

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Teori Umum

Dalam bagian ini, penulis akan menjelaskan mengenai teori-teori umum menurut pendapat ahli.

2.1.1. Sistem Informasi

Menurut (Husda, 2012, p. 119) Sistem informasi dapat didefinisikan sebagai suatu sistem di dalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan.

Informasi dalam suatu lingkungan sistem informasi harus mempunyai persyaratan umum sebagai berikut:

1. Harus diketahui oleh penerima sebagai referensi yang tepat.
2. Harus sesuai dengan kebutuhan yang ada dalam proses pembuatan/pengambilan keputusan.
3. Harus mempunyai nilai *surprise*, yaitu hal yang sudah diketahui hendaknya jangan diberikan.

4. Harus dapat menuntun pemakai untuk membuat keputusan. Suatu keputusan tidak selalu menuntut adanya tindakan.

Sistem informasi harus mempunyai beberapa sifat seperti:

1. Pemrosesan informasi yang efektif. Hal ini berhubungan dengan pengujian terhadap data yang masuk, pemakaian perangkat keras dan perangkat lunak yang sesuai.
2. Manajemen informasi yang efektif. Dengan kata lain, operasi manajemen, keamanan dan keutuhan data yang ada harus diperhatikan.
3. Keluwesan. Sistem informasi hendaknya cukup luwes untuk menangani suatu macam operasi.
4. Kepuasan pemakai. Hal yang paling penting adalah pemakai mengetahui dan puas terhadap sistem informasi.

Sistem informasi mempunyai enam buah komponen atau disebut juga dengan blok bangunan (*building block*), yaitu:

1. Komponen *input* atau komponen masukan.
2. Komponen model.
3. Komponen *output* atau komponen keluaran.
4. Komponen teknologi.
5. Komponen basis data.
6. Komponen kontrol atau komponen pengendalian.

(Husda, 2012, p. 121) mengatakan, keenam komponen ini harus ada bersama-sama dan membentuk satu kesatuan. Jika satu atau lebih komponen tersebut tidak ada, maka sistem informasi tidak akan dapat melakukan fungsinya,

yaitu pengolahan data dan tidak dapat mencapai tujuannya, yaitu menghasilkan informasi yang relevan, tepat waktu dan akurat. Komponen–komponen dari sistem informasi ini dapat digambarkan sebagai berikut:

1. Blok Masukkan (*Input Block*)

Input merupakan data yang masuk ke dalam sistem informasi.

2. Blok Model (*Model Block*)

Kombinasi prosedur, logika, dan model matematis yang akan memanipulasi data *input* dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran (*Output Block*)

Keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi (*Technology Block*)

Teknologi merupakan kotak alat (*tool box*) dalam sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara menyeluruh.

5. Blok Basis Data (*Database Block*)

Merupakan kumpulan dari data yang saling berhubungan satu sama lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak memanipulasinya.

6. Blok Kendali (*Control Block*)

Beberapa pengendalian yang dirancang secara khusus untuk menanggulangi gangguan–gangguan terhadap sistem.

Cara pandang pengklasifikasian sistem informasi adalah:

1. Level Organisasi

Sistem informasi menurut level organisasi:

a. Sistem informasi departemen

Contoh: Sistem Informasi SDM (*HRIS*)

b. Sistem informasi perusahaan (*enterprise information system*)

Contoh: Sistem Informasi Perguruan Tinggi

c. Sistem informasi antar organisasi

Contoh: e-Commerce

2. Sistem Informasi Fungsional

Sistem informasi berdasarkan area fungsional ditujukan untuk memberikan informasi bagi kelompok orang yang berada pada bagian tertentu dalam perusahaan.

Contoh:

a. Sistem informasi akuntansi

Sistem informasi yang menyediakan informasi yang dipakai oleh fungsi akuntansi (departemen / bagian akuntansi).

b. Sistem informasi keuangan

c. Sistem informasi yang menyediakan informasi pada fungsi keuangan yang menyangkut keuangan perusahaan.

Contoh: *Cash Flow* dan informasi pembayaran

3. Klasifikasi Sistem Informasi Berdasarkan Fungsi

a. Sistem Pemrosesan Transaksi (*Transaction Processing System/TPS*)

Sebuah sistem yang meng – *capture* dan memproses data transaksi bisnis.

Misalnya: pesanan, kartu absensi, pembayaran, KRS, reservasi, dll.

b. Sistem Informasi Manajemen (*Management Information System/MIS*)

Sistem informasi yang menyediakan pelaporan yang berorientasi manajemen berdasarkan pemrosesan transaksi dan operasi organisasi.

c. Sistem Pendukung Keputusan (*Decision Support System/DSS*)

Sistem informasi yang menidentifikasi berbagai alternatif keputusan atau menyediakan informasi untuk membantu pembuatan keputusan.

d. Sistem Informasi Eksekutif (*Executive Information System/EIS*)

Sistem informasi yang diperuntukkan oleh manajer eksekutif untuk mendukung perencanaan bisnis dan menilai performa rencana tersebut.

e. Sistem Pakar (*Expert System*)

Sistem informasi yang meng–*capture* dan menghasilkan kembali pengetahuan asli pemecahan masalah atau para pengambil keputusan dan mensimulasikan kembali “pemikiran” ahli tersebut.

f. Sistem Komunikasi dan Kolaborasi (*Communication and Collaboration System*)

Sistem yang memungkinkan komunikasi lebih efektif antara orang–orang dalam maupun luar organisasi untuk meningkatkan kemampuan berkolaborasi.

g. Sistem Otomasi Kantor (*Office Automation System*)

Sistem informasi yang mendukung aktifitas bisnis kantor secara luas yang menyediakan aliran kerja yang diperbaiki antar personil.

4. Sistem Informasi Berdasarkan Dukungan Yang Tersedia

a. Sistem Pemrosesan Transaksi (*Transaction Processing System* atau *TPS*)

b. Sistem Informasi Manajemen (*Management Information System* atau *MIS*)

c. Sistem Otomasi Perkantoran (*Office Automation System* atau *OAS*)

d. Sistem Pendukung Keputusan (*Decision Support System* atau *DSS*)

e. Sistem Informasi Eksekutif (*Executive Information System* atau *EIS*)

f. Sistem Pendukung Kelompok (*Group Support System* atau *GSS*)

g. Sistem Pendukung Cerdas (*Intelligent Support System* atau *ISS*)

2.1.2. Software Development Life Cycle (SDLC)

Menurut (A.S, Rosa & Salahuddin, 2013, p. 24), *SDLC* atau *Software Development Life Cycle* atau sering disebut juga *System Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model–model dan metodologi yang digunakan orang untuk mengembangkan sistem–sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara–cara sudah teruji dengan baik).

Tahapan–tahapan yang ada pada *SDLC* secara global adalah sebagai berikut:

1. Inisiasi (*initiation*)

Tahap ini biasanya ditandai dengan pembuatan proposal proyek perangkat lunak.

2. Pengembangan konsep sistem (*system concept development*)

Mendefinisikan lingkup konsep termasuk dokumen lingkup sistem, analisis manfaat biaya, manajemen rencana, dan pembelajaran kemudahan sistem.

3. Perencanaan (*Planning*)

Mengembangkan rencana manajemen proyek dan dokumen perencanaan lainnya. Menyediakan dasar untuk mendapatkan sumber daya (*resources*) yang dibutuhkan untuk memperoleh solusi.

4. Analisis kebutuhan (*requirements analysis*)

Menganalisis kebutuhan pemakai sistem perangkat lunak (*user*) dan mengembangkan kebutuhan *user*. Membuat dokumen kebutuhan fungsional.

5. Desain (*design*)

Mentransformasikan kebutuhan detail menjadi kebutuhan yang sudah lengkap, dokumen desain sistem fokus pada bagaimana dapat memenuhi fungsi–fungsi yang dibutuhkan.

6. Pengembangan (*development*)

Mengkonversi desain ke sistem informasi yang lengkap termasuk bagaimana memperoleh dan melakukan instalasi lingkungan sistem yang dibutuhkan, membuat basis data dan mempersiapkan prosedur kasus pengujian, mempersiapkan berkas atau *file* pengujian, pengodean,

pengompilasian, memperbaiki dan membersihkan program; peninjauan pengujian.

7. Integrasi dan pengujian (*integration and test*)

Mendemonstrasikan sistem perangkat lunak bahwa telah memenuhi kebutuhan yang dispesifikasikan pada dokumen kebutuhan fungsional. Dengan diarahkan oleh staf penjamin kualitas (*quality assurance*) dan *user*. Menghasilkan laporan analisis penjualan.

8. Implementasi (*implementation*)

Termasuk pada persiapan implementasi, implementasi perangkat lunak pada lingkungan produksi (lingkungan pada *user*) dan menjalankan resolusi dari permasalahan yang teridentifikasi dari fase integrasi dan pengujian.

9. Operasi dan pemeliharaan (*operations and maintenance*)

Mendeskripsikan pekerjaan untuk mengoperasikan dan memelihara sistem informasi pada lingkungan produksi (lingkungan pada *user*), termasuk implementasi akhir dan masuk pada proses peninjauan.

10. Disposisi (*disposition*)

Mendeskripsikan aktifitas akhir dari pengembangan sistem dan membangun data yang sebenarnya sesuai dengan aktifitas *user*.

SDLC memiliki beberapa model dalam penerapan tahapan prosesnya, yaitu:

1. Model *Waterfall*

Model *SDLC* air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara

sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian dan tahap pendukung (*support*).

2. Model Prototipe

Model prototipe dapat digunakan untuk menyambungkan ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak.

3. Model *Rapid Application Development (RAD)*

Rapid Application Development (RAD) adalah model proses pengembangan perangkat lunak yang bersifat inkremental terutama untuk waktu pengerjaan yang pendek. Model *RAD* adalah adaptasi dari model air terjun versi kecepatan tinggi dengan menggunakan model air terjun untuk pengembangan setiap komponen perangkat lunak.

4. Model Iteratif

Model iteratif (*iterative model*) mengombinasikan proses–proses pada model air terjun dan iteratif pada model prototipe. Model inkremental akan menghasilkan versi–versi perangkat lunak yang sudah mengalami penambahan fungsi untuk setiap pertambahannya (*inkremen/increment*).

5. Model Spiral

Model Spiral (*spiral model*) memasangkan iteratif pada model prototipe dengan kontrol dan aspek sistematis yang diambil dari model air terjun. Model spiral menyediakan pengembangan dengan cara cepat dengan perangkat lunak yang memiliki versi yang terus bertambah fungsinya (*increment*).

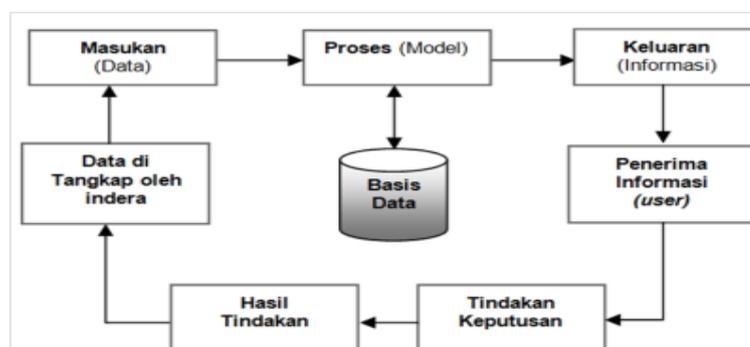
2.1.3. Basis Data

Menurut (A.S, Rosa & Salahuddin, 2013, p. 44), Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.

(Husda, 2012, p. 152) mengatakan basis data sendiri dapat disimpulkan sebagai:

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan, untuk memenuhi berbagai kebutuhan.
3. Kumpulan *file / table / arsip* yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Berikut adalah contoh ilustrasi dari basis data:



Gambar 2. 1 Ilustrasi Basis Data

Sebuah data pada basis data mempunyai beberapa tingkatan diantaranya adalah:

1. *Characters* (karakter)

Karakter merupakan bagian data yang paling kecil, dapat berupa angka, huruf atau karakter khusus yang membentuk suatu item data atau *field*.

Contoh: 1, 2, 3, A, B, c, d, =, >, #, dan sebagainya

2. *Field* (Data Item)

Field merupakan representasi suatu atribut dari *record* yang sejenis yang menunjukkan suatu item dari data.

Contoh: Nama, alamat, tanggal, dan sebagainya

3. *Record*

Record adalah kumpulan elemen yang saling berkaitan menginformasikan tentang satu entitas secara lengkap. Satu *record* mewakili satu data atau informasi.

Contoh: NPM, Nama, alamat, tanggal_lahir

4. *File*

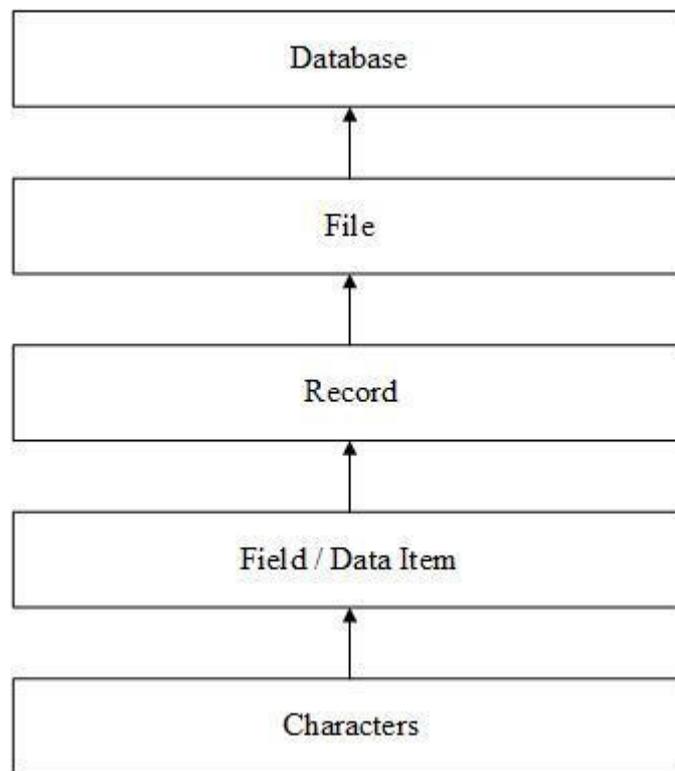
File merupakan kumpulan dari record-record dalam basis yang menggambarkan satu kesatuan data yang sejenis.

Contoh: *file* mata_kuliah yang berisi data tentang semua mata kuliah yang ada.

5. *Database* (Basis Data)

Database merupakan kumpulan dari beberapa file atau tabel yang saling berhubungan sehingga membentuk satu basis data.

Contoh: Basis data perpustakaan, Basis data penjualan, sebagainya



Gambar 2. 2 Tingkatan Basis Data

2.1.4. Sistem Pendukung Keputusan (*Decision Support System*)

Sistem Pendukung Keputusan (*Decision Support System*) merupakan sistem informasi interaktif yang menyediakan informasi pemodelan, dan pemanipulasian data. Sistem ini digunakan untuk membantu pengambilan keputusan dalam situasi yang semiterstruktur dan situasi yang tidak terstruktur, di mana tak seorangpun tahu secara pasti bagaimana keputusan seharusnya dibuat (Rudiansyah, 2014, p. 57).

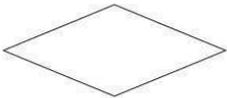
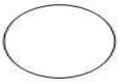
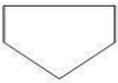
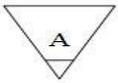
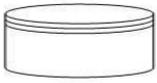
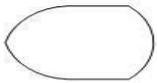
Decision Support System biasanya dibangun untuk mendukung keputusan solusi atas suatu masalah atau untuk mengevaluasi suatu peluang. *Decision Support System* yang seperti itu disebut aplikasi *Decision Support System*.

Aplikasi *Decision Support System* menggunakan CBIS (*Computer Based Information Systems*) yang fleksibel, interaktif, dan dapat diadaptasi, yang dikembangkan untuk mendukung solusi atas masalah manajemen spesifik yang tidak terstruktur. Aplikasi *Decision Support System* menggunakan data, memberikan antarmuka pengguna yang mudah, dan dapat menggabungkan pemikiran pengambilan keputusan (Rudiansyah, 2014, pp. 57–58).

Sistem pendukung keputusan semakin penting di banyak bidang sains dan teknologi dan digunakan secara efektif, terutama bila masalah yang harus diselesaikan rumit dengan banyak kriteria. Dalam masalah seperti ini, salah satu tantangan utama bagi para pengambil keputusan adalah kadang-kadang mereka tidak dapat menghasilkan data yang dapat dihitung untuk mengevaluasi kriteria tetapi pengetahuan dan rasa para ahli (Yanar, 2015, p. 317).

2.1.5. Aliran Sistem Informasi

Menurut (Bidang et al., 2017, p. 149), Aliran Sistem Informasi (ASI) merupakan bagan yang menunjukkan arus pekerjaan serta keseluruhan dari sistem. Bagan ini menjelaskan urutan dari prosedur-prosedur yang ada dalam sistem. Adapun simbol-simbol yang dapat digunakan pada Aliran Sistem Informasi (ASI) ini adalah sebagai berikut:

Daftar Simbol		
Simbol	Pengertian	Keterangan
	<i>Terminator</i>	Menyatakan permulaan atau akhir suatu program
	Dokumen	Digunakan untuk merekam data terjadinya suatu transaksi
	Proses Manual	Menunjukkan proses yang dikerjakan secara manual
	Proses Komputer	Menunjukkan proses yang dari operasi program komputer
	Garis Alir	Menunjukkan arus data antar simbol/proses
	<i>Decision</i>	Menunjukkan pilihan yang akan dikerjakan atau keputusan yang harus dibuat dalam proses pengolahan data
	<i>Penghubung (On-page reference)</i>	Digunakan untuk penghubung dalam satu halaman
	<i>Penghubung (Off-page reference)</i>	Digunakan untuk penghubung berbeda halaman
	<i>Offline-line Storage</i>	Digunakan untuk menyimpan data secara manual dan sementara jika "A" berarti disimpan menurut abjad, "N" berarti disimpan menurut Nomor urut dan "T" berarti disimpan menurut kronologis atau menurut tanggal
	<i>Database</i>	Tempat penyimpanan data dari proses simpan yang dilakukan
	<i>Input/Output</i>	Memasukkan data/menunjukkan hasil dari suatu proses
	<i>Display</i>	Menunjukkan output yang ditampilkan ke monitor

Gambar 2. 3 Simbol Aliran Sistem Informasi

2.1.6. Unified Modeling Language (UML)

Menurut (A.S, Rosa & Salahuddin, 2013, p. 118), *Unified Modeling Language* (UML), adalah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks–teks pendukung.

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.

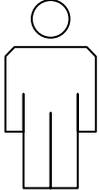
Ada empat diagram UML yang digunakan oleh peneliti dan akan dijelaskan menurut sebagai berikut:

1. *Use case Diagram*

Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behaviour*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi–fungsi itu.

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa saja yang disebut aktor dan *use case*.

Tabel 2. 1 Simbol *Use Case*

Simbol	Deskripsi
<p data-bbox="316 667 432 696"><i>Use case</i></p>  <p data-bbox="469 770 671 799">Nama Use Case</p>	<p data-bbox="849 667 1370 994">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>.</p>
<p data-bbox="316 1037 472 1066"><i>Aktor/actor</i></p>  <p data-bbox="491 1312 647 1341">Nama Aktor</p>	<p data-bbox="849 1037 1370 1659">Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p data-bbox="316 1697 584 1727"><i>Asosiasi/association</i></p> 	<p data-bbox="849 1697 1370 1951">Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor</p>

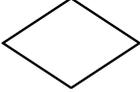
Lanjutan **Tabel 2. 1** Simbol *Use Case*

<p>Ekstensi/<i>extend</i></p> <p style="text-align: center;">-- <<extend>> -></p>	<p>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 pemograman berorientasi objek, biasanya <i>use case</i> tambahan.</p>
<p>Generalisasi/<i>generalization</i></p> <p style="text-align: center;">—————></p>	<p>Hubungan generalisasi dan spesialisasi (umum–khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>
<p>Menggunakan/<i>include/uses</i></p> <p style="text-align: center;">-- <<include>> -> —————></p>	<p>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.</p>

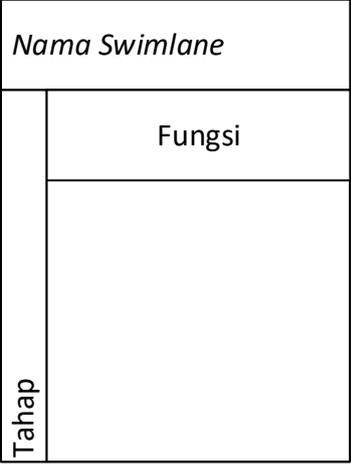
2. *Activity Diagram*

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Tabel 2. 2 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 / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

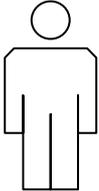
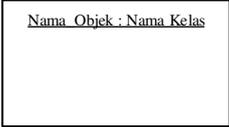
Lanjutan **Tabel 2. 2** Simbol *Activity Diagram*

<p><i>Swimlane</i></p> 	<p>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.</p>
--	---

3. *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendiskripsikan waktu hid up objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diintansiasi menjadi objek itu.

Tabel 2. 3 Simbol *Sequence Diagram*

Simbol	Deskripsi
<p data-bbox="316 450 395 479">Aktor</p>  <p data-bbox="483 725 655 754">Nama_Aktor</p>	<p data-bbox="847 450 1369 1066">Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p data-bbox="316 1111 568 1140">Garis hidup/<i>lifeline</i></p> 	<p data-bbox="847 1111 1321 1140">Menyatakan kehidupan suatu obyek.</p>
<p data-bbox="316 1581 395 1610">Objek</p> 	<p data-bbox="847 1581 1369 1686">Menyatakan objek yang berinteraksi pesan.</p>

Lanjutan **Tabel 2. 3** Simbol *Sequence Diagram*

<p>Waktu Aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.</p>
<p>Pesan tipe <i>Create</i></p> <p><<create>></p> 	<p>Menyatakan suatu obyek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>
<p>Pesan tipe <i>call</i></p> <p>1 : nama_metode()</p> 	<p>Menyatakan suatu obyek memanggil operasi/metode yang ada pada obyek lain atau dirinya sendiri.</p>
<p>Pesan tipe <i>send</i></p> 	<p>Menyatakan bahwa suatu obyek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>
<p>Pesan tipe <i>return</i></p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada obyek yang menerima kembalian.</p>

Lanjutan **Tabel 2. 3** Simbol *Sequence Diagram*

<p>Pesan tipe <i>destroy</i></p> <p style="text-align: center;"> <<destroy>> —————▶ </p>	<p>Menyatakan suatu obyek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>.</p>
---	---

4. *Class Diagram*

Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

Atribut merupakan variabel–variabel yang dimiliki oleh suatu kelas, sedangkan operasi atau metode adalah fungsi–fungsi yang dimiliki oleh suatu kelas.

Kelas–kelas yang ada pada struktur sistem harus dapat melakukan fungsi–fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis–jenis kelas berikut:

a. Kelas main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

b. Kelas yang menangani tampilan sistem

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

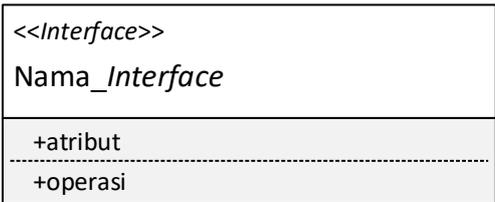
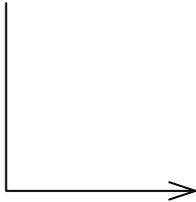
c. Kelas yang diambil dari pendefinisian *use case*

Kelas yang menangani fungsi–fungsi yang harus ada diambil dari pendefinisian *use case*.

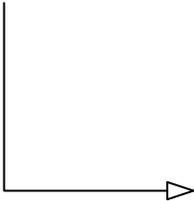
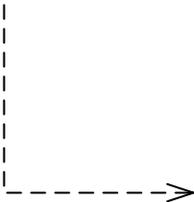
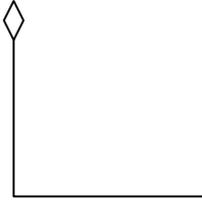
d. Kelas yang diambil dari pendefinisian data

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Tabel 2. 4 Simbol *Class Diagram*

<p>Kelas</p> 	<p>Kelas pada struktur system</p>
<p>Antarmuka/<i>interface</i></p> 	<p>Sama dengan konsep interface dalam pemograman berorientasi objek.</p>
<p>Asosiasi/<i>association</i></p> 	<p>Relasi antara kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity.</p>
<p>Asosiasi berarah/<i>directed association</i></p> 	<p>Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity.</p>

Lanjutan **Tabel 2. 4** Simbol *Class Diagram*

<p>Generalisasi</p> 	<p>Relasi antar kelas dengan makna generalisasi–spesialisasi (umum khusus).</p>
<p>Kebergantungan/<i>dependency</i></p> 	<p>Relasi antar kelas dengan makna kebergantungan antar kelas</p>
<p>Agregasi/<i>aggregation</i></p> 	<p>Relasi antar kelas dengan makna semua bagian (<i>whole–part</i>).</p>

2.2. Tinjauan Teori Khusus

Dalam bagian ini, penulis akan menjelaskan mengenai teori–teori khusus menurut pendapat ahli.

2.2.1. Kolesterol

Menurut (Zerbinati & Iuliano, 2017, p. 151), Kolesterol adalah blok bangunan utama membran plasma, terhitung 50% lipid pada basis molar dan sangat penting untuk menjaga integritas membran dan organisasi. Kolesterol memiliki peran penting dalam sintesis hormon steroid, asam empedu, dan vitamin D. Efek biologis kolesterol tambahan terkait dengan produk oksigen yang terkait, yaitu Oksisterol. Oksisterol telah disarankan untuk berperan dalam berbagai pengaturan klinis, termasuk aterosklerosis, penyakit neurodegeneratif, sistem kekebalan tubuh, dan kanker. Oksisterol diproduksi melalui mekanisme enzimatik atau non-enzimatik.

Menurut (Odabaşı et al., 2017), Kolesterol adalah komponen utama membran plasma dan molekul kunci yang terlibat dalam transportasi intraseluler, peraturan aktivitas protein, pensinyalan sel, dan konduksi saraf. Ini adalah prekursor sintesis hormon steroid dan juga bermanfaat bagi metabolisme asam empedu dan vitamin D. Konsentrasi kolesterol tinggi dikaitkan dengan hipertensi, penyakit jantung koroner, dan infark miokard. Selain itu, mereka dapat menyebabkan pembentukan plak arteri aterosklerotik di arteri koroner, yang mengakibatkan stroke atau bahkan kematian. *Low-density lipoprotein* (LDL) adalah salah satu kolesterol utama yang membawa lipoprotein dalam plasma. Ini dikeluarkan dari sirkulasi darah oleh reseptor LDL hepatic (LDLR). Berkurangnya fungsi atau ketiadaan reseptor LDL adalah penyebab utama familial hiperkolesterolemia, di mana kadar LDL plasma meningkat.

(College, 2013, pp. 585–586) Hiperkolesterolemia merupakan kondisi saat konsentrasi kolesterol di dalam darah melebihi batas normal. Hiperkolesterolemia terjadi akibat akumulasi kolesterol dan lipid pada dinding pembuluh darah. Kolesterol LDL-teroksidasi berperan dalam pembentukan plak aterosklerosis atau penyempitan pembuluh darah.

(College, 2013, p. 586) Salah satu alternatif yang aman menurunkan kadar kolesterol LDL yaitu modifikasi pola diet. Diet yang dianjurkan adalah membatasi konsumsi makanan yang mengandung kolesterol dengan mengonsumsi makanan yang bersifat antihiperkolesterolemia. Salah satu bahan makanan yang bersifat antihiperkolesterolemia adalah kacang hijau atau *Phaseolus radiatus*.

(College, 2013, p. 586) Kacang hijau merupakan tumbuhan kacang-kacangan yang mudah dijumpai di Indonesia. Kacang hijau memiliki kandungan karbohidrat dan serat yang baik. Karbohidrat merupakan komponen terbesar dari kacang hijau yaitu sebesar 62-63%. Kandungan lemak pada kacang hijau adalah 0,7-1 gr/kg kacang hijau segar yang terdiri atas 73% lemak tak jenuh dan 27% lemak jenuh, sehingga aman dikonsumsi oleh orang yang mengalami hiperkolesterolemia.

2.2.2. Analytic Hierarchy Process (AHP)

AHP adalah sebuah konsep untuk pembuatan keputusan berbasis *multicriteria* (kriteria yang banyak). Beberapa kriteria yang dibandingkan satu dengan lainnya (tingkat kepentingannya) adalah penekanan utama pada konsep *AHP* ini. *AHP* menjadi sebuah konsep penentuan/pembuatan keputusan, yang

menggabungkan prinsip-prinsip subjektifitas dan objektifitas si pembuat SPK atau keputusannya (Utama, 2017, p. 114).

Analytic Hierarchy Process (AHP) adalah salah satu metode *Multi Criteria Decision Making (MCDM)* yang paling banyak digunakan yang diperkenalkan pada literatur oleh Thomas SAATY pada tahun 1970an (Saaty & Wind, 1, 1980). *AHP* adalah metode yang menggunakan nilai eigen untuk mendapatkan prioritas kriteria dan alternatif dari matriks berpasangan yang membandingkan unsur hirarkis yang dianalisis. Definisi masalah dan hirarki kriteria elemen yang digunakan dalam matriks berpasangan harus dilakukan oleh para ahli di bidang itu untuk mencapai hasil yang konsisten dan masuk akal. Proses pertama menentukan prioritas kriteria dan kemudian tingkat masing-masing alternatif untuk memenuhi setiap kriteria. Setidaknya hasilnya akan tergantung pada sintesis hasil ini (Yanar, 2015, p. 318).

Pada dasarnya, proses pengambilan keputusan adalah memilih suatu alternatif. Peralatan utama *AHP* adalah sebuah hierarki fungsional dengan input utamanya persepsi manusia. Keberadaan hierarki memungkinkan dipecahnya masalah kompleks atau tidak terstruktur dalam sub-sub masalah, lalu menyusunnya menjadi suatu bentuk hierarki (Rudiansyah, 2014, p. 58).

Metode *AHP (Analytical Hierarchy Process)* memiliki banyak keunggulan dalam menjelaskan proses pengambilan keputusan. Salah satunya adalah dapat digambarkan secara grafis sehingga mudah dipahami oleh semua pihak yang terlibat dalam pengambilan keputusan (Rudiansyah, 2014, p. 58).

Dalam menyelesaikan permasalahan dengan AHP ada beberapa prinsip yang harus dipahami, diantaranya adalah (Am, N, & K., 2015, p. 51):

1. Membuat hierarki

Sistem yang kompleks bisa dipahami dengan memecahnya menjadi elemen-elemen pendukung, menyusun elemen secara hierarki dan menggabungkannya atau mensintesisnya.

2. Penilaian kriteria dan alternatif

Kriteria dan alternatif dilakukan dengan perbandingan berpasangan. Menurut Saaty, untuk berbagai persoalan, skala 1 sampai 9 adalah skala terbaik untuk mengekspresikan pendapat. Nilai dan definisi pendapat kualitatif dari skala perbandingan Saaty bisa diukur menggunakan tabel analisis berikut.

Tabel 2. 5 Skala Penilaian Perbandingan Pasangan (Am et al., 2015, p. 49)

Intensitas Kepentingan	Keterangan
1	Kedua elemen sama pentingnya
3	Elemen yang satu sedikit lebih penting
5	Elemen yang satu lebih penting daripada elemen lainnya
7	Satu elemen jelas lebih mutlak penting daripada elemen lainnya
9	Satu elemen mutlak penting daripada elemen lainnya
2, 4, 6, 8	Nilai-nilai antara dua nilai pertimbangan yang berdekatan
Kebalikan	Jika aktivitas <i>i</i> mendapat satu angka dibandingkan dengan aktivitas <i>j</i> , maka <i>i</i> memiliki nilai kebalikannya dibandingkan dengan <i>i</i>

3. *Synthesis of priority* (menentukan prioritas)

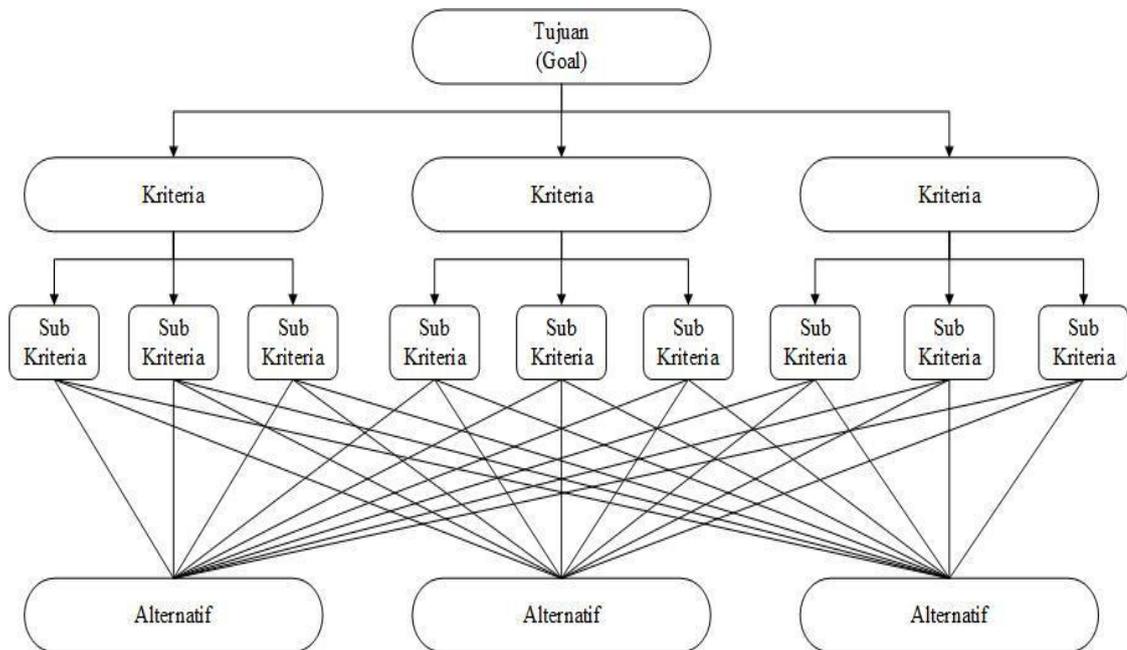
Untuk setiap kriteria dan alternatif, perlu dilakukan perbandingan berpasangan (*pairwise Comparisons*). Nilai-nilai perbandingan relatif dari seluruh alternatif kriteria bisa disesuaikan dengan *judgement* yang telah ditentukan untuk menghasilkan bobot dan prioritas. Bobot dan prioritas dihitung dengan memanipulasi 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.

Selain itu AHP juga memiliki beberapa prosedur. Prosedur atau langkah-langkah dalam metode AHP meliputi (Am et al., 2015, p. 51):

1. Mendefinisikan masalah dan menentukan solusi yang diinginkan, lalu menyusun hierarki dari permasalahan yang dihadapi. Penyusunan hierarki adalah dengan menetapkan tujuan yang merupakan sasaran sistem secara keseluruhan pada level teratas.



Gambar 2. 4 Model Hirarki yang digunakan dalam AHP (Yanar, 2015, p. 319).

2. Menentukan prioritas elemen

- a. Langkah pertama dalam menentukan prioritas elemen adalah membuat perbandingan pasangan, yaitu membandingkan elemen secara berpasangan sesuai kriteria yang diberikan.
- b. Matriks perbandingan berpasangan diisi menggunakan bilangan untuk merepresentasikan kepentingan relatif dari suatu elemen terhadap elemen yang lainnya.

3. Sintesis

Pertimbangan-pertimbangan terhadap perbandingan berpasangan disintesis untuk memperoleh keseluruhan prioritas. Hal-hal yang dilakukan dalam langkah ini adalah:

- a. Menjumlahkan nilai-nilai dari setiap kolom pada matriks

- b. Membagi setiap nilai dari kolom dengan total kolom yang bersangkutan untuk memperoleh normalisasi matriks
- c. Menjumlahkan nilai-nilai dari setiap baris dan membaginya dengan jumlah elemen untuk mendapatkan nilai rata-rata

4. Mengukur Konsistensi

Dalam pembuatan keputusan, penting untuk mengetahui seberapa baik konsistensi yang ada karena kita tidak menginginkan keputusan berdasarkan pertimbangan dengan konsistensi yang rendah. Hal-hal yang dilakukan dalam langkah ini adalah:

- a. Kalikan setiap nilai pada kolom pertama dengan prioritas relatif elemen pertama, nilai pada kolom kedua dengan prioritas relatif elemen kedua, dan seterusnya
- b. Jumlahkan setiap baris
- c. Hasil dari penjumlahan baris dibagi dengan elemen prioritas relatif yang bersangkutan
- d. Jumlahkan hasil bagi diatas dengan banyaknya elemen yang ada, hasilnya disebut λ maks

5. Hitung *Consistency Index* (CI) dengan rumus:

$$CI = (\lambda \text{ maks} - n) / n$$

dimana n = banyaknya elemen

6. Hitung Rasio Konsistensi/*Consistency Ratio* (CR) dengan rumus:

$$CR = CI / RC$$

dimana CR = *Consistency Ratio*

CI = *Consistency Index*

IR = *Index Random Consistency*

7. Memeriksa konsistensi hierarki. Jika nilainya lebih dari 10%, maka penilaian data *judgment* harus diperbaiki. Namun jika rasio konsistensi (CI/IR) kurang atau sama dengan 0,1 maka hasil perhitungan bisa dinyatakan benar. Daftar Indeks Random Konsistensi (IR) dapat dilihat pada Tabel 2.6.

Tabel 2. 6 Daftar Indeks Random Konsistensi (Am et al., 2015, p. 52)

Ukuran Matriks (n)	Nilai IR
1, 2	0.00
3	0.58
4	0.90
5	1.12
6	1.24
7	1.32
8	1.41
9	1.45
10	1.49
11	1.51
12	1.48
13	1.56
14	1.57
15	1.59

2.2.3. Software Yang Digunakan

Berikut adalah beberapa bahasa pemrograman untuk pembuatan aplikasi *web* yang penulis gunakan.

1. *HTML 5*

Menurut (Saputra, 2012, p. 3), HTML merupakan singkatan dari *Hyper Text Markup Language*. HTML bisa disebut bahasa paling dasar dan penting yang digunakan untuk menampilkan dan mengelola tampilan pada halaman *website*. Menurut Wikipedia yang dikutip dari (Saputra, 2012, p. 3), HTML digunakan untuk menampilkan berbagai informasi didalam sebuah penjelajah *web Internet* dan *formatting hypertext* sederhana yang ditulis ke dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata, disimpan ke dalam format ASCII normal sehingga menjadi *homepage* dengan perintah-perintah HTML. HTML menggunakan dua macam ekstensi file yaitu *.htm* dan *.html*. Format ekstensi berformat *.htm* awalnya hanyalah untuk mengakomodasi penggunaan html dalam operasi DOS.

Berikut adalah fitur-fitur terbaru dalam HTML 5 yang wajib diketahui:

1. Unsur canvas untuk gambar.
2. Bentuk kontrol form seperti kalender, tanggal, waktu, *email*, url, dan *search*.
3. Elemen konten yang lebih spesifik, seperti artikel, *footer*, *header*, navigasi, dan *section*.
4. Dukungan yang lebih baik untuk penyimpanan secara *offline*.
5. Dan tentunya juga dukungan untuk pemutaran video dan audio.

Berikut adalah elemen-elemen baru dalam HTML 5:

1. *Section*, serupa H1 – H6.

2. *Article*, bisa berupa entri blog atau tulisan konten.
3. *Aside*, menyajikan konten pelengkap.
4. *Header*, digunakan untuk judul, deskripsi, bahkan nav untuk navigasi.
5. *Footer*, digunakan untuk bagian bawah *web* yang digunakan untuk menerangkan informasi *copyright* (hak cipta), perusahaan, nama pembuat, kontak, dan sebagainya.
6. *Dialog*, yang dikombinasikan dengan *dt* dan *dd*, digunakan untuk menyajikan percakapan.
7. Dan yang paling mencolok dalam HTML 5 adalah adanya penggunaanelemen *figure*, video, audio, *source*, *embed*, *canvas*, dan elemen-elemen lainnya yang berkaitan dengan multimedia.

2. XAMPP

Menurut (Ngige W., 2015a, p. 4), XAMPP (Sebuah *cross-platform Apache Mysql Php Processor*) adalah paket *server web cross-platform open source* termasuk basis data dan penerjemah *Apache HTTP Server*, *MySQL (My Structure Query Language)* dan *script* untuk *PHP* yang ditulis dalam *PHP (PHP: Hypertext Processor)* dan bahasa pemrograman *Perl*. Ini digunakan sebagai lingkungan pengembangan yang memungkinkan portal *web* diuji di komputer tanpa akses ke *internet*. Program ini digunakan karena kontennya, ukurannya kecil dan portabilitas.

3. PHP

Menurut (Ngige W., 2015b, p. 4), PHP (*PHP: Hypertext Preprocessor*) adalah bahasa *script sisi server* yang dirancang untuk pengembangan *web*

namun juga digunakan sebagai bahasa pemrograman tujuan umum. PHP digunakan karena fitur pemrosesan teksnya yang memungkinkan keluaran HTML, gambar, *file* PDF dan bahkan teks seperti XHTML dan XML. Ini juga bekerja dengan baik dengan database *MySQL* dan dapat digunakan pada sebagian besar *server web* dan sistem operasi.

4. CSS 3

Menurut (Saputra, 2012, p. 27), CSS yang merupakan singkatan dari *Cascading Style Sheet* merupakan bahasa pemrograman *web* yang didesain khusus untuk mengendalikan dan membangun berbagai komponen dalam *web* sehingga tampilan *web* lebih rapi, terstruktur, dan seragam. CSS merupakan salah satu pemrogram wajib disamping html yang harus dikuasai oleh para setiap pemrogram *web*, terlebih lagi itu adalah *Web Designer*.

Tujuan utama dari CSS adalah untuk memisahkan konten utama dengan tampilan dokumen lainnya (html dan sejenisnya). Dengan adanya pemisahan ini, akses konten pada *web* meningkat. Web yang menggunakan CSS akan lebih ringan dan mudah untuk dibuka dibandingkan dengan *web* yang tidak menggunakan CSS. Perbedaan ini akan semakin terasa ketika *web* yang Anda buka mempunyai data yang banyak.

Tujuan lainnya adalah untuk mempercepat pembuatan halaman *web*. Anda hanya perlu membuat satu properti dan properti tersebut dapat digunakan pada elemen lainnya, artinya, Anda tidak perlu menulis ulang kode program yang digunakan berulang kali. Berikut fitur-fitur baru yang dapat kita gunakan:

1. *Border*
 2. *Backgrounds*
 3. *Color*
 4. *Text Effects*
 5. *User Interface*
 6. *Selector, attribute selectors*
 7. *Basis Box Model, Overflow-x, Overflow-y*
 8. *Generated Content, Content*
 9. *Other Modules*
 10. *Web Fonts, Speech*
5. *JavaScript*

Menurut (Yatini, 2014, p. 2) *JavaScript* adalah bahasa *scripting* kecil, ringan, berorientasi objek yang ditempelkan pada kode HTML dan di proses di sisi *client*. *JavaScript* digunakan dalam pembuatan *website* agar lebih interaktif dengan memberikan kemampuan tambahan terhadap HTML melalui eksekusi perintah di sisi *browser*. *JavaScript* dapat merespon perintah *user* dengan cepat dan menjadikan halaman *web* menjadi responsif. *JavaScript* memiliki struktur sederhana, kodenya dapat disisipkan pada dokumen HTML atau berdiri sebagai satu kesatuan aplikasi. Struktur penulisan *JavaScript* adalah sebagai berikut.

```
<script language = "javascript">
```

```
<!--Penulisan kode javascript!-->
```

```
</script>
```

6. *MySQL*

Menurut (Saputra, 2012, p. 77), *MySQL* merupakan salah satu *database* kelas dunia yang sangat cocok bila dipadukan dengan bahasa pemrograman PHP. *MySQL* bekerja menggunakan bahasa SQL (*Structure Query Language*) yang merupakan bahasa standar yang digunakan untuk manipulasi *database*.

Pada umumnya, perintah yang paling sering digunakan dalam *MySQL* adalah *SELECT* (mengambil), *INSERT* (menambah), *UPDATE* (mengubah), dan *DELETE* (menghapus). Selain itu, SQL juga menyediakan perintah untuk membuat *database*, *field*, ataupun *index* untuk menambah atau menghapus data. Ada beberapa alasan yang menjadikan *database MySQL* sangat diminati oleh para *programmer*, diantaranya:

1. Bersifat *open source*.
2. Menggunakan bahasa SQL (*Structure Query Language*), yang merupakan standar bahasa dalam pengolahan data.
3. *Performance* dan *reliable*, pemrosesan *database*-nya sangat cepat dan stabil.
4. Sangat mudah dipelajari (*ease of use*).
5. Memiliki dukungan (*group*) pengguna *MySQL*.
6. Lintas *Platform*, dapat digunakan pada berbagai Sistem Operasi berbeda.
7. *Multiuser*, dimana *MySQL* dapat digunakan oleh banyak *user* dalam waktu yang bersamaan tanpa mengalami konflik.

8. Dan masih banyak lagi.

2.3. Peneliti Terdahulu

Pada tinjauan penelitian terdahulu akan dibahas secara lengkap jurnal dan artikel yang mendukung sebagai dasar pembahasan interpretasi penelitian pada bahan sebelumnya.

1. Marsono, Ahmad Fitri Boy. (2015). Sistem Pendukung Keputusan Pemilihan Menu Makanan pada Penderita Obesitas dengan menggunakan Metode Topsis. *Jurnal SAINTIKOM Vol.14, No. 3*. ISSN: 1978-6603. Metode TOPSIS dapat digunakan dalam menentukan pemilihan menu makanan pada penderita obesitas secara tepat dan baik.
2. Rudiansyah. (2014). SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN MAKANAN PADA BAYI LIMA TAHUN (BALITA) DENGAN MENGGUNAKAN METODE ANALYTICAL HIERARCHY PROCESS (AHP). *Pelita Informatika Budi Darma, Volume : VII, Nomor: 3*. ISSN: 2301-9425. Proses penyeleksian makanan pada bayi lima tahun (Balita) sangat tergantung pada kelengkapan data-data kriteria yang di *inputkan* dengan menggunakan metode *Analytical Hierarchy Process* (AHP) membutuhkan proses yang cukup lama bergantung pada kelengkapan data-data kriteria yang di *inputkan*.
3. Dkk., Nanik Hidayati. (2016). SISTEM PENDUKUNG KEPUTUSAN METODE AHP DAN AHP TOPSIS UNTUK PENENTUAN STAF KURIKULUM SEKOLAH. ISSN: 2302-3805. *AHP* digunakan untuk

menentukan bobot kriteria menurut pengambil keputusan, kemudian metode TOPSIS digunakan untuk menentukan peringkat calon staf kurikulum sekolah.

4. Faroqi, A. (2012). Sistem Pendukung Keputusan Untuk Menyeleksi Calon Siswa Menggunakan Metode Analytical Hierarchy Process (AHP), *VI* (1), 60–70. ISSN: 1979-8911. Sistem yang mampu memberikan dukungan keputusan dalam menyeleksi calon siswa berdasarkan Hasil UAS, Nilai UN, Keahlian, Wawancara, *Psikotest* dan Tes fisik.
5. Rais, M. S. (2016). Sistem Pendukung Keputusan Untuk Pemilihan Lokasi Perumahan Menggunakan Analytical Hierarchy Process (AHP). *Riau Journal Of Computer Science*, *2* (2), 59–72. ISSN: 2477-6890. AHP digunakan sebagai metode dalam perhitungan dalam pemilihan lokasi perumahan.