

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Teori Umum

Pada bagian ini, peneliti akan menjelaskan mengenai teori-teori umum yang digunakan dalam penelitian ini yang diambil dari pendapat beberapa ahli.

2.1.1. Sistem

Menurut (R. Saputra, 2015), Sistem merupakan kumpulan elemen-elemen yang saling terkait antara satu dengan yang lain yang tak dapat dipisahkan, untuk mencapai satu tujuan tertentu.

Menurut (Susanti, 2016), Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.

Menurut (Wibawa & Julianto, 2016), Secara umum sistem adalah kumpulan elemen yang saling berhubungan dan berinteraksi dalam satu kesatuan untuk menjalankan suatu proses pencapaian suatu tujuan utama.

Dengan pengertian diatas dapat disimpulkan bahwa sistem merupakan suatu komponen yang saling berhubungan yang membentuk suatu tujuan yang ingin dicapai.

2.1.2. Informasi

Menurut (Hermawan, 2016), Informasi adalah data yang telah diolah menjadi bentuk yang lebih berarti dan berguna bagi penerimanya untuk mengambil keputusan masa kini maupun masa yang akan datang.

Menurut(Susanti, 2016), Informasi adalah data yang telah diolah menjadi bentuk yang lebih berarti bagi yang penerimanya.

Menurut (Wibawa & Julianto, 2016), Informasi adalah data yang telah diklasifikasikan atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan.

Dengan pengertian diatas dapat disimpulkan bahwa informasi adalah suatu data yang telah diolah dan mempunyai arti yang penting dalam pengambilan keputusan.

2.1.3. Sistem Informasi

Menurut (Husda, 2012), 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.

Sedangkan menurut (Hutahaeen, 2015), sistem informasi adalah suatu sistem didalam organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan.

Pendapat lain dari (Anwar, Efendi, & Dzuhri, 2016) mengatakan, Sistem informasi merupakan sistem yang berisi jaringan sistem pengolahan data, yang dilengkapi dengan kanal-kanal komunikasi yang digunakan dalam sistem organisasi data. Elemen proses dari sistem informasi antara lain mengumpulkan data, mengelola data yang tersimpan, menyebarkan informasi.

2.1.3.1. Komponen Sistem Informasi

Menurut (Hutahaeen, 2015), Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building block*) yaitu :

1. Blok masukan (*input block*)

Input mewakili data yang masuk ke dalam sistem informasi. Input disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Blok model (*model block*)

Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data *input* dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok keluaran (*output block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok teknologi (*technology block*)

Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu mengendalikan dari sistem secara keseluruhan.

5. Blok basis data (*data base block*)

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

6. Blok kendali (*control block*)

Banyak faktor yang dapat merusak sistem informasi, misalnya bencana alam, api, temperatur tinggi, air, debu, kecurangan-kecurangan, kegagalan sistem itu sendiri, kesalahan-kesalahan ketidakefisienan, sabotase dan sebagainya. Beberapa pengendalian perlu di rancang dan diterapkan untuk menyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah atau bila terlanjur terjadi kesalahan dapat langsung diatasi.

2.1.3.2. Tipe-tipe Sistem Informasi

Menurut (Fatta, 2009), sistem informasi biasanya dibedakan menjadi beberapa tipe aplikasi, yaitu :

1. *Transaction Processing Systems* (TPS), sistem informasi terkomputerisasi yang dikembangkan untuk memproses sejumlah besar data untuk transaksi bisnis rutin.
2. *Management Information Systems* (MIS), sebuah sistem informasi pada level manajemen yang berfungsi untuk membantu perencanaan, pengendalian, dan pengambilan keputusan dengan menyediakan resume rutin dan laporan-laporan tertentu.
3. *Decision Support Systems* (DSS), sistem informasi pada level manajemen dari suatu organisasi yang mengkombinasikan data dan model analisis canggih atau peralatan data analisis untuk mendukung pengambilan yang semi terstruktur dan tidak terstruktur.
4. *Expert System* (ES), representasi pengetahuan yang menggambarkan cara seorang ahli dalam mendekati suatu masalah.

2.1.3.3. Fungsi Sistem Informasi

Fungsi sistem informasi (Kevin et al., 2016) sebagai berikut:

1. Untuk meningkatkan aksesibilitas data yang ada secara efektif dan efisien kepada pengguna, tanpa dengan prantara sistem informasi.

2. Memperbaiki produktivitas aplikasi pengembangan dan pemeliharaan sistem.
3. Menjamin tersedianya kualitas dan keterampilan dalam memanfaatkan sistem informasi secara kritis.
4. Mengidentifikasi kebutuhan mengenai keterampilan pendukung sistem informasi.
5. Mengantisipasi dan memahami akan konsekuensi ekonomi.
6. Menetapkan investasi yang akan diarahkan pada sistem informasi.
7. Mengembangkan proses perencanaan yang efektif.

2.1.4. Sistem Pendukung Keputusan

Sistem pendukung keputusan merupakan pasangan intelektual dari sumber daya manusia dengan kemampuan komputer untuk memperbaiki keputusan, yaitu sistem pendukung keputusan berbasis komputer bagi pembuat keputusan manajemen yang menghadapi masalah semi terstruktur (Ananta & Winiarti, 2013).

Pendapat lain menurut (Rohayani, 2013), Sistem pendukung keputusan merupakan suatu sistem interaktif yang mendukung keputusan dalam proses pengambilan keputusan melalui alternatif-alternatif yang diperoleh dari hasil pengolahan data, informasi dan rancangan model.

Sedangkan menurut (Khuntari & Ferdiana, 2015), Sistem pendukung keputusan merupakan sistem yang mendukung seseorang atau sekelompok kecil manajer yang bekerja sebagai *problem solving team*, untuk membuat keputusan

mengenai masalah semi terstruktur, dengan cara menyediakan sejumlah informasi yang spesifik.

2.1.4.1. Karakteristik Sistem Pendukung Keputusan

Karakteristik dari sistem pendukung keputusan adalah, (Rohayani, 2013) :

1. Sistem pendukung keputusan memberikan dukungan bagi pengambil keputusan pada situasi semi terstruktur dan tak terstruktur dengan memadukan pertimbangan manusia dan informasi terkomputerisasi.
2. Dukungan untuk semua level manajerial, dari eksekutif puncak sampai manajer lini.
3. Dukungan untuk individu dan kelompok.
4. Dukungan untuk keputusan independen dan sekuensial.
5. Dukungan di semua fase proses pengambilan keputusan, yaitu *intelligence, design, choice, dan implementation*.
6. Dukungan di berbagai proses dan gaya yang berbeda-beda.
7. Adaptivitas sepanjang waktu.
8. Mudah untuk digunakan *user*.
9. Peningkatan efektivitas dari pengambilan keputusan daripada efisiensi.
10. Kontrol penuh oleh pengambil terhadap semua langkah proses pengambilan keputusan.

11. Pengguna akhir bisa mengembangkan dan memodifikasi sendiri sistem sederhana.
12. Biasanya, model-model digunakan untuk menganalisis situasi pengambilan keputusan.
13. Akses disediakan untuk berbagai sumber daya, format, dan tipe, mulai dari sistem informasi sampai sistem berorientasi objek.
14. Dapat digunakan sebagai standalone oleh seorang pengambil keputusan pada satu lokasi atau didistribusikan di suatu organisasi secara keseluruhan dan di beberapa organisasi sepanjang rantai persediaan.

2.1.4.2. Tujuan Sistem Pendukung Keputusan

Tujuan dari sistem pendukung keputusan adalah, (Rohayani, 2013) ;

1. Membantu manajer dalam pengambilan keputusan atas masalah semi- terstruktur.
2. Meningkatkan efektivitas keputusan yang diambil manajer lebih daripada perbaikan efisiensinya.
3. Peningkatan produktivitas. Pendukung terkomputerisasi bisa mengurangi ukuran kelompok dan memungkinkan para anggotanya untuk berasal dari berbagai lokasi yang berbeda-beda (menghemat biaya perjalanan).

2.1.5. Laptop

Menurut (Ginting, 2015) Laptop atau komputer jinjing adalah komputer bergerak yang berukuran relatif kecil dan ringan, tergantung pada ukuran, bahan, dan spesifikasi laptop tersebut. Laptop terkadang disebut juga dengan komputer notebook atau notebook saja.

Sebagai komputer pribadi, laptop memiliki fungsi yang sama dengan komputer desktop (*desktop computers*) pada umumnya. Laptop sangat membantu dalam menyelesaikan tugas-tugas mereka dengan lebih cepat dan lebih baik. Di samping itu, dengan laptop dapat digunakan untuk menambah penghasilan dengan banyaknya penjualan secara online. Bagi kalangan anak muda, ada kalanya untuk hiburan dengan banyaknya game yang ditawarkan.

2.2. Tinjauan Teori Khusus

Pada bagian ini, peneliti akan menjelaskan mengenai teori-teori khusus yang digunakan dalam penelitian ini yang diambil dari pendapat beberapa ahli.

2.2.1. WEB

Web merupakan media informasi berbasis jaringan komputer yang dapat diakses di mana saja dengan biaya relatif murah. *Web* merupakan bentuk implementasi dari bahasa pemrograman web (*web programming*). Sejarah perkembangan bahasa pemrograman web diawali dengan munculnya HTML

(*Hypertext Markup Language*), yang kemudian dikembangkan dengan munculnya CSS (*Cascading Style Sheet*) yang bertujuan untuk memperindah tampilan website. Pada kenyataannya, bahasa tersebut masih memiliki kelemahan, yaitu skrip program dapat dilihat secara utuh, sehingga anda dapat mengetahui kerangka situs tersebut dan memberi peluang hacker untuk mengubahnya dengan mudah.

Bahasa pemrograman saat ini sudah sangat berkembang dengan berbagai kemudahan dalam penyajian dan *interface* yang lebih *friendly used*. Penyajian yang baik dari suatu bahasa pemrograman akan menghasilkan sebuah web yang dinamis sehingga pengunjung akan lebih mudah mendapatkan informasi yang dibutuhkan.

Beberapa bahasa pemrograman web berbasis *server (server side)* mulai muncul dan dikembangkan oleh beberapa perusahaan perangkat lunak, seperti : ASP (*Active Server Pages*) oleh *Microsoft*, JSP (*Java Server Pages*) oleh *sun Microsystems*, dan CGI (*Common Gateway Interface*).

Bahasa pemrograman tersebut memiliki beberapa kendala, misalnya: *Active Server Pages* keluaran *Microsoft* hanya dapat dijalankan melalui sistem operasi buatan *Microsoft*, dan *Common Gateway Interface* yang hanya berjalan pada sistem operasi berbasis UNIX. Untuk mengatasi keterbatasan tersebut, muncullah bahasa pemrograman web yang dapat dijalankan pada sistem operasi *Microsoft*, sekaligus sistem operasi berbasis UNIX, yaitu: bahasa pemrograman PHP (*PHP Hypertext Preprocessor*) (Wakana, 2009).

2.2.2. Analytical Hierarchy Process (AHP)

Menurut (Saragih, 2013) *Analytical Hierarchy Process* (AHP) merupakan suatu model pendukung keputusan yang dikembangkan oleh Thomas L. Saaty. Model pendukung keputusan ini akan menguraikan masalah multi faktor atau multi kriteria yang kompleks menjadi suatu hirarki, menurut Saaty, hirarki didefinisikan sebagai suatu representasi dari sebuah permasalahan yang kompleks dalam suatu struktur multi level dimana level pertama adalah tujuan, yang diikuti level faktor, kriteria, sub kriteria, dan seterusnya ke bawah hingga level terakhir dari alternatif. Dengan hirarki, suatu masalah yang kompleks dapat diuraikan ke dalam kelompok-kelompoknya yang kemudian diatur menjadi suatu bentuk hirarki sehingga permasalahan akan tampak lebih terstruktur dan sistematis. sering digunakan sebagai metode pemecahan masalah dibanding dengan metode yang lain karena alasan-alasan sebagai berikut:

1. Struktur yang berhirarki, sebagai konsekuensi dari kriteria yang dipilih, sampai pada subkriteria yang paling dalam.
2. Memperhitungkan validitas sampai dengan batas toleransi inkonsistensi berbagai kriteria dan alternatif yang dipilih oleh pengambil keputusan. Memperhitungkan daya tahan *output* analisis sensitivitas pengambilan keputusan.

2.2.3. Tahapan Metode AHP

Dalam metode *Analytical Hierarchy Process* dilakukan langkah-langkah sebagai berikut (Saragih, 2013):

1. Mendefinisikan masalah dan menentukan solusi yang diinginkan.

Dalam tahap ini penulis berusaha menentukan masalah yang akan penulis pecahkan secara jelas, detail dan mudah dipahami. Dari masalah yang ada penulis coba tentukan solusi yang mungkin cocok bagi masalah tersebut. Solusi dari masalah mungkin berjumlah lebih dari satu. Solusi tersebut nantinya penulis kembangkan lebih lanjut dalam tahap berikutnya.

2. Membuat struktur hierarki yang diawali dengan tujuan utama. Setelah menyusun tujuan utama sebagai level teratas akan disusun level hirarki yang berada di bawahnya yaitu kriteria-kriteria yang cocok untuk mempertimbangkan atau menilai alternatif yang penulis berikan dan menentukan alternatif tersebut. Tiap kriteria mempunyai intensitas yang berbeda-beda. Hirarki dilanjutkan dengan subkriteria (jika mungkin diperlukan).

3. Membuat matrik perbandingan berpasangan yang menggambarkan kontribusi relatif atau pengaruh setiap elemen terhadap tujuan atau kriteria yang setingkat di atasnya. Matriks yang digunakan bersifat sederhana, memiliki kedudukan kuat untuk kerangka konsistensi, mendapatkan informasi lain yang mungkin dibutuhkan dengan semua perbandingan yang mungkin dan mampu menganalisis kepekaan

prioritas secara keseluruhan untuk perubahan pertimbangan. Pendekatan dengan matriks mencerminkan aspek ganda dalam prioritas yaitu mendominasi dan didominasi. Perbandingan dilakukan berdasarkan judgment dari pengambil keputusan dengan menilai tingkat kepentingan suatu elemen dibandingkan elemen lainnya. Untuk memulai proses perbandingan berpasangan dipilih sebuah kriteria dari level paling atas hirarki misalnya K dan kemudian dari level di bawahnya diambil elemen yang akan dibandingkan misalnya E1,E2,E3,E4,E5.

4. Melakukan Mendefinisikan perbandingan berpasangan sehingga diperoleh jumlah penilaian seluruhnya sebanyak $n \times [(n-1)/2]$ buah, dengan n adalah banyaknya elemen yang dibandingkan. Hasil perbandingan dari masing-masing elemen akan berupa angka dari 1 sampai 9 yang menunjukkan perbandingan tingkat kepentingan suatu elemen. Apabila suatu elemen dalam matriks dibandingkan dengan dirinya sendiri maka hasil perbandingan diberi nilai 1. Skala 9 telah terbukti dapat diterima dan bisa membedakan intensitas antar elemen. Hasil perbandingan tersebut diisikan pada sel yang bersesuaian dengan elemen yang dibandingkan. Skala perbandingan perbandingan berpasangan dan maknanya yang diperkenalkan oleh Saaty bisa dilihat di bawah.

Intensitas Kepentingan:

- a. 1 berarti kedua elemen sama pentingnya, Dua elemen mempunyai pengaruh yang sama besar.
 - b. 3 berarti elemen yang satu sedikit lebih penting daripada elemen yang lainnya, Pengalaman dan penilaian sedikit menyokong satu elemen dibandingkan elemen yang lainnya.
 - c. 5 berarti elemen yang satu lebih penting daripada yang lainnya, Pengalaman dan penilaian sangat kuat menyokong satu elemen dibandingkan elemen yang lainnya.
 - d. 7 berarti satu elemen jelas lebih mutlak penting daripada elemen lainnya, Satu elemen yang kuat disokong dan dominan terlihat dalam praktek.
 - e. 9 berarti satu elemen mutlak penting daripada elemen lainnya, Bukti yang mendukung elemen yang satu terhadap elemen lain memiliki tingkat penegasan tertinggi yang mungkin menguatkan.
 - f. 2,4,6,8 berarti nilai-nilai antara dua nilai pertimbangan-pertimbangan yang berdekatan, Nilai ini diberikan bila ada dua kompromi di antara 2 pilihan Kebalikan = Jika untuk aktivitas i mendapat satu angka dibanding dengan aktivitas j , maka j mempunyai nilai kebalikannya dibanding dengan i.
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 yang merupakan bobot setiap elemen untuk penentuan prioritas elemen-elemen pada tingkat hirarki terendah sampai mencapai tujuan. Penghitungan dilakukan lewat cara menjumlahkan nilai setiap kolom dari matriks, membagi setiap nilai dari kolom dengan total kolom yang bersangkutan untuk memperoleh normalisasi matriks, dan menjumlahkan nilai-nilai dari setiap baris dan membaginya dengan jumlah elemen untuk mendapatkan rata-rata.
8. Memeriksa konsistensi hirarki. Adapun yang diukur dalam Analytical Hierarchy Process adalah rasio konsistensi dengan melihat index konsistensi. Konsistensi yang diharapkan adalah yang mendekati sempurna agar menghasilkan keputusan yang mendekati valid. Walaupun sulit untuk mencapai yang sempurna, rasio konsistensi diharapkan kurang dari atau sama dengan 10 %.

Rumus Untuk Menentukan Rasio Konsistensi (CR) Indeks konsistensi dari matriks berordo n dapat diperoleh dengan rumus :

$$CI = \frac{\lambda_{maksimum} - n}{n - 1}$$

dimana :

CI = Indek konsistensi (Consistency Index)

λ maksimum = Nilai eigen terbesar dari matrik berordo n

λ maksimum didapat dengan menjumlahkan hasil perkalian jumlah kolom dengan eigen vektor utama.

Apabila $C.I = 0$, berarti matriks konsisten.

Batas ketidakkonsistenan yang ditetapkan Saaty diukur dengan menggunakan rasio konsistensi (CR), yakni perbandingan indek konsistensi dengan nilai pembangkit random (RI). Nilai RI bergantung pada ordo matrik n .

Tabel 2.1 Tabel Nilai RI

N	RI
1	0.00
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	1.58

CR dirumuskan :

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

Dimana :

CR = Consistency Ratio

CI = Consistency Indeks

IR = Indeks Random Consistency

2.2.4. Software Development Life Cycle (SDLC)

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) (S & Shalahuddin, 2015).

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*)

Mengonversi 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*.

2.2.4.1. 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 terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*) (S & Shalahuddin, 2015).

1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2. Desain

Desain perangkat lunak adalah proses multistep yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. Pembuatan kode program

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

4. Pengujian

Pengujian fokus pada perangkat lunak yang dilakukan dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal

ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

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

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

2.2.5. Metode Pengujian *Black Box*

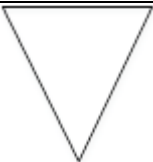
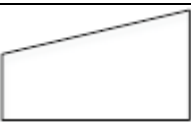
Metode ujicoba *black box* memfokuskan pada keperluan fungsional dari sistem, karena itu ujicoba *black box* memungkinkan pengembang sistem atau aplikasi untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program. Ujicoba *black box* bukan merupakan alternatif dari ujicoba *white box*, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain menggunakan metode *white box*. Ujicoba *black box* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya; fungsi-fungsi yang salah atau hilang, kesalahan interface, kesalahan dalam struktur data atau akses *database* eksternal (jika ada), kesalahan performa, serta kesalahan inisialisasi dan terminasi (Pare, 2013).

2.2.6. Aliran Sistem Informasi (ASI)

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, (Ismael, 2017).

Tabel 2.2 Simbol-simbol Aliran Sistem Informasi

Simbol	Keterangan
	Proses Komputer
	Dokumen
	Penghubung
	Garis alur
	<i>Output</i> di monitor

Simbol	Keterangan
	Manual Proses
	<i>File Storage</i>
	Penghubung antar halaman
	<i>Input menggunakan keyboard</i>
	<i>Input output menggunakan disket</i>

2.2.7. Unified Modeling Language (UML)

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 memspesifikasikan, 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 (S & Shalahuddin, 2015).

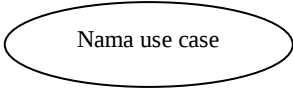
2.2.7.1. Use Case Diagram

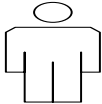
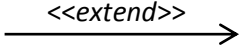
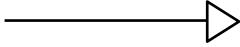
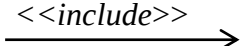
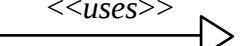
Use case merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Syarat penamaan pada *use case* adalah nama didefinisikan sesimple mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefenisian apa yang disebut *actor* dan *use case* (S & Shalahuddin, 2015).

1. *Actor* merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari *actor* itu orang, tapi *actor* belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukaran pesan antara unit atau *actor*.

Tabel 2.3 Penjelasan Use Case Diagram

Simbol	Deskripsi
<i>Use case</i> 	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>

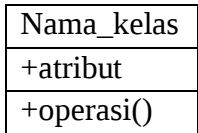
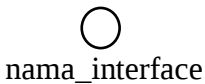
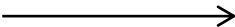
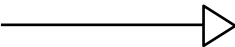
Simbol	Deskripsi
Aktor  Nama aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
Asosiasi 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek
Generalisasi / <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan / <i>include</i> / <i>uses</i>  	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i>: 1. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan 2. <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan. Kedua interpretasi diatas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan

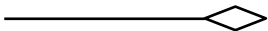
2.2.7.2. Class Diagram

Diagram kelas menggambarkan struktur sistem dari segi pendefenisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi (S & Shalahuddin, 2015).

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Tabel 2.4 Penjelasan Class Diagram

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi / <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi beralih / <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
generalisasi 	Relasi antar kelas dengan makna generalisasi – spesialisasi (umum – khusus)

Simbol	Deskripsi
	Kebergantungan / <i>dependency</i> Relasi antar kelas dengan makna kebergantungan antar kelas
	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>)

2.2.7.3. Activity Diagram

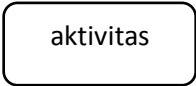
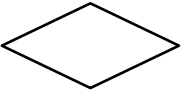
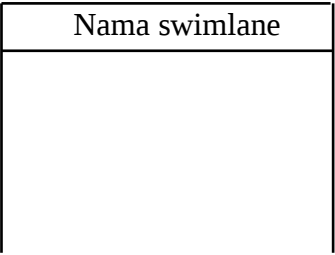
Diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (S & Shalahuddin, 2015).

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

4. Rancangan menu yang ditampilkan pada perangkat lunak.

Tabel 2.5 Penjelasan 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
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

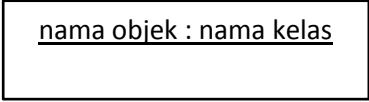
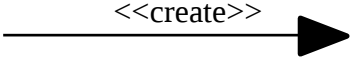
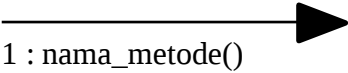
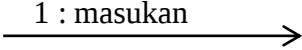
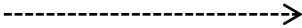
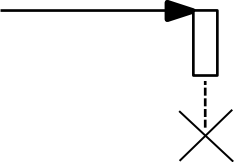
2.2.7.4. Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek.

Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefenisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak (S & Shalahuddin, 2015).

Tabel 2.6 Penjelasan Sequence Diagram

Simbol	Deskripsi
Aktor  Nama aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri
Garis hidup / <i>lifeline</i> 	Menyatakan kehidupan suatu objek

Simbol	Deskripsi
Objek 	Menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan
pesan tipe create 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
pesan tipe call 	Menyatakan suatu objek operasi memanggil / metode yang ada pada objek lain atau dirinya sendiri.
Pesan tipe send 	Menyatakan bahwa suatu objek mengirim data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
Pesan tipe return 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
Pesan tipe destroy 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada creat maka ada destroy

2.2.8. PHP

PHP (Hypertext Preprocessor) merupakan bahasa pemrograman berbasis *web* yang memiliki kemampuan untuk memroses dan mengolah data secara dinamis. *PHP* dapat dikatakan sebagai sebuah *server-side embedded script language*, artinya semua sintaks dan perintah program yang anda tulis akan sepenuhnya dijalankan oleh *server*, tetapi dapat disertakan pada halaman HTML biasa. Pada umumnya, semua aplikasi yang dibangun menggunakan *PHP* akan memberikan hasil web browser, tetapi prosesnya secara seluruhan dijalankan pada server (Wakana, 2009).

2.2.9. MySQL

MySQL (My Structured Query Language) atau yang biasa dibaca mai-sekuel adalah sebuah program pembuat dan pengelola *database* atau yang sering disebut denagn DBMS (*Database Management System*), sifat dari DBMS ini adalah Open Source. *MySQL* sebenarnya produk yang berjalan pada platform Linux, dengan adanya perkembangan dan banyaknya pengguna, serta lisensi dari *database* ini adalah Open Source, maka para pengembang kemudian merilis versi windows.

Selain itu *MySQL* juga merupakan program pengakses *database* yang bersifat jaringan, sehingga dapat digunakan untuk aplikasi Multi User (Banyak Pengguna). Kelebihan lain dari *MySQL* adalah menggunakan bahasa query (permintaan) standar *SQL (Structures Query Language)*. *SQL* adalah suatu bahasa

permintaan yang terstruktur, *SQL* telah distandardkan untuk semua program pengakses *database* seperti *Oracle*, *PosgreSQL*, *SQL server* dan lain-lain.

Sebuah program penghasil *database*, *MySQL* tidak mungkin berjalan sendiri tanpa adanya sebuah aplikasi pengguna (interface) yang berguna sebagai program aplikasi pengakses *database* yang dihasilkan. *MySQL* dapat didukung oleh hampir semua program aplikasi baik yang *Open Source* seperti *PHP* maupun yang tidak *Open Source* yang ada pada platform windows seperti *visual basic*, *Delphi* dan lainnya (Nugroho, 2008).

Menurut (A. Saputra, 2012) *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. *Multiusers*, dimana *MySQL* dapat digunakan oleh banyak *user* dalam waktu yang bersamaan tanpa mengalami konflik.

2.2.10. XAMPP

XAMPP merupakan perangkat lunak bebas, yang mendukung banyak sistem operasi, dan merupakan kompilasi dari beberapa program. *Xampp* merupakan kepanjangan dari hurufnya yaitu :

X : Program ini dapat dijalankan di banyak sistem operasi, seperti *Windows*, *Linux*, *Mac OS* dan juga *Solaris*.

A : **Apache**, merupakan aplikasi web server. Tugas utama dari *Apache* adalah menghasilkan halaman web yang benar kepada *user* berdasarkan kode PHP yang dituliskan oleh pembuat web atau *user*.

M : **MySQL**, merupakan aplikasi data server. Perkembangannya disebut juga *Sql* yang merupakan kepanjangan dari *Structured Query Language*. *Sql* merupakan bahasa terstruktur yang digunakan untuk mengolah *database*.

P : **PHP**, merupakan bahasa pemrograman web, dimana *user* dapat menggunakan bahasa pemrograman ini untuk membuat web yang bersifat *server-side scripting*.

P : **Perl**, yaitu merupakan bahasa pemrograman untuk segala keperluan, dan dikembangkan pertama kali oleh Larry Wall di mesin *Unix*. (Andriyani & Gea, 2016).

2.3. Penelitian Terdahulu

Berikut ini beberapa penelitian terdahulu yang menjadi acuan penulis dalam melakukan penelitian.

Tabel 2. 7 Penelitian Terdahulu

No	Nama Peneliti & Judul Penelitian	Jurnal Penelitian	Metode Penelitian	Hasil Penelitian
1.	(Priyanto, H, & Sutinah, 2017) Pemilihan Laptop Terbaik Menggunakan Metode Analytical Hierarchy Process Pada PT Swadharma Sarana Informatika	<i>BINA INSANI ICT JOURNAL</i> , Vol.4, No.1, ISSN :2355-3421	Analytical Hierarchy Process	Hasil penelitian ini adalah pemilihan laptop yang akan digunakan pada bagian monitoring serta kriteria yang terdapat pada laptop ASUS A456 akan dijadikan sebagai minimum spesifikasi terupdate bagi PT. Swadharma Sarana Informatika.
2.	(Hartanto & Prasetyowati, 2012) Sistem Pendukung Keputusan Pemilihan Laptop Berbasis Web dengan Metode Analytical Hierarchy Process (Studi Kasus : SAMCO COMPUTER)	ULTIMATIC S, Vol.IV, No. 2, ISSN 2085-4552	Analytical Hierarchy Process	Hasil penelitian ini ditemukan bahwa Metode Analytical Hierarchy Process membuat permasalahan yang luas dan tidak terstruktur menjadi suatu model yang fleksibel dan mudah dipahami. dan menunjukkan hasil rekomendasi laptop yang diberikan oleh sistem.
3.	(Saragih, 2013) Penerapan Metode Analitical Hierarchy Process(AHP) Pada Sistem Pendukung Keputusan Pemilihan Laptop	Pelita Informatika Budi Dharma, Vol.I V, No.2, ISSN : 2301-9425	Analytical Hierarchy Process	Hasil Penelitian ini sebuah sistem pendukung keputusan pemilihan laptop dengan menggunakan metode Analitical Hierarchy Process (AHP), agar pengguna dapat menentukan pilihan laptop dengan tepat sesuai dengan keinginan dan anggarannya.
4	(Perdhana et al.,	ISSN 2338-	Analytical	Hasil penelitian ini untuk merancang dan

	n.d.) 2015 Sistem Pendukung Keputusan Pemilihan Jenis Laptop Dengan Menggunakan Metode Analytical Hierarchy Process	4018	Hierarchy Process	membuat aplikasi “Sistem Pendukung Keputusan Pemilihan Jenis Laptop dengan Metode AHP” untuk menjadi alternatif solusi alat bantu pengambilan keputusan atau rekomendasi dalam memilih notebook, dan PC Tablet yang akan dibeli oleh konsumen.
5.	(Nasution, 2014) Sistem Pendukung Keputusan Penentuan Pemilihan Laptop Dengan Menerapkan Fuzzy Tahani	Pelita Informatika Budi Darma, Vol.V I, No.1, ISSN : 2301-9425	Fuzzy Tahani	Hasil penelitian ini adalah perancangan aplikasi untuk memudahkan dalam pencarian suatu paket komputer sesuai dengan keinginan pembeli. Studi kasus ini memberikan hasil berupa paket komputer hasil rekomendasi yang didapat dari proses-proses fuzzy berdasarkan nilai-nilai batas himpunan fuzzy, variabelvariabel fuzzy dan variabel-variabel non fuzzy.
6.	(Hendry & Seng, 2016) Rancang Bangun Aplikasi Rekomendasi Pembelian Laptop Dengan Metode Fuzzy Database Model Tahani Berbasis Web	Jurnal Ilmiah Komputer dan Informatika(KOMPUTA), Vol.3, No.2, I SSN : 2089- 9033	Fuzzy Database Model Tahani	Hasil Penelitian ini adalah Aplikasi yang dibangun dapat diimplementasikan dengan baik menggunakan metode Fuzzy Database Model Tahani untuk menentukan rekomendasi laptop yang sesuai untuk pembeli pada Toko Ricky Computer.
7.	(Suryanto, 2016) Pengambilan Keputusan Dalam Pemilihan Notebook Berbasis Teknologi Dengan Metode Multycriteria Decision Making(MCDM)	<i>BINA INSANI ICT JOURNAL, Vol. 3, No.2, I SSN : 2355- 3421</i>	Multycriteri a Decision Making(MC DM)	Hasil penelitian yang melakukan pengujian prioritas dengan Software package Expert Choice memberikan prioritas yang sama dengan metode MCDM, walaupun hasil yang diperoleh terjadi perbedaan yang sangat tipis yaitu sebesar 0.001; hal ini membuktikan bahwa pendekatan metode MCDM dapat dijadikan sebagai tolak ukur pengambilan keputusan dan pembuktian praktis Software package Expert Choice.
8.	(Kurniasih, 2013) Sistem Pendukung Keputusan Pemilihan Laptop Dengan Metode Topsis	Pelita Informatika Budi Darma, Vol.II I, No.2, ISSN : 2301-9425	Topsis	Hasil penelitian dalam perhitungan metode TOPSIS yang didapatkan secara manual sama dengan hasil perhitungan yang didapatkan secara komputerisasi. Hasil dari proses pengimplementasian metode dan TOPSIS dapat mengurutkan alternatif

9.	(Purwanto, 2017) Sistem Penunjang Keputusan Pemilihan Notebook Dengan Menggunakan Metode Topsis	Jurnal Ilmu Topsis Pengetahuan dan Teknologi Komputer, Vol.2, No.2, ISSN :2527-4864	Topsis	dari nilai yang terbesar ke nilai yang terkecil. Hasil penelitian ini adalah Tehnique for Order Preference by Similarity to Ideal Solution (TOPSIS) dapat menjadi alternatif pemilihan Notebook. Hasil dari preferensi disarankan bahwa urutan prioritas alternatif Notebook nilai tertinggi adalah Toshiba, Asus, Samsung, dan HP.
10.	(Syafitri & Dewi, 2016) Penerapan Metode Weighted Product Dalam Sistem Pendukung Keputusan Pemilihan Laptop Berbasis WEB.	Seman TIK, Vol.2, No.1, ISSN 2502-8928	Weighted Product	Hasil dari penelitian ini memberikan saran laptop sesuai dengan kebutuhan spesifikasi untuk calon pembeli dengan tingkat akurasi perhitungan 100% berdasarkan perhitungan manual dan perhitungan pada sistem pendukung keputusan pemilihan laptop.
11.	(Khairul, Simaremare, & Siahaan, 2016) Sistem Pendukung Keputusan dalam Memilih Laptop yang Tepat Menggunakan Simple Additive Weighting	<i>International Journal of Recent Trends in Engineering & Research (IJRTER)</i> , Volume 02, Issue 12, ISSN: 2455-1457	Simple Additive Weighting	Hasil penelitian ini adalah merancang sebuah sistem pendukung keputusan untuk pemilihan laptop agar pengguna bisa menentukan pilihan sesuai dengan keinginan dan kemampuan.
12.	(Tampi & Tumewu, n.d.) 2016 Pengambilan keputusan pelanggan dalam memilih laptop menggunakan metode analytical hierarchy process (Kasus :Hp, Asus dan Toshiba	Jurnal EMBA , Vol.4, No.1, ISSN 2303-1174	Analytical Hierarchy Process	Hasil penelitian berdasarkan data keseluruhan, pelanggan akan memilih HP sebagai laptop yang paling diminati di Manado dibandingkan dengan alternatif lainnya. Hasil ini menunjukkan bahwa ketika orang ingin memilih laptop, mereka lebih suka pilih HP sebagai laptop yang paling disukai di antara alternatif lainnya.
