakan fungsi yang tepat untuk tugas-tugas tertentu dan tujuan pengguna?
	accuracy	Sejauh mana produk perangkat lunak memberikan hasil yang tepat atau spesifik dengan tingkat presisi yang diperlukan?
Karakteristik	**Subkarakteristik**	**Penjelasan**
	Compliance	Pangkat dimana perangkat lunak mematuhi standar, konvensi, atau peraturan dalam undang-undang dan peraturan serupa yang berkaitan dengan kesesuaian fungsional

2. *Reliability*

Reliability didefinisikan sejauh mana perangkat lunak dapat mempertahankan tingkat kinerja dalam kondisi tertentu. Subkarakteristik *Reliability* meliputi *availability, fault tolerance, recoverability, reliability compliance.* Berikut ini penjelasan untuk masing-masing subkarakteristik *Reliability* yang dapat dilihat pada tabel 2.5

Tabel 2.5 Penjelasan Subkarakteristik *Reliability*

Karakteristik	Subkarakteristik	Penjelasan
Reliability	*Availability*	Sejauh mana komponen perangkat lunak beroperasi dan tersedia saat diperlukan untuk digunakan.
Reliability	*Fault tolerance*	Sejauh mana produk perangkat lunak dapat mempertahankan tingkat kinerja tertentu dalam kasus kesalahan perangkat lunak atau pelanggaran antarmuka yang ditentukan.
	coverability	Tingkat di mana produk perangkat lunak dapat menetapkan kembali tingkat kinerja

		ditentukan dan memulihkan data secara langsung terpengaruh kasus kegagalan.
	Reliability compliance	Sejauh mana produk perangkat lunak mematuhi standar, konvensi atau peraturan yang berkaitan dengan keandalan.

3. *Performance efficiency*

Performance efficiency merupakan sejauh mana perangkat lunak memberikan kinerja yang sesuai dan relatif terhadap jumlah sumber daya yang digunakan dalam kondisi tertentu. Subkarakteristik *Performance efficiency* meliputi *time behaviour, resource utilization, performance efficiency compliance.* Berikut ini penjelasan untuk masing-masing subkarakteristik *Performance efficiency* yang dapat dilihat pada tabel 2.6.

Tabel 2.6 Penjelasan Subkarakteristik *Performance efficiency*

Karakteristik	**Subkarakteristik**	**Penjelasan**
Performance efficiency	*Time behaviour*	Sejauh mana perangkat lunak memberikan respons yang tepat, waktu pemrosesan dan laju keluaran ketika menjalankan fungsinya.
Performance efficiency	*Resource utilization*	Sejauh manaperangkat lunak menggunakan jumlah dan jenis sumber daya yang tepat ketika perangkat lunak menjalankan fungsinya.
	Performance Efficiency Compliance	Sejauh mana perangkat lunak mematuhi standar atau

		konvensi yang berkaitan dengan efisiensi kinerja.

4. *Operability*

Operability merupakan sejauh mana produk perangkat lunak dapat dipahami, dipelajari, digunakan dan menarik bagi pengguna bila digunakan dalam kondisi tertentu. Subkarakteristik *Operability* meliputi *appropriateness recognisability, learnability, ease of use, helpfulness, attractiveness, technical accessibility,* dan *compliance.* Berikut ini penjelasan untuk masing-masing subkarakteristik *Operability* yang dapat dilihat pada tabel 2.7.

Tabel 2.7 Penjelasan Subkarakteristik *Operability*

Karakteristik	Subkarakteristik	Penjelasan
Operability	*A propriateness Recognisability*	Sejauh mana perangkat lunak memungkinkan pengguna untuk mengenali apakah perangkat lunak sesuai dengan kebutuhan pengguna?
	Learnability	jauh mana perangkat lunak pengguna untuk mempelajari aplikasinya?
	Ease of use	Sejauh mana perangkat lunak memudahkan pengguna mengoperasikan dan mengendalikan?
Operability	*Helpfulness*	Sejauh mana perangkat lunak dapat membantu pengguna?
	Attractiveness	Apakah antarmuka terlihat baik?
	Technical Accessibility	Tingkat pengoperasian perangkat lunak untuk pengguna dengan cacat tertentu?
	Compliance	Apakah perangkat lunak sudah mematuhi standar, konvensi, panduan gaya atau

		peraturan yang berkaitan dengan pengoperasian?

5. *Security*

Security merupakan perlindungan item sistem dari akses yang tidak disengaja atau berbahaya, penggunaan, modifikasi, perusakan dan pengungkapan. Subkarakteristik *Security* meliputi *confidentiality, integrity, non-repudiation, accountability, authenticity, security compliance.* Berikut ini penjelasan untuk masing-masing sub karakteristik *security* yang dapat dilihat pada tabel 2.8

Tabel 2.8 Penjelasan Subkarakteristik *Security*

Karakteristik	**Subkarakteristik**	**Penjelasan**
Security	*Confidentiality*	Sejauh mana perangkat lunak memberikan perlindungan dari pengungkapan data atau informasi yang tidak sah, baik disengaja atau disengaja.
	Integrity	Sejauh mana ketepatan dan kelengkapan aset dijaga.
	Non-repudiation	Sejauh mana tindakan atau peristiwa dapat dibuktikan telah terjadi, sehingga peristiwa atau tindakan tidak dapat ditolak.
Security	*Ac countability*	Sejauh mana tindakan suatu entitas dapat dilacak secara unik kepada entitas.
	Authenticity	Sejauh mana identitas suatu subjek atau sumber daya dapat dibuktikan sebagai yang diklaim.
	Security compliance	Sejauh mana produk perangkat lunak mematuhi standar, konvensi atau peraturan yang berkaitan dengan keamanan.

6. *Compatibility*

Compatibility merupakan kemampuan dua atau lebih komponen perangkat lunak untuk bertukar informasi dan untuk melakukan fungsi yang diperlukan saat berbagi perangkat keras atau perangkat lunak yang sama. Subkarakteristik *compatibility* meliputi *replaceability, co-existence, interoperability, compatibility compliance.* Berikut ini penjelasan untuk masing-masing subkarakteristik *compatibility* yang dapat dilihat pada tabel 2.9.

Tabel 2.9 Penjelasan Subkarakteristik *Compatibility*

Karakteristik	Subkarakteristik	Penjelasan
Compatibility	*Replaceability*	Sejauh mana perangkat lunak dapat digunakan ditempat perangkat lunak lain yang ditentukan untuk tujuan yang sama dilingkungan yang sama.
Compatibility	*Co-existence*	Sejauh mana perangkat lunak dapat bekerja sama dengan perangkat lunak independen lainnya dalam lingkungan umum berbagi sumber daya umum tanpa ada dampak yang merugikan.
	Interoperability	Sejauh mana perangkat lunak dapat dioperasikan secara kooperatif dengan satu atau lebih perangkat lunak lainnya.
	Compatibility compliance	Sejauh mana perangkat lunak mematuhi standar, konvensi atau peraturan yang berkaitan dengan kompatibilitas.

7. *Maintainability*

Maintainability merupakan sejauh mana perangkat lunak dapat dimodifikasi. Modifikasi dapat mencakup koreksi, peningkatan atau adaptasi perangkat lunak terhadap perubahan lingkungan, dan persyaratan serta spesifikasi fungsional. Subkarakteristik *Maintainability* meliputi *modularity, reusability, analyzability, changeability, modification stability, testability, maintainability compliance.* Berikut ini penjelasan untuk masing-masing subkarakteristik *maintainability* yang dapat dilihat pada tabel 2.10.

Tabel 2.10 Penjelasan Subkarakteristik *Maintainability*

Karakteristik	**Subkarakteristik**	**Penjelasan**
Maintainability	*Modularity*	
	Reusability	Sejauh mana aset dapat digunakan lebih dari satu sistem perangkat lunak, atau dalam membangun aset lainnya.
	Analyzability	Tingkat dimana perangkat lunak dapat didiagnosis untuk kekurangan atau penyebab kegagalan dalam perangkat
	Changeability	Sejauh mana perangkat lunak memungkinkan modifikasi tertentu untuk diimplementasikan.
	Modification stability	Sejauh manaperangkat lunak dapat menghindari efek tak terduga dari modifikasi perangkat lunak.
	Te stability	Sejauh mana perangkat lunak yangdimodifikasi untuk divalidasi.
	Maintainability compliance	Sejauh manaperangkat lunak mematuhi standar atau konvensi yang berkaitan dengan pemeliharaan.

8. *Transferability*

Merupakan sejauh mana perangkat lunak dapat ditransfer dari satu lingkungan ke lingkungan lain. Subkarakteristik *transferability meliputi portability, adaptability, installability, transferability compliance.* Berikut ini penjelasan untuk masing-masing sub karakteristik *transferability* yang dapat dilihat pada tabel 2.11

Tabel 2.11 Penjelasan Subkarakteristik *Transferability***.**

Karakteristik	Subkarakteristik	Penjelasan
Transferability	*Portability*	Kemudahan sistem atau komponen yang dapat ditransfer dari satu perangkat keras atau perangkat lunak ke perangkat lain
	Adaptability	Apakah perangkat lunak dapat disesuaikan dengan lingkungan tertentu yang berbeda?
	Installability	Apakah perangkat lunak dapat diinstal dan dihapus pada ingkungan tertentu?
	Transferability compliance	Apakah perangkat lunak mematuhi standar atau konvensi yang berkaitan dengan portabilitas.

Pada penelitian ini pengujian berfokus pada karakteritik *Functional* dan *Operability.*

Skala pengukuran yang digunakan peneliti adalah skala *likert*. Menurut Djaali (2008) skala *likert* adalah skala yang digunakan untuk mengukur persepsi atau pendapat sesorang mengenai sebuah peristiwa sosial, berdasarkan definisi operasional yang telah ditetapkan oleh peneliti. Skala ini merupakan suatu skala psikometri yang biasa diaplikasikan dalam angket dan paling sering digunakan untuk riset yang berupa survei. Jadi peneliti menggunakan skala *likert* untuk menilai sejauh mana responden setuju atau tidak setuju.