

BAB 2 TINJAUAN PUSTAKA

1.1 Penelitian Terdahulu

Sub bab ini berisikan data penelitian terdahulu yang terkait dengan topik penelitian. Pembahasan penelitian terdahulu digunakan sebagai acuan atau dasar referensi kegiatan penelitian ini agar dapat menghasilkan penelitian yang lebih baik.

Tabel 2.1 Referensi Penelitian Terdahulu

No.	Nama Penulis dan Tahun Penelitian	Judul Penelitian	Kesimpulan
1.	R Moh Andriawan Adikara (2018)	Sistem Pendukung Keputusan Pemilihan Varietas Unggul Jagung Hibrida Menggunakan Metode Ahp-Smart	Berdasarkan hasil penelitian diatas diambil beberapa kesimpulan yaitu Sistem pemilihan pemilihan bibit jagung terunggul dapat ditentukan secara cepat dan mudah. Hasil yang ditentukan lebih efisien akurat. Penentuan bobot dari kriteria yang digunakan sangat mempengaruhi hasil dari nilai perhitungan pada metode PSI dan Sistem Pendukung Keputusan menggunakan metode PSI ini sangatlah tepat dalam pemilihan bibit jagung terunggul dan dapat membantu bagi pertanian dan dapat meningkatkan kesejahteraan petani dan mencerdaskan petani itu sendiri.

2.	YUNUS ADI PRASETYO	Sistem Pendukung Keputusan Pemilihan Bibit Padi Unggul Menggunakan Metode Simple Additive Weight (Saw)	Dengan adanya sistem pendukung keputusan pemilihan bibit padi unggul yang menggunakan metode SAW, para petani dapat memiliki panduan dan menjadi lebih mudah dalam memilih benih yang bagus untuk penanaman padi mereka. Sehingga dapat menghasilkan produksi padi yang berkualitas baik.
3.	Chairul Imam (2019)	Sistem Pendukung Keputusan Spesifikasi Biji Jagung Berkualitas Terbaik Dengan Metode Multi Attribute Utility Theory	Hasil perhitungan menggunakan metode attribute utility theory spesifikasi biji jagung berkualitas terbaik, didapatkan hasil akurasi manual dan dengan software akurasi 86.7% sehingga menggunakan cara ini akan lebih mempermudah penyelesaian masalah pada spesifikasi biji jagung berkualitas terbaik di PT Charoen Pokphand Indonesia
4	Ferly ardhya (2018)	Sistem Pengambilan Keputusan Pemilihan Bibit Jagung Dengan Metode Analytical Hierarchy Process (Ahp) Pada Toko Abadi Jaya Lampung Timur	Dengan menggunakan SPK Pemilihan Bibit Jagung maka diharapkan agar dapat membantu proses pengolahan data secara efektif dan efisien,. Lebih membantu petani dalam memilih bibit jagung yang sesuai sehingga meningkatkan hasil panen.

5	Agung Prasetyo (2020)	Implementasi Metode Analytical Hierarchy Process Dalam Pemilihan Varietas Unggul Tanaman Kedelai Di Balitkabi Malang	penelitian ini mengusulkan pengembangan sistem penunjang keputusan pemilihan varietas unggul tanaman kedelai dengan menggunakan metode Analytical Hierarchy Process (AHP). Hasil pemilihan varietas unggul tanaman kedelai oleh penyuluh dan hasil pemilihan varietas unggul tanaman kedelai dengan menggunakan sistem penunjang keputusan metode AHP menghasilkan rata-rata akurasi sebesar 60,842%. Hasil tersebut didapat dari lima kali pengujian yang dilakukan, yang artinya hasil pemilihan varietas unggul tanaman kedelai oleh penyuluh failure atau gagal.
	zhar (2020)	Faktor Analisis Prioritas Dalam Pemilihan Bibit Jagung Unggul Menggunakan Metode AHP	Penelitian ini bertujuan untuk menganalisis faktor/kriteria prioritas dalam pemilihan bibit jagung unggul dengan metode Analytical Hierarchy Process. Berdasarkan hasil analisis perhitungan yang telah dilakukan, maka secara berurutan kriteria yang prioritas tertinggi yaitu kriteria ketahanan terhadap hama, hasil produksi , ukuran buah, waktu panen dan adaptasi Lingkungan
	Dedi Kurniawan (2015)	Perancangan Sistem Pendukung Keputusan Pemilihan Bibit Padi Unggul Menggunakan Metode Ahp	Penelitian ini menghasilkan Sistem pendukung keputusan dengan menggunakan metode AHP (Analytic Hierarchy Process) yang akan memberikan informasi dan membantu pengguna dalam pengambilan keputusan mengenai benih padi yang akan mereka gunakan. Dengan adanya sistem pendukung keputusan

			<p>pemilihan bibit padi unggul yang menggunakan metode AHP, para petani dapat memiliki panduan dan menjadi lebih mudah dalam memilih benih yang bagus untuk penanaman padi mereka. Sehingga dapat menghasilkan produksi padi yang berkualitas baik.</p>
	Ferda Fandesa (2018)	<p>Penerapan Sistem Pendukung Keputusan Pada Pemilihan Varietas Padi Unggul Dengan Menggunakan Metode <i>Analytical Hierarchy Process</i></p>	<p>Penelitian ini bertujuan untuk membantu petani memilih varietas bibit padi unggul dengan metode AHP. Berdasarkan hasil pengujian yang telah dilakukan, terdapat 19 varietas Unggulan dan 11 varietas tidak unggul. Berdasarkan perbandingan dengan Hasil penelitian yang dilakukan oleh bbpadi terdapat 24 data varietas yang sesuai dan 6 data varietas tidak sesuai.</p>
	Muhammad Bagus Nugroho (2019)	<p>Implementasi Metode Analytical Hierarchy Process (Ahp) Untuk Pemilihan Jenis Bibit Jagung Terbaik</p>	<p>Pemilihan jenis jagung tersebut menjadi permasalahan karena banyaknya kriteria yang perlu dipertimbangkan. Permasalahan tersebut dapat dipecahkan dengan adanya sistem pendukung keputusan. Dalam penelitian ini sistem dibuat dengan metode Analytic Hierarchy Process (AHP). Sistem ini mampu menghasilkan keputusan yang akurat dan cepat. Hasil dari sistem berupa perankingan jenis jagung terbaik hingga yang paling rendah sesuai dengan kriteria yang telah diberikan. Metode ini dipilih karena dapat menghasilkan keputusan yang lebih akurat dan lebih objektif. Sedangkan hasil dari sistem yang dibuat berupa</p>

			peringkat jenis bibit jagung mulai dari yang paling baik hingga yang paling buruk dengan presentase kemudahan dalam menggunakan sistem
	Yartiwi, Yahumri (2018)	Tingkat Keputusan Petani Dalam Pemilihan Sistem Tanam Dan Varietas Pada Teknologi Tumpangsari Jagung-Kedelai Dengan Metode Ahp	Penelitian bertujuan untuk mengukur tingkat keputusan petani terhadap pemilihan sistem tanam tumpangsari jagung-kedelai, serta pemilihan varietas jagung dan kedelai. Analisis menggunakan metode <i>Analytical Hierarchy Process</i> (AHP) dengan menggunakan <i>Expert Choice</i> Versi 11.0 untuk mengambil keputusan dengan melakukan perbandingan berpasangan antara kriteria pilihan dan juga perbandingan berpasangan antara pilihan yang ada. Kemudian informasi yang ada disintesis untuk menentukan peringkat relatif dari alternatif pilihan yang ada. Hasil penelitian menunjukkan bahwa tingkat keputusan petani lebih memilih sistem monokultur sebanyak 81,0% dan sistem tumpangsari sebanyak 19,0%.

2.1.1 Komperasi Dengan Penelitian Terdahulu

Penelitian ini memiliki perbedaan pada penggunaan metode *fuzzy AHP* sebagai dasar analisa penentuan bibit jagung varietas *hibrida* yang tepat. Metode *fuzzy AHP* belum pernah digunakan pada 10 penelitian pembandingan, sehingga perlu dilakukan penelitian untuk mengetahui ketepatan hasil analisa. Selain pada penggunaan metode, penelitian ini memiliki keunggulan pengaturan kriteria dan bobot kriteria yang dinamis, sehingga sistem dapat tetap digunakan apabila diperlukan penambahan kriteria untuk lebih menghasilkan hasil analisa yang tepat. Sistem pada penelitian ini dibuat dengan menggunakan bahasa pemrograman web *responsive*, sehingga sistem dapat tetap digunakan oleh pengguna melalui perangkat mobile.

1.2 Sistem

Sistem berasal dari bahasa Latin (*systema*) dan bahasa Yunani (*systema*). Sistem adalah suatu kumpulan atau himpunan dari unsur, komponen atau variable-variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu. Sistem merupakan suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu (Jogiyanto, 2017). Jadi sistem adalah kumpulan dari komponen atau modul kerja yang saling mendukung untuk menghasilkan keluaran yang diharapkan.

1.3 Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (SPK) adalah proses pengambilan keputusan dibantu menggunakan komputer untuk membantu pengambil keputusan dengan menggunakan beberapa data dan model tertentu untuk menyelesaikan beberapa masalah yang tidak terstruktur (Kosasi, 2017). Keberadaan SPK pada perusahaan atau organisasi bukan untuk menggantikan tugas-tugas pengambil keputusan, tetapi merupakan sarana yang membantu bagi mereka dalam pengambilan keputusan, dengan menggunakan data-data yang diolah menjadi informasi untuk mengambil keputusan dari masalah-masalah semi-terstruktur. Pada implementasi SPK, hasil dari keputusan-keputusan dari sistem bukanlah hal yang menjadi patokan, pengambilan keputusan tetap berada pada pengambil keputusan. Sistem hanya menghasilkan keluaran yang mengkalkulasi data-data sebagaimana pertimbangan seorang pengambil keputusan, sehingga kerja pengambil keputusan dalam mempertimbangkan keputusan dapat dimudahkan. SPK berfungsi sebagai tambahan atau pendukung bagi pembuat keputusan, dapat memperluas pengetahuan dan kemungkinan, namun tidak menggantikan penilaian. Sistem ini ditujukan untuk keputusan yang membutuhkan penilaian dan keputusan yang dapat diolah dengan algoritma atau secara teknis.

Sistem Pendukung Keputusan dirancang untuk mendukung seluruh tahap pengambilan keputusan mulai dari mengidentifikasi masalah, memilih data yang relevan, dan menentukan pendekatan yang digunakan dalam proses pengambilan keputusan sampai mengevaluasi pemilihan alternatif-alternatif yang ada (Surbakti, 2015). Adapun karakteristik sistem pendukung keputusan yaitu:

1. Sistem Pendukung Keputusan dirancang untuk membantu pengambil keputusan dalam memecahkan masalah yang sifatnya semi terstruktur ataupun tidak terstruktur dengan menambahkan kebijaksanaan manusia dan informasi komputerisasi.

2. Dalam proses pengolahannya, sistem pendukung keputusan mengkombinasikan penggunaan model-model analisis dengan teknik pemasukan data konvensional serta fungsi-fungsi pencari/interogasi informasi.
3. Sistem Pendukung Keputusan, dirancang sedemikian rupa sehingga dapat digunakan/dioperasikan dengan mudah.
4. Sistem Pendukung Keputusan dirancang dengan menekankan pada aspek fleksibilitas serta kemampuan adaptasi yang tinggi.

Dengan berbagai karakter khusus di atas, SPK dapat memberikan berbagai manfaat dan keuntungan. Adapun manfaat yang dapat diambil dari SPK yaitu:

1. SPK memperluas kemampuan pengambil keputusan dalam memproses data/informasi bagi pemakainya.
2. SPK membantu pengambil keputusan untuk memecahkan masalah terutama berbagai masalah yang sangat kompleks dan tidak terstruktur.
3. SPK dapat menghasilkan solusi dengan lebih cepat serta hasilnya dapat diandalkan.

1.4 Analytical Hierarchy Process (AHP)

AHP dikembangkan oleh Thomas Saaty pada tahun 1970-an dalam menghadapi masalah perencanaan militer Amerika Serikat untuk menghadapi berbagai pilihan (*contingency planning*) (Suryadi., 2017). AHP merupakan sistem pembuat keputusan dengan menggunakan model matematis (Kusrini, 2015). Pada umumnya AHP digunakan dengan tujuan untuk menyusun prioritas dari berbagai alternatif pilihan yang ada dan pilihan-pilihan tersebut bersifat kompleks atau multikriteria (Suryadi., 2017). Beberapa kelebihan penggunaan metode AHP adalah sebagai berikut:

Struktur yang berbentuk hirarki sebagai konsekuensi dari kriteria yang dipilih sampai pada subkriteria yang paling dalam.

1. Memperhatikan validitas sampai dengan batas toleransi inkonsistensi berbagai kriteria dan alternatif yang dipilih oleh para pengambil keputusan.
2. Memperhitungkan daya tahan atau ketahanan keluaran analisis sensitivitas pembuat keputusan.

Selain itu metode AHP mempunyai kemampuan untuk memecahkan masalah yang multi-objektif dan multikriteria yang berdasar pada perbandingan preferensi dari

setiap elemen dalam hirarki. Jadi metode AHP merupakan suatu bentuk pemodelan pembuatan keputusan yang sangat komprehensif. Terdapat 4 aksioma-aksioma yang terkandung dalam model AHP:

1. *Reciprocal Comparison* artinya pengambilan keputusan harus dapat memuat perbandingan dan menyatakan preferensinya. Preferensi tersebut harus memenuhi syarat resiprokal yaitu apabila A lebih disukai daripada B dengan skala x , maka B lebih disukai daripada A dengan skala $1/x$.
2. *Homogeneity* artinya preferensi seseorang harus dapat dinyatakan dalam skala terbatas atau dengan kata lain elemen- elemennya dapat dibandingkan satu sama lainnya. Kalau aksioma ini tidak dipenuhi maka elemen- elemen yang dibandingkan tersebut tidak homogen dan harus dibentuk cluster (kelompok elemen) yang baru.
3. *Independence* artinya preferensi dinyatakan dengan mengasumsikan bahwa kriteria tidak dipengaruhi oleh alternatif-alternatif yang ada melainkan oleh objektif keseluruhan. Ini menunjukkan bahwa pola ketergantungan dalam AHP adalah searah, maksudnya perbandingan antara elemen-elemen dalam satu tingkat dipengaruhi atau tergantung oleh elemen-elemen pada tingkat di atasnya.
4. *Expectation* artinya untuk tujuan pengambil keputusan. Struktur hirarki diasumsikan lengkap. Apabila asumsi ini tidak dipenuhi maka pengambil keputusan tidak memakai seluruh kriteria atau objektif yang tersedia atau diperlukan sehingga keputusan yang diambil dianggap tidak lengkap.

Pada dasarnya terdapat beberapa langkah yang perlu diperhatikan dalam menggunakan metode AHP, antara lain (Kosasi, 2017):

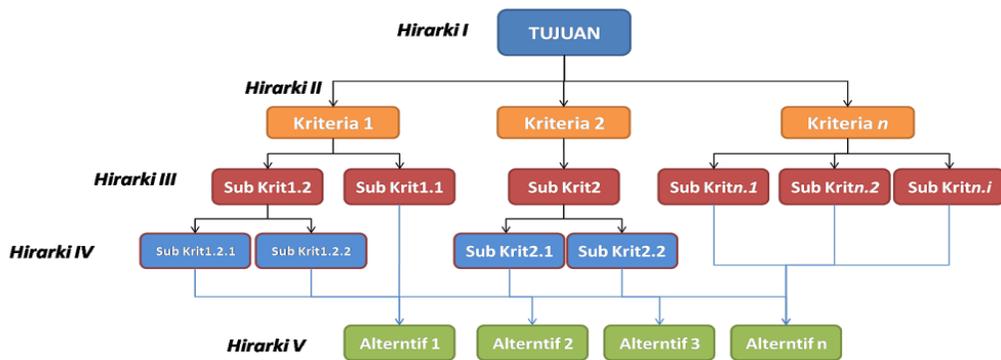
1. Mendefinisikan masalah dan menentukan solusi yang diinginkan.
2. Membuat struktur hirarki yang diawali dengan tujuan umum dilanjutkan dengan subtujuan-subtujuan, kriteria dan kemungkinan alternatif-alternatif pada tingkatan kriteria yang paling bawah.
2. Membuat matriks perbandingan berpasangan yang menggambarkan kontribusi relatif atau pengaruh setiap elemen terhadap masing-masing tujuan atau kriteria yang setingkat di atasnya. Perbandingan dilakukan berdasarkan judgment dari pembuat keputusan dengan menilai tingkat kepentingan suatu elemen dibandingkan elemen lainnya.
3. Melakukan perbandingan berpasangan sehingga diperoleh nilai judgment seluruhnya yaitu sebanyak $n \times [(n-1)/2]$ buah dengan n adalah banyaknya elemen yang dibandingkan.
4. Menghitung nilai λ_{max} dan menguji konsistensinya jika tidak konsisten maka pengambilan data diulangi.

5. Mengulangi langkah 3, 4 dan 5 untuk seluruh tingkat hirarki.
6. 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.
7. Memeriksa konsistensi hirarki. Jika nilai lebih dari 10% (persen) atau 0,1 maka penilaian data harus diperbaiki.

Dalam menyelesaikan permasalahan dengan AHP ada beberapa prinsip yang harus dipahami, diantaranya adalah (Kosasi, 2017):

1. Membuat Hirarki

Sistem yang kompleks bisa dipahami dengan memecahnya menjadi elemen - elemen pendukung, menyusun elemen secara hirarki, dan menggabungkannya atau mensistesisnya. Untuk lebih jelasnya dapat dilihat pada gambar 2.1



Gambar 2.1 Struktur Hirarki AHP (Kosasi, 2017)

2. Penilaian kriteria dan alternative

Kriteria dan alternatif dilakukan dengan perbandingan berpasangan. Untuk berbagai persoalan, skala 1 sampai 9 adalah skala terbaik untuk mengekspresikan pendapat. Nilai dan definisi pendapat kualitatif dari skala perbandingan Saaty dapat dilihat pada Tabel di bawah ini:

Tabel 2.2 Skala Penilaian Perbandingan (Kosasi, 2017)

Intensitas Kepentingan	Keterangan
1	Kedua elemen sama pentingnya.

3	Elemen yang satu sedikit lebih penting daripada elemen yang lainnya.
5	Elemen yang satu lebih penting daripada elemen yang lainnya.
7	Satu elemen jelas lebih mutlak penting daripada elemen yang lainnya.
9	Satu elemen mutlak penting daripada elemen yang lainnya.
2,4,6,8	Nilai-nilai antara dua nilai pertimbangan yang berdekatan.
Kebalikan	Jika aktivitas i mendapat satu angka dibandingkan dengan aktivitas j, maka j memiliki nilai kebalikannya dibandingkan dengan i.

Pengisian nilai tabel perbandingan berpasangan dilakukan berdasarkan kebijakan pembuat keputusan dengan melihat tingkat kepentingan antar satu elemen dengan elemen yang lainnya. Proses perbandingan berpasangan, dimulai dari perbandingan kriteria misalnya A1, A2 dan A3. Maka susunan elemen-elemen yang dibandingkan tersebut akan tampak seperti pada tabel 2 di bawah ini:

Tabel 2.3 Contoh Matriks Perbandingan Berpasangan (Kosasi, 2017)

	A1	A2	A3
A1	1		
A2		1	
A3			1

Berdasarkan tabel 2.2, Apabila suatu elemen dibandingkan dengan dirinya sendiri maka diberi nilai 1. Jika elemen i dibandingkan dengan elemen j mendapatkan nilai tertentu, maka elemen j dibandingkan dengan elemen i merupakan kebalikannya. Pengujian konsistensi dilakukan terhadap perbandingan antar elemen yang didapatkan pada tiap tingkat hirarki. Konsistensi perbandingan ditinjau dari per matriks perbandingan dan keseluruhan hirarki untuk memastikan bahwa urutan prioritas yang dihasilkan didapatkan dari suatu rangkaian perbandingan yang masih berada dalam

batas-batas preferensi yang logis. Setelah melakukan perhitungan bobot elemen, langkah selanjutnya adalah melakukan pengujian konsistensi matriks. Untuk melakukan perhitungan ini diperlukan bantuan tabel *Random Index* (RI) yang nilainya untuk setiap ordo matriks dapat dilihat pada tabel berikut ini:

Tabel 2.4 Tabel Nilai Random Index (Kosasi, 2017)

Urutan Matriks	1	2	3	4	5	6	7	8	9	10
(RI)	0	0	0	0	1	1	1	1	1	1

	0	0	5	9	1	2	3	4	4	4
	0	1	8	0	2	4	2	1	5	9

3. *Synthesis of Priority* (Penentuan Prioritas)

Setiap kriteria dan alternatif, perlu dilakukan perbandingan berpasangan (*Pairwise Comparisons*). Nilai-nilai perbandingan relatif dari seluruh alternatif kriteria bisa disesuaikan dengan judgement yang telah ditentukan untuk menghasilkan bobot dan prioritas. Bobot atau prioritas dihitung dengan manipulasi matriks atau melalui penyelesaian persamaan matematika.

4. *Logical Consistency* (Konsistensi Logis)

Konsistensi memiliki dua makna, pertama, objek-objek yang serupa bisa dikelompokkan sesuai dengan keseragaman dan relevansi. Kedua, menyangkut tingkat hubungan antar objek yang didasarkan pada kriteria tertentu. Penggunaan metode AHP biasa digunakan untuk permasalahan pemilihan alternatif yang memiliki multikriteria. Setiap kriteria dan sub kriteria memiliki bobot, sebagai acuan prioritas perbandingan antar kriteria atau sub kriteria.

1.5 Unified Modelling Language

Unified Modeling Language (UML) merupakan sistem arsitektur yang bekerja dalam *OOAD* (*Object-Oriented Analysis/Design*) dengan satu bahasa yang konsisten untuk menentukan, visualisasi, mengkontruksi, dan mendokumentasikan *artifact* (sepotong informasi yang digunakan atau dihasilkan dalam suatu proses rekayasa *software*, dapat berupa model, deskripsi, atau *software*) yang terdapat dalam sistem

software (Sugiarti, 2015). *UML* merupakan bahasa pemodelan yang paling sukses dari tiga metode OO yang telah ada sebelumnya, yaitu *Booch*, *OMT (Object Modeling Technique)*, dan *OOSE (Object-Oriented Software Engineering)* (Amrullah, 2016). *UML* merupakan kesatuan dari ketiga pemodelan tersebut dan ditambah kemampuan lebih karena mengandung metode tambahan untuk mengatasi masalah pemodelan yang tidak dapat ditangani ketiga metode tersebut. *UML* dikeluarkan oleh *OMG (Object Management Group, Inc)* yaitu organisasi internasional yang dibentuk pada 1989, terdiri dari perusahaan sistem informasi, *software developer*, dan para *user* sistem komputer.

Dengan adanya *UML*, diharapkan dapat mengurangi kekacauan dalam bahasa pemodelan yang selama ini terjadi dalam lingkungan industri. *UML* diharapkan juga dapat menjawab masalah penotasian dan mekanisme tukar menukar model yang terjadi selama ini. Tujuan penggunaan diagram *UML* diantaranya adalah sebagai berikut (Sholih, 2016):

1. Memberikan model yang siap pakai, bahasa pemodelan *visual* yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
 2. Memberikan bahasa pemodelan yang bebas dari berbagai Bahasa pemrograman dan proses rekayasa.
 3. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan
- Jenis-jenis dari sub diagram *Unified Modelling Language (UML)* adalah sebagai berikut.

2.5.1 Use Case Diagram

Use case diagram adalah rangkaian/uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. *Use Case Diagram* merupakan rangkaian tindakan yang dilakukan oleh sistem, aktor mewakili *user* atau sistem lain yang berinteraksi dengan sistem yang dimodelkan (Munawar, 2015).

Tabel 2.5 Tipe Relasi pada Use Case Diagram (Munawar, 2015)

NO	GAMBAR	NAMA	KETERANGAN
----	--------	------	------------

1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

2.5.2 Activity Diagram

Activity Diagram atau Diagram Aktivitas adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus (Munawar, 2015). Diagram aktivitas mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah diagram aktivitas bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa. Sebuah *activity diagram* memiliki tipe relasi sebagai berikut ini.:

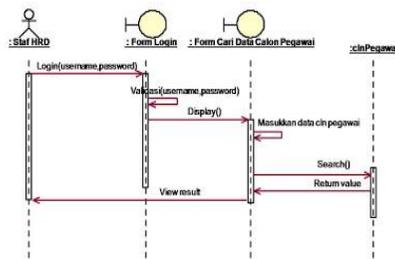
Tabel 2.6 Tipe Relasi pada Activity Diagram (Munawar, 2015)

Simbol	Penjelasan
	Mempresentasikan dimulainya alur kerja suatu sistem dalam <i>activity diagram</i> .
	suah <i>state</i> yang menggambarkan eksekusi dari aksi <i>atomic</i> .
	Mengidentifikasi bahwa suatu objek dari <i>state</i> pertama akan menampilkan aksi-aksi tertentu dan memasuki <i>state</i> kedua ketika peristiwa terjadi pergerakan dari aksi ke aksi lainnya.
	Menentukan kapan alur dalam aktivitas menjadi bercabang.
	Adanya percabangan paralel dari aktivitas

■	
●	Mempresentasikan bahwa telah diakhirinya alur suatu sistem dalam <i>activity diagram</i> .

2.5.3 Sequence Diagram

Merupakan diagram yang menunjukkan aliran fungsionalitas dalam *use case* (Munawar, 2015). *Sequence* adalah satu dari dua interaksi diagram yang mengilustrasikan objek-objek yang berhubungan dengan *use case* dan *message* atau pesan-pesannya. Komponen utama *sequence* diagram terdiri atas objek yang dituliskan dengan kotak segiempat bernama *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan progress *vertical*.



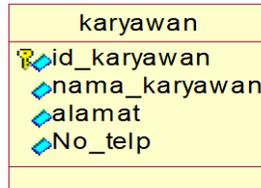
Gambar 2.2 Contoh Sequence Diagram (Munawar, 2015)

2.5.4 Class Diagram

Diagram kelas atau *Class diagram* sangat membantu dalam visualisasi struktur kelas dari suatu sistem (Munawar, 2015). Hal ini disebabkan karena *class* adalah deskripsi kelompok obyek-obyek dengan *property*, operasi dan relasi yang sama. Disamping itu diagram kelas bisa memberikan pandangan global atas sebuah sistem. Hal tersebut tercermin dari *class-class* yang ada dan relasinya satu dengan lainnya. Itulah sebabnya diagram kelas menjadi diagram yang paling populer di *UML*. Komponen diagram kelas adalah sebagai berikut ini.

1. Asosiasi adalah *class-class* yang berhubungan satu sama lain secara konseptual yaitu menghubungkan dua kelas menjadi satu asosiasi.

2. Atribut adalah properti dari sebuah kelas. Atribut ini menjelaskan batas nilai yang mungkin ada pada obyek dari kelas. Sebuah kelas mungkin mempunyai nol atau lebih atribut. Contoh dari atribut :



Gambar 2.3 Contoh Atribut class diagram (Munawar, 2015)

3. Operasi adalah sesuatu yang bisa dilakukan oleh sebuah *class* atau yang anda (atau *class* yang lain) dapat lakukan untuk sebuah *class*. Sama halnya dengan atribut, kita bisa juga memberikan tambahan informasi untuk operasi dengan menambahkan parameter yang akan dilakukan oleh operasi dengan tanda kurung.

2.6 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah gambaran pada sistem dimana di dalamnya terdapat hubungan antara *entity* beserta relasinya (Indrajani., 2015). *Entity* merupakan sesuatu yang ada dan terdefinisikan di dalam suatu organisasi, dapat abstrak dan nyata. Setiap *entity* mempunyai *atribute* yang merupakan ciri *entity* tersebut. Relasi adalah hubungan antar *entity* yang berfungsi sebagai hubungan yang mewujudkan pemetaan antar *entity*.

1. *Simple Attribute*

Attribute ini merupakan *attribute* yang unik dan tidak dimiliki oleh *attribute* lainnya, misalnya *entity* mahasiswa yang *attribute*-nya NIM.

2. *Composite Attribute*

Composite attribute adalah *attribute* yang memiliki dua nilai harga, misalnya nama besar (nama keluarga) dan nama kecil (nama asli).

3. *Single Value Attribute*

Atribute yang hanya memiliki satu nilai harga, misalnya *entity* mahasiswa dengan *attribute*-nya Umur (tanggal lahir).

4. *Multi Value Attribute*

Multi value attribute adalah *attribute* yang banyak memiliki nilai harga, misalnya *entity* mahasiswa dengan *attribute*-nya pendidikan (SD, SMP, SMA).

5. *Null Value Attribute*

Null value attribute adalah *attribute* yang tidak memiliki nilai harga, misalnya *entity* tukang becak dengan *attribute*-nya pendidikan (tanpa memiliki ijazah).

ERD dibagi menjadi dua jenis model, yaitu (Indrajani., 2015):

1. *Conceptual Data Model (CDM)*

Conceptual Data Model (CDM) adalah jenis model data yang menggambarkan hubungan antar tabel secara konseptual.

2. *Physical Data Model (PDM)*

Physical Data Model (PDM) adalah jenis model data yang menggambarkan hubungan antar tabel secara fisik.

2.7 **Peripheral Hypertext Preprocessor (PHP)**

PHP merupakan singkatan dari *Peripheral Hypertext Preprocessor*, sebuah bahasa scripting yang terpasang pada HTML (Kadir, 2014). Sebagian besar sintaks mirip dengan bahasa C, Java, asp dan Perl, ditambah beberapa fungsi PHP yang spesifik. Tujuan utama bahasa pemrograman *PHP* dibuat untuk memungkinkan perancang web untuk menulis halaman web dinamik (Solichin, 2016). Seluruh aplikasi berbasis web dapat dibuat dengan PHP. Namun kekuatan yang paling utama PHP adalah pada konektivitasnya dengan system database di dalam web. Sistem database yang dapat didukung oleh PHP adalah Oracle MySQL, Ms. Access, Sybase PostgreSQL, dan lainnya. PHP dapat berjalan di berbagai system operasi seperti Windows 98/NT, UNIX/LINUX, Solaris maupun Macintosh (Hidayatullah, 2017). Keunggulan lainnya dari PHP adalah PHP juga mendukung komunikasi dengan layanan seperti protocol IMAP, SNMP, NNTP, POP3 bahkan HTTP.

2.8 **MySQL**

MySQL adalah *Relational Database Management System (RDBMS)* yang didistribusikan secara gratis dibawah lisensi *GPL (General Public License)* (Kadir, 2014). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat closed source atau komersial. MySQL merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis (Solichin, 2016). Keandalan suatu sistem database (DBMS) dapat diketahui dari cara kerja optimizernya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya. MySQL biasanya digunakan atau diinstall bersamaan dengan XAMPP sehingga untuk melihat isi table bias menggunakan PHPmyAdmin.

Sebagai software database dengan konsep database modern, MySQL memiliki banyak kelebihan antara lain (Indrajani., 2015).

1. Protability

MySQL dapat digunakan dengan stabil tanpa kendala, berarti pada berbagai sistem operasi diantaranya seperti Windows, Linux, Mac OS X Server, Solaris, Amiga HP-UX dan masih banyak lagi. Open source MySQL didistribusikan secara open source di bawah lisensi GPL, sehingga dapat memperoleh menggunakannya secara cuma-cuma tanpa dipungut biaya sepeserpun.

2. Multi user

MySQL dapat digunakan untuk menangani beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik. Hal ini akan memungkinkan sebuah database server MySQL dapat diakses client secara bersamaan dalam waktu yang bersamaan pula.

3. Performance Tuning

MySQL memiliki kecepatan yang cukup menakjubkan dalam menangani query sederhana, serta mampu memproses lebih banyak SQL persatuan waktu.

4. Column Types

MySQL didukung tipe kolom (tipe data) yang sangat kompleks.

5. Command dan Functions

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

6. Scalability dan Limits

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

7. Interface

Sama halnya dengan software database lainnya, MySQL memiliki interface (antarmuka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).

8. Struktur tabel

Struktur tabel MySQL cukup baik, serta cukup fleksibel. Misalnya ketika menangani Alter Table, dibandingkan database lainnya semacam ProgresSQL ataupun Oracle.

2.9 *Hypertext Markup Language (HTML)*

Hypertext Markup Language (HTML) adalah bahasa yang digunakan untuk menulis halaman web (M.Rudyanto, 2015). *HTML* merupakan pengembangan dari standar pemformatan dokumen teks yaitu *Standard Generalized Markup Language (SGML)* (Kadir, 2012:58). *HTML* sebenarnya adalah dokumen ASCII atau teks biasa, yang dirancang untuk tidak tergantung pada suatu sistem operasi tertentu. Mendesain *HTML* dapat dilakukan dengan cara menggunakan *HTML Editor*, seperti *Microsoft FrontPage*, *Adobe Dreamweaver*, dan lain-lain. Dalam pemrograman web, kode *HTML* wajib digunakan untuk melakukan desain tampilan sistem yang dibuat. Dengan menggunakan *HTML Editor* moderen, programmer tidak perlu lagi melakukan pengetikan kode *HTML*, melainkan langsung melakukan desain tampilan sistem secara *drag and drop*.

2.10 *Cascading Sytle Sheets (CSS)*

Cascading Style Sheets (CSS) adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa markup (Abdul Kadir, 2012:61). Penggunaan yang paling umum dari CSS adalah untuk memformat halaman web yang ditulis dengan HTML dan XHTML (Shalahuddin, 2012:32). Walaupun demikian, bahasanya sendiri dapat dipergunakan untuk semua jenis dokumen XML termasuk SVG dan XUL. Spesifikasi CSS diatur oleh World Wide Web Consortium (W3C) (M.Rudyanto, 2015). CSS merupakan pelengkap *HTML* untuk memberikan tampilan yang lebih menarik seperti menentukan warna, jenis huruf dan tata letak komponen di halaman website. Oleh karena itu pengkodean CSS menjadi satu bagian dalam melakukan pembuatan halaman website.

