

BAB I

PENDAHULUAN

1.1 Latar Belakang

Posyandu merupakan salah satu bentuk Upaya Kesehatan Bersumber Masyarakat (UKBM) yang dikelola dari, oleh, untuk dan bersama masyarakat, guna memberdayakan masyarakat dan memberikan kemudahan kepada masyarakat dalam memperoleh pelayanan kesehatan dasar. Upaya peningkatan peran dan fungsi Posyandu bukan semata-mata tanggung jawab pemerintah saja, namun semua komponen yang ada di masyarakat, termasuk kader.

Dalam Peraturan Menteri Dalam Negeri Nomor 19 Tahun 2011 tentang Pedoman Pengintegrasian Layanan Sosial Dasar di Pos Pelayanan Terpadu (Posyandu) menerangkan bahwa Posyandu adalah salah satu bentuk upaya kesehatan bersumber daya masyarakat yang dikelola dan diselenggarakan dari, oleh, untuk dan bersama masyarakat dalam penyelenggaraan pembangunan kesehatan, guna memberdayakan masyarakat dan memberikan kemudahan kepada masyarakat dalam memperoleh pelayanan kesehatan dasar untuk mempercepat penurunan angka kematian ibu dan bayi. Serta pada Pasal 1 ayat 10 menerangkan bahwa Kader Posyandu yang selanjutnya disebut kader adalah anggota masyarakat yang bersedia, mampu dan memiliki waktu untuk menyelenggarakan kegiatan Posyandu secara sukarela.

Keberhasilan Posyandu salah satunya dipengaruhi oleh kinerja kader, dengan motivasi yang tinggi dalam kegiatan Posyandu akan meningkatkan kinerja kader Posyandu. Namun permasalahan yang terjadi adalah masih banyak kader yang kurang termotivasi dalam kegiatan Posyandu. Permasalahan lain yang sering

terjadi yaitu pelayanan yang dilakukan kader posyandu selama ini masih mendapatkan komplain dari masyarakat, komplain tersebut diantaranya dikarenakan pelayanan yang lambat ketika masyarakat membutuhkan bantuan, sering salah dalam memberikan penanganan bantuan kepada masyarakat serta komunikasi yang kurang ramah terhadap masyarakat. Untuk itu diperlukan adanya penilaian kinerja setiap kader posyandu. Namun pihak posyandu Aliantan Kabupaten Rokan Hulu belum memiliki metode dan sistem yang dapat digunakan untuk melakukan penilaian kinerja kader posyandu tersebut. Sehingga hal ini menjadi sebuah permasalahan yang cukup berarti bagi posyandu Aliantan Kabupaten Rokan Hulu.

Maka untuk menyelesaikan permasalahan di atas dapat menggunakan sistem pendukung keputusan. Sistem pendukung keputusan merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan dan manipulasi data. Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semi terstruktural dan situasi yang tidak terstruktural, dimana tidak seorangpun tahu secara pasti bagaimana keputusan seharusnya dibuat (Adri Priadana, 2018).

Salah satu metode dalam sistem pendukung keputusan adalah metode *Additive Ratio Assesment* (ARAS). Metode ARAS merupakan metode pengambilan keputusan multikriteria berdasarkan pada konsep perankingan menggunakan *utility degree* yaitu dengan membandingkan nilai indeks keseluruhan alternatif terhadap nilai indeks keseluruhan alternatif optimal (Abdul Yunus, 2020). Pada penelitian yang dilakukan Asnita Susilawati Nadeak pada tahun 2019 dengan judul “Penerapan Metode ARAS (*Additive Ratio Assesment*) Dalam Penilaian Guru Terbaik” telah terbukti bahwa metode ARAS dapat

digunakan untuk pengambilan keputusan untuk penilaian kinerja dengan banyak alternatif dan kriteria.

Hal-hal tersebut yang mendorong untuk melakukan penelitian ini dengan judul “**Aplikasi Penilaian Kinerja Kader Posyandu Aliantan Kabupaten Rokan Hulu Menggunakan Metode ARAS**”, diharapkan dengan dibuatnya penelitian ini dapat berguna untuk memberikan kemudahan dalam menentukan penilaian kinerja kader posyandu Aliantan Kabupaten Rokan Hulu.

1.2 Rumusan Masalah

Dengan mengacu pada uraian latar belakang masalah diatas, maka permasalahan yang dibahas dan diteliti yaitu :

1. Bagaimana merancang aplikasi dengan menggunakan metode ARAS yang dapat digunakan untuk penilaian kinerja kader pada Posyandu Aliantan secara tepat dan akurat?
2. Bagaimana menguji aplikasi yang menerapkan metode ARAS yang dapat digunakan untuk penilaian kinerja kader pada Posyandu Aliantan?

1.3 Manfaat Penelitian

Adapun manfaat yang bisa didapat dari penulisan penelitian ini adalah:

1. Dapat mempermudah pihak Posyandu Aliantan dalam mengambil sebuah keputusan dalam penilaian kinerja kader.
2. Dapat menambah pengetahuan dan wawasan bagi peneliti mengenai sistem pendukung keputusan ini serta memberi bekal pengetahuan yang dapat

dipergunakan untuk persiapan dalam menghadapi kerja di masa yang akan datang.

3. Dapat menjadi referensi dan acuan dalam membangun sistem pendukung keputusan menggunakan metode ARAS.

1.4 Tujuan Penelitian

Adapun beberapa tujuan dari penelitian ini adalah :

1. Untuk merancang aplikasi dengan menggunakan metode ARAS yang dapat digunakan untuk penilaian kinerja kader pada Posyandu Aliantan secara tepat dan akurat.
2. Untuk menguji aplikasi yang menerapkan metode ARAS yang dapat digunakan untuk penilaian kinerja kader pada Posyandu Aliantan.

1.5 Batasan Masalah

Untuk menghindari pembahasan yang meluas, maka ada beberapa batasan permasalahan yaitu :

1. Data yang digunakan sebagai sampel berjumlah 15 data kader posyandu selama tahun 2021.
2. Aplikasi yang dirancang menggunakan aplikasi pemrograman berbasis website dengan PHP dan database MySQL.

1.6 Sistematika Penulisan

Agar penelitian ini tersusun dengan mudah dan dapat dimengerti dengan baik maka sistematika penulisan sebagai berikut :

BAB I PENDAHULUAN

Pada bab pendahuluan menguraikan tentang latar belakang masalah, rumusan masalah, manfaat penelitian, tujuan penelitian, batasan masalah dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini membahas dasar teori tentang teknik-teknik pengambilan keputusan, metode ARAS, pemodelan sistem, *flowchart*, aplikasi-aplikasi yang digunakan dan teori-teori yang mendukung pengembangan sistem serta analisa perancangan sistem yang digunakan.

BAB III METODE PENELITIAN

Bab ini berisi tentang menganalisis dan mendefinisikan masalah dan solusinya untuk pembentukan sistem terkomputerisasi.

BAB IV HASIL DAN PEMBAHASAN

Bab ini membahas implementasi dari sistem yaitu hasil rancangan aplikasi sistem pendukung keputusan disertai cara kerja dan penggunaannya. Adapun pengujian-pengujian aplikasi dilakukan untuk mengetahui kekurangan yang terdapat pada perangkat lunak. Analisa dilakukan untuk memperbaiki aplikasi apabila terdapat kesalahan atau kekurangan pada program yang diperoleh.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yaitu merupakan jawaban dari permasalahan penelitian ini sedangkan untuk saran adalah pengembangan dan perbaikan serta penyempurnaan terhadap program aplikasi yang telah dibuat untuk meningkatkan kualitas metode ARAS

BAB II

TINJAUAN PUSTAKA

2.1 Aplikasi

Menurut Adi Widarna dan Sri Rahayu (2017), aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. Pengertian aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya, aplikasi merupakan suatu perangkat komputer yang siap pakai bagi user.

Secara istilah pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Menurut kamus computer eksekutif, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu tehnik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan (Andi Juansyah, 2015).

Pengertian aplikasi menurut Kamus Besar Bahasa Indonesia, aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu.

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi (*application suite*). Contohnya adalah *Microsoft Office* dan *OpenOffice.org*, Bahasa Pemrograman yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi.

Berikut ini beberapa penggolongan aplikasi yaitu (Sevi Andiasari, 2017):

1. Perangkat lunak perusahaan (*enterprise*)
2. Perangkat lunak infrastruktur perusahaan
3. Perangkat lunak informasi kerja
4. Perangkat lunak media dan hiburan
5. Perangkat lunak pendidikan
6. Perangkat lunak pengembangan media
7. Perangkat lunak rekayasa produk

Selain itu aplikasi juga mempunyai fungsi sebagai pelayan kebutuhan beberapa aktivitas yang dilakukan oleh manusia seperti sistem untuk *software* jual beli, permainan atau *game online*.

2.2 Sistem Pendukung Keputusan

Sistem Pendukung Keputusan merupakan penggabungan sumber-sumber kecerdasan individu dengan kemampuan komponen untuk memperbaiki kualitas keputusan. Sistem Pendukung Keputusan juga merupakan sistem informasi berbasis komputer untuk manajemen pengambilan keputusan yang menangani masalah-masalah semi struktur (Rika Yunitarini, 2015).

Sistem pendukung keputusan merupakan suatu penerapan sistem informasi yang ditujukan untuk membantu pimpinan dalam proses pengambilan keputusan. Sistem pendukung keputusan menggabungkan kemampuan komputer dalam pelayanan interaktif dengan pengolahan data yang memanfaatkan model atau aturan penyeleksian yang tidak terstruktur (Malisa, 2016).

Dengan pengertian diatas dapat dijelaskan bahwa sistem pendukung keputusan bukan merupakan alat pengambilan keputusan, melainkan merupakan sistem yang membantu pengambil keputusan dengan melengkapi mereka dengan informasi dari data yang telah diolah dengan relevan dan diperlukan untuk membuat keputusan tentang suatu masalah dengan lebih cepat dan akurat.

2.2.1 Pengertian Sistem

Sebuah sistem terdiri dari berbagai unsur yang saling melengkapi dalam mencapai tujuan dan sasaran. Unsur-unsur yang terdapat dalam sistem itulah yang disebut dengan subsistem. Subsistem-subsistem tersebut harus saling berhubungan dan berinteraksi melalui komunikasi yang relevan sehingga sistem dapat bekerja secara efektif dan efisien.

Kata Sistem berasal dari bahasa Yunani (*systema*) dan bahasa latin (*systema*) adalah suatu kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi. Definisi sistem berkembang sesuai dengan konteks dimana pengertian sistem itu digunakan.

Sistem merupakan elemen-elemen yang saling berinteraksi untuk mencapai tujuan tertentu. Suatu sistem terdiri dari beberapa sub sistem yang

saling berhubungan untuk membentuk suatu kesatuan sebagai sasaran dari sistem tersebut dapat tercapai.

Ada beberapa pendapat mengenai sistem diantaranya adalah (Paulina Thomas, 2015):

1. Ludwing : sistem adalah seperangkat unsur yang saling bergubungan dan saling mempengaruhi dalam satu lingkungan tertentu.
2. Davis : sistem merupakan bagian-bagian yang beroperasi secara bersama-sama untuk mencapai beberapa tujuan.
3. McLeod : sistem, yaitu sekelompok elemen yang terintegrasi untuk mencapai suatu tujuan.
4. Shorde : sistem yaitu perilaku berdasarkan tujuan tertentu, keseluruhan, keterbukaan, terjadi informasi, terjadi korelasi, memiliki mekanisme kontrol artinya terdapat kekuatan yang mempersatukan dan mempertahankan sistem yang bersangkutan.

Jadi pengertian sistem secara umum adalah jaringan kerja sama bagian-bagian atau unsur-unsur yang saling berhubungan guna mencapai tujuan yang diinginkan.

2.2.2 Pengertian Sistem Pendukung Keputusan

Sistem Pendukung Keputusan merupakan bagian dari sistem informasi berbasis komputer termasuk sistem berbasis pengetahuan atau manajemen pengetahuan yang dipakai untuk mendukung pengambilan keputusan, dapat juga dikatakan sebagai sistem yang mengelola data menjadi informasi untuk mengambil keputusan (Noprin Pakaya, 2017).

Sistem Pendukung Keputusan (SPK) atau *Decision Support Sistem (DSS)* merupakan sistem yang berbasis komputer dan ditujukan untuk membantu pengambil keputusan dengan memanfaatkan data dan model tertentu untuk memecahkan berbagai persoalan yang tidak terstruktur. SPK dibangun bukan untuk mengambil keputusan secara langsung, akan tetapi sebagai sistem yang membantu dalam pengambilan keputusan (Bany Setiadji, 2016).

Dalam membangun sistem informasi, juga dibutuhkan sistem manajemen data efektif, sehingga data yang terkumpul dapat diolah, dieksplorasi tepat agar sistem dapat bekerja dengan maksimal. Agar sistem informasi tersebut dapat beroperasi secara optimal, maka dibutuhkan teknik informasi yang telah terbukti memiliki kinerja yang sangat unggul. Digunakan teknik informasi sebagai basis pembangunan sistem yang akan member jaminan lancarnya aliran data.

Aplikasi Sistem Pendukung Keputusan menggunakan data, memberikan antarmuka pengguna yang mudah, dan dapat menggabungkan pemikiran pengambil keputusan (Nasyuha, 2017).

2.2.3 Jenis Keputusan

Ada beberapa jenis keputusan berdasarkan sifat dan jenisnya (Abdul Halim Hasugian, 2018) :

1. Keputusan Terprogram

Yaitu keputusan yang bersifat berulang dan rutin, sedemikian sehingga suatu prosedur pasti telah dibuat untuk menanganinya.

2. Keputusan Tak Terprogram

Yaitu keputusan yang bersifat baru, tidak terstruktur dan jarang konsekuen. Tidak ada metode yang pasti untuk menangani masalah tersebut.

2.1.4 Fase-Fase Pengambilan Keputusan

Terdapat beberapa model yang menggambarkan proses pengambilan keputusan. Proses ini terdiri dari tiga fase, yaitu sebagai berikut:

1. Fase *Intelligence*

Tahap ini merupakan proses penelusuran dan pendeteksian dari lingkup problematika serta proses pengenalan masalah. Data masukan diperoleh, diproses, dan diuji dalam rangka mengidentifikasi masalah.

2. Fase *Design*

Tahap ini merupakan proses menemukan, mengembangkan, dan menganalisis alternatif tindakan yang bisa dilakukan. Tahap ini meliputi proses untuk mengerti masalah, menurunkan solusi dan menguji kelayakan solusi.

3. Fase *Choice*

Pada tahap ini dilakukan proses pemilihan diantara berbagai alternatif tindakan yang mungkin dijalankan. Hasil pemilihan tersebut kemudian diimplementasikan dalam proses pengambilan keputusan.

2.2.5 Struktur Keputusan

Keputusan yang diambil untuk menyelesaikan suatu masalah dilihat dari kestrukturannya, dapat dibagi menjadi:

1. Keputusan Terstruktur (*Structured Decision*)

Keputusan terstruktur adalah keputusan yang dilakukan secara berulang-ulang dan bersifat rutin. Prosedur pengambilan keputusan sangatlah jelas. Keputusan tersebut terutama dilakukan pada manajemen tingkat bawah. Misalnya, keputusan pemesanan barang dan keputusan penagihan hutang.

2. Keputusan Semi Terstruktur (*Structured Decision*)

Keputusan semiterstruktur adalah keputusan yang memiliki dua sifat. Sebagian keputusan bisa ditangani oleh komputer dan yang lain tetap harus dilakukan pengambil keputusan. Prosedur dalam pengambilan keputusan tersebut secara garis besar sudah ada, tetapi ada beberapa hal yang masih memerlukan kebijakan dari pengambil keputusan. Contohnya penjadwalan produksi dan pengendalian persediaan.

3. Keputusan Tak Terstruktur (*Unstructured Decision*)

Keputusan tak terstruktur adalah keputusan yang penanganannya rumit karena tidak terjadi berulang-ulang atau tidak selalu terjadi. Keputusan tersebut menurut pengalaman dan berbagai sumber yang bersifat eksternal. Keputusan tersebut umumnya terjadi pada *manager* tingkat atas. Contohnya adalah keputusan untuk pengembangan teknologi baru.

2.2.6 Komponen-Komponen Sistem Pendukung Keputusan

Sistem pendukung keputusan terdiri dari 4 komponen utama (Nila Susanti, 2013), yaitu :

1. Subsistem manajemen data berfungsi sebagai memasukkan suatu *database* yang berisi data yang relevan untuk situasi dan dikelola oleh perangkat lunak yang disebut sistem manajemen *database* (DBMS). *Knowledge Base* berisi semua fakta, ide, hubungan dan interaksi suatu domain tertentu.
2. Subsistem manajemen basis pengetahuan bertugas untuk mendukung semua subsistem lain atau bertindak sebagai suatu komponen independen. Ia memberikan intelegensi untuk memperbesar pengetahuan pengambil keputusan.

3. Subsistem manajemen model Merupakan paket perangkat lunak yang memasukkan model keuangan statistik, ilmu manajemen atau model kuantitatif lainnya yang memberikan kapabilitas analitik dan manajemen perangkat lunak yang tepat.
4. Subsistem antar muka pengguna (dialog) untuk mengimplementasikan sistem kedalam program aplikasi sehingga pengguna atau pemakai dapat dengan sistem yang dirancang.

2.2.7 Tujuan Sistem Pendukung Keputusan

Adapun tujuan Sistem Pendukung Keputusan (*Decision Support System*) adalah (Dwi Citra Hartini, 2014):

1. Membantu manajer dalam pengambilan keputusan atas masalah-masalah semistruktur.
2. Memberikan dukungan atas pertimbangan manager dan bukannya dimaksudkan untuk menggantikan fungsi manager.
3. Meningkatkan efektifitas keputusan yang diambil manager lebih daripada perbaikan efisiennya.
4. Kecepatan komputasi. Komputer memungkinkan pengambil keputusan untuk melakukan banyak komputasi secara cepat dengan biaya yang rendah.
5. Peningkatan produktifitas. Membangun suatu kelompok dan memungkinkan para anggotanya untuk berada di berbagai lokasi yang berbeda-beda (menghemat biaya perjalanan). Selain itu produktifitas staf pendukung (misalnya analisis keuangan dan hukum) bisa ditingkatkan. Produktifitas juga bisa ditingkatkan menggunakan peralatan optimasi yang menentukan cara terbaik untuk menjalankan sebuah bisnis.

6. Dukungan kualitas. Komputer bisa meningkatkan kualitas yang dibuat. Sebagai contoh, semakin banyak data yang diakses, maka semakin banyak juga alternatif yang bisa dievaluasi. Analisis resiko bisa dilakukan dengan cepat dan pandangan dari para pakar (beberapa dari mereka berada dilokasi jauh) bisa dikumpulkan dengan cepat dan dengan biaya yang lebih rendah. Keahlian bahkan bisa diambil langsung dari sebuah sistem melalui metode kecerdasan buatan. Dengan komputer, para pengambil keputusan bisa melakukan evaluasi yang kompleks, memeriksa banyak skenario yang memungkinkan, dan menilai berbagai pengaruh secara cepat dan ekonomis. Semua kapabilitas tersebut mengarah kepada keputusan yang lebih baik.
7. Berdaya saing. Manajemen dan pemberdayaan sumber daya perusahaan. Tekanan persaingan menyebabkan tugas pengambilan keputusan menjadi sulit. Persaingan didasarkan tidak hanya pada harga, tetapi juga kualitas, kecepatan, kustomasi produk, dan dukungan pelanggan. Organisasi harus mampu secara sering dan cepat mengubah mode operasi, merekayasa ulang proses dan struktur, memberdayakan Calon Penerima, serta berinovasi. Teknologi pengambilan keputusan bisa menciptakan pemberdayaan yang signifikan dengan cara memperbolehkan seorang untuk membuat keputusan yang baik secara cepat, bahkan jika mereka memiliki pengetahuan yang kurang.
8. Mengatasi keterbatasan kognitif dalam pemrosesan dan penyimpanan. Otak manusia memiliki kemampuan yang terbatas untuk memproses dan menyimpan informasi. Orang-orang kadang sulit mengingat dan menggunakan sebuah informasi dengan cara yang bebas dari kesalahan.

2.3 Metode *Additive Ratio Assessment* (ARAS)

Pertama kali tahun 2010 Zavadskas dan Turskis membuat metode ARAS, yang bertujuan untuk memilih alternatif terbaik berdasarkan jumlah atribut dan peringkat akhir dari alternatif dibuat dengan menentukan tingkat utilitas setiap alternatif (Asep Supriatna, 2021).

Adapun langkah-langkah dari metode *Additive Ratio Assessment* (ARAS) sebagai berikut (Liza Handayani, 2019):

1. Pembentukan *Decision Making Matrix*

$$X = \begin{pmatrix} \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix} \quad i = \overline{0, m}; \quad j = \overline{1, n}$$

dimana :

m = jumlah alternatif

n = jumlah kriteria

x_{ij} = nilai performa dari alternatif i terhadap kriteria j x_{0j} = nilai optimum dari kriteria j

2. Penormalisasian *Decision Making Matrix* untuk semua kriteria

$$\overline{X} = \begin{pmatrix} \overline{x_{11}} & \dots & \overline{x_{1n}} & \dots & \overline{x_{1m}} \\ \overline{x_{21}} & \dots & \overline{x_{2n}} & \dots & \overline{x_{2m}} \\ \overline{x_{31}} & \dots & \overline{x_{3n}} & \dots & \overline{x_{3m}} \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix} \quad i = \overline{0, m}; \quad j = \overline{1, n}$$

Jika pada kriteria yang diusulkan bernilai maksimum, maka normalisasinya adalah:

$$\overline{x_{ij}} = \frac{x_{ij}}{\Sigma}$$

3. Menentukan bobot matriks yang sudah dinormalisasi pada tahap 2.

$$= 1$$

$$\left(\begin{array}{cccc} \overline{w_{01}} & \dots & \overline{w_{0m}} & \dots & \overline{w_{0n}} \\ \overline{w_{10}} & \dots & \overline{w_{1m}} & \dots & \overline{w_{1n}} \\ \dots & \dots & \dots & \dots & \dots \\ \overline{w_{m0}} & \dots & \overline{w_{mm}} & \dots & \overline{w_{mn}} \\ \dots & \dots & \dots & \dots & \dots \end{array} \right) \quad i = \overline{0, m}; \quad j = \overline{1, n}$$

4. Menentukan nilai dari fungsi optimum.

$$= \sum_{i = \overline{0, m}}$$

Dimana $\overline{w_{ij}}$ adalah nilai fungsi optimum alternatif i. Nilai terbesar adalah yang terbaik, dan nilai yang paling sedikit adalah yang terburuk. Dengan memperhitungkan proses, hubungan proporsional dengan nilai dan bobot kriteria yang diteliti berpengaruh pada hasil akhir.

5. Menentukan tingkatan peringkat.

$$= \overline{w_{ij}}; \quad i = \overline{0, m}$$

Dimana $\overline{w_{ij}}$ dan 0 merupakan nilai kriteria optimalitas, diperoleh dari persamaan. Sudah jelas, itu dihitung nilai $\overline{w_{ij}}$ berada pada interval [0,1] dan merupakan pesanan yang diinginkan didahulukan efisiensi relatif kompleks dari alternatif yang layak bisa ditentukan sesuai dengan nilai fungsi utilitas.

2.4 Unified Modelling Language (UML)

UML (*Unified Modeling Language*) adalah metode pemodelan secara visual sebagai sarana untuk merancang dan membuat *software* berorientasi objek. Karena UML ini merupakan bahasa visual untuk pemodelan bahasa berorientasi objek. UML adalah salah satu *tool/model* untuk merancang pengembangan *software* yang berbasis *object oriented*. Dalam kerangka spesifikasi, *Unified*

Modeling Language (UML) menyediakan model-model yang tepat, tidak mendua arti (ambigu) serta lengkap.

Sejarah UML sendiri cukup panjang. Sampai era tahun 1990 seperti kita ketahui puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah metodologi Booch, metodologi Coad, metodologi OOSE, metodologi OMT, metodologi Shlaer-Mellor, metodologi Wirfs-Brock, dan sebagainya. Masa itu terkenal dengan masa perang metodologi (*method war*) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerja sama dengan grup/perusahaan lain yang menggunakan metodologi yang berlainan.

UML menyediakan beberapa *notasi* dan *artifact standart* yang bisa digunakan sebagai alat komunikasi bagi para pelaku dalam proses analisis dan desain. *Artifact* didalam UML didefinisikan sebagai informasi dalam bentuk yang digunakan atau dihasilkan dalam proses pengembangan perangkat. Contohnya adalah *source code* yang dihasilkan oleh proses pemrograman. Yang harus diperhatikan untuk menjaga konsistensi antar artifact selama proses analisis.

2.4.1 Kegunaan UML

Adapun manfaat dari penggunaan *Unified Modeling Language* adalah sebagai berikut :

1. UML sebagai bahasa visualisasi yang digunakan untuk merancang suatu model yang dapat dibaca oleh banyak orang dengan pengertian yang sama.
2. UML merupakan bahasa pendefenisian dengan perincian seluruh hasil analisa, desain dan implementasi yang harus dilakukan dalam pengembangan sistem.

3. UML sebagai bahasa dokumentasi yang digunakan untuk mendokumentasikan arsitektur beserta perinciannya.

2.4.2 Tujuan Penggunaan *Unified Modeling Language*

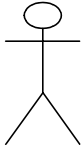
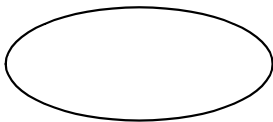

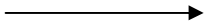
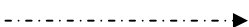
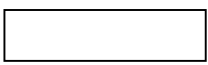
Adapun tujuan dari penggunaan *Unified Modeling Language* adalah sebagai berikut :

1. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
2. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.
3. Memberikan model yang siap pakai, bahasa pemodelan visual yang *ekspresif* untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
4. UML bisa juga berfungsi sebagai sebuah (*blue print*) cetak biru karena sangat lengkap dan detail. Dengan cetak biru ini maka akan bisa diketahui informasi secara detail tentang *coding* program atau bahkan membaca program dan merepresentasikan kembali ke dalam bentuk diagram (*reverse engineering*).

2.4.3 Use Case

Use Case diagram melukiskan perilaku suatu sistem yang menghasilkan nilai terukur terhadap aktor. *Use Case* diagram menggambarkan segala sesuatu yang diinginkan oleh aktor agar dikerjakan oleh sistem (kebutuhan sistem dari sudut pandang user).

Tabel 2.1 Simbol *Use Case Diagram*

No	Simbol	Notasi	Keterangan
1		<i>Actor</i>	Menggambarkan orang, sistem atau <i>external</i> entitas atau stakeholder yang menyediakan atau menerima informasi dari sistem.
2		<i>Use Case</i>	Menerangkan “apa” yang dikerjakan sistem, bukan “bagaimana” sistem mengerjakannya.
3		<i>Association</i>	Menggambarkan bagaimana <i>actor</i> terlibat dalam <i>use case</i>
4		<i>Generalization</i>	Dibuat ketika ada sebuah keadaan yang lain/perlakuan khusus
5		<i>Include</i>	Realsi <i>use case</i> dimana proses bersangkutan akan dijalankan ke proses yang dituju
6		<i>Sistem Boundary</i>	Batasan gambaran antara sistem dengan <i>actor</i>

Use case diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. *Use case diagram* terdiri atas diagram untuk *use case* dan *actor*.

2.4.4 Class Diagram

Class adalah spesifikasi yang akan menghasilkan objek dan merupakan inti dari pengembangan dan desain berorientasi objek (Tofik Isa, 2017).

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok antara lain yaitu adalah:

1. Nama (*stereotype*)
2. Atribut
3. Metoda

Atribut dan metoda dapat mewakili salah satu sifat berikut:

1. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan.
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya.
3. *Public*, dapat dipanggil oleh siapa saja.

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan dalam (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi) sesuai dengan perkembangan *class* model, *class* dapat dikelompokkan menjadi *package*. Kita juga dapat membuat diagram yang terdiri atas *package*.



Ada beberapa hubungan antar *class* dalam *class*, hubungan itu antara lain sebagai berikut:




1. *Asosiasi*, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.
2. *Agregasi*, yaitu hubungan yang menyatakan bagian (“terdiri atas”).
3. Pewarisan yaitu hubungan hirarkis antar *class* yang dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
4. Hubungan dinamis yaitu rangkaian pesan (*message*) yang di *passing* dari satu *class* kepada *class* lain.

2.4.5 Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainnya seperti *use case* atau interaksi.

Tabel 2.2 Tabel Simbol *Activity Diagram*

No	Gambar	Nama	Keterangan
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Activity</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi

No	Gambar	Nama	Keterangan
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

2.5 Bagan Alir (*Flowchart*)

Bagan alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi (Verawati, 2018).


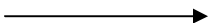

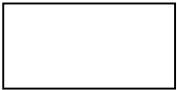
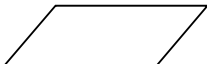

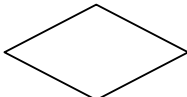

Tujuan utama dari penggunaan *Flowchart* ini adalah untuk menggambarkan suatu tahapan penyelesaian masalah yang sederhana, teratur, rapi dan jelas dengan menggunakan simbol-simbol yang standar. Tahap penyelesaian masalah yang disajikan harus jelas, sederhana, efektif dan tepat.

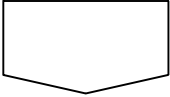
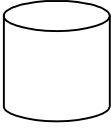

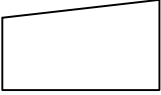
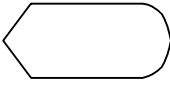

Gunakan simbol-simbol bagan alir yang standart. *Flowchart* disusun dengan simbol-simbol yang dipakai sebagai alat bantu menggambarkan proses didalam program. Simbol-simbol yang digunakan dibagi menjadi tiga kelompok, yaitu :

1. Simbol penghubung/alur (*Flow direction symbol*)
2. Simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan yang lainnya.
3. Simbol proses (*Procesing Symbols*)

4. Simbol yang menunjukkan jenis operasi pengolahan dalam suatu proses atau prosedur
5. Simbol input ouput
6. Simbol yang menunjukkan jenis peralatan yang digunakan sebagai media input–output. Berikut simbol-simbol dari *Flowchart* yang terdapat dibawah ini :

Tabel 2.3 Simbol *Flowchart*

No	Simbol	Nama	Fungsi
1		<i>Terminator</i>	Permulaan/akhir program
2		Garis Alir (<i>Flow Line</i>)	Arah aliran program atau merepresentasikan alur kerja
3		<i>Preparation</i>	Proses inisialisasi atau proses pemberian harga awal
4		Proses	Proses perhitungan atau proses pengolahan data
5		<i>Input/Output Data</i>	Proses input/output data, parameter, informasi
6		<i>Predefined process</i>	Permulaan sub program atau menjalankan sub program
7		<i>Decision</i> (Keputusan)	Perbandingan pernyataan yang memberikan pilihan untuk langkah selanjutnya
8		<i>On-page Connector</i>	Penghubung bagian-bagian <i>flowchart</i> yang berada pada satu halaman

No	Simbol	Nama	Fungsi
9		<i>Off-page Connector</i>	Penghubung bagian-bagian <i>flowchart</i> yang berada pada halaman berbeda
10		<i>Magnetic Disk</i>	Input/output yang menggunakan disk <i>magnetic</i>
11		<i>Magnetic Drum</i>	Input/output yang menggunakan drum <i>magnetic</i>
12		<i>Manual Input</i>	Input yang dimasukkan secara manual dari <i>keyboard</i>
13		<i>Display</i>	Output yang ditampilkan pada terminal
14		<i>Operation</i>	Operasi manual

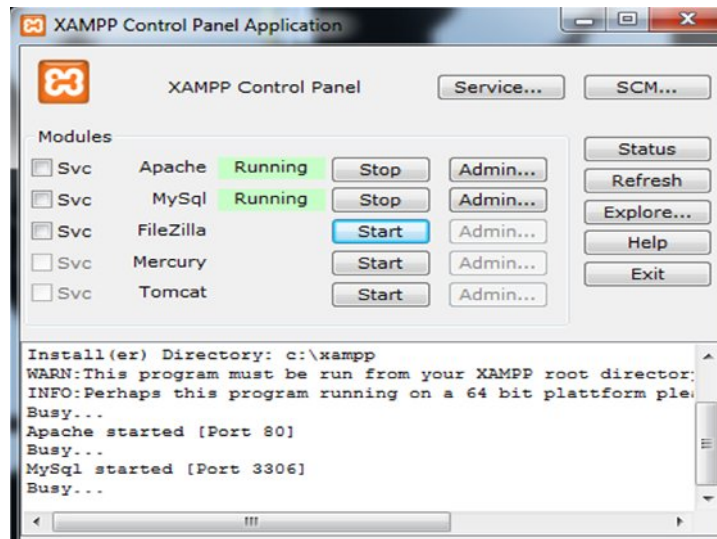
2.6 Aplikasi Pengembangan Sistem

Adapun aplikasi-aplikasi yang digunakan dalam pengembangan sistem yaitu sebagai berikut:

1. XAMPP

XAMPP adalah salah satu paket *software web server* yang didalamnya telah terdapat *Apache*, MySQL, PHP, dan phpMyAdmin. Proses instalasi XAMPP sangat mudah, karena tidak perlu melakukan konfigurasi *Apache*, PHP dan MySQL secara manual, XAMPP melakukan instalasi dan konfigurasi secara otomatis.

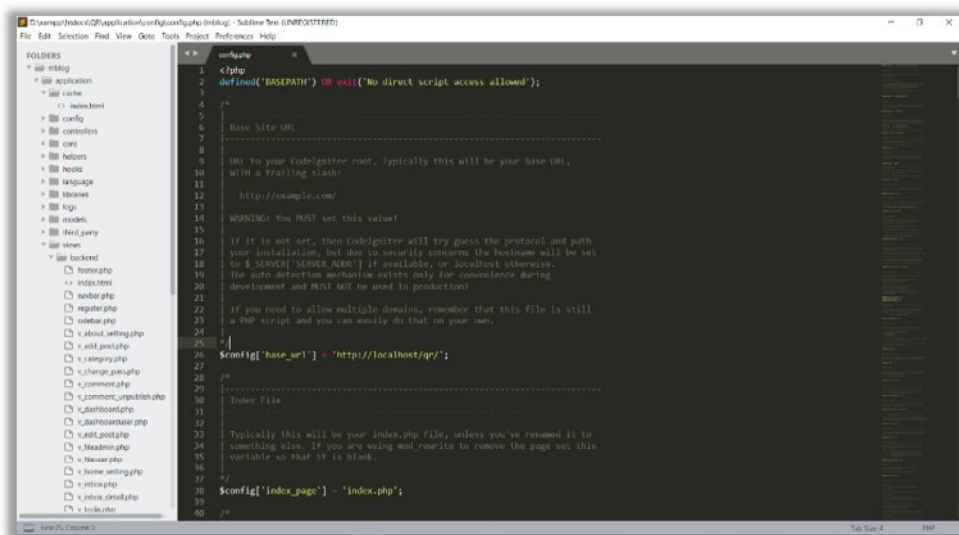
XAMPP adalah salah satu paket instalasi *apache*, PHP, dan MySQL secara instant yang dapat digunakan untuk membantu proses instalasi ketiga produk tersebut.



Gambar 2.1 Tampilan Untuk Mengaktifkan XAMPP

2. *Sublime Text*

Sublime Text adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai *platform operating system* dengan menggunakan teknologi *Phyton* API. Terciptanya aplikasi ini terinspirasi dari aplikasi Vim, Aplikasi ini sangatlah fleksibel dan *powerfull*. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan *sublime-packages*. Berikut ini adalah tampilan dari *Sublime Text*:



Gambar 2.2 Tampilan *Sublime Text*

3. MySQL

MySQL adalah salah satu jenis *database server* yang sangat terkenal, kepopulerannya disebabkan MySQL menggunakan *Structure Query Language* (SQL) sebagai dasar untuk mengakses basis datanya. Selain itu, MySQL bersifat *free* pada berbagai *platform* (kecuali pada Windows, yang bersifat *shareware* atau anda perlu membayar setelah melakukan evaluasi dan memutuskan untuk digunakan untuk keperluan produksi) atau tidak dicekal. MySQL termasuk jenis *Relational Database Management Sistem* (RDBMS). Itulah sebabnya istilah seperti tabel, baris dan kolom digunakan di dalam MySQL. Sebuah basis data mengandung satu atau sejumlah tabel, tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom.

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi diseluruh dunia. *MySQL AB* membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi *GNU General*

Public License (*GPL*), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan *GPL*. MySQL merupakan bahasa pemrograman *open-source* yang paling populer dan banyak digunakan di lingkungan Linux kepopuleran ini karena ditunjang oleh performansi *query* dari *database*-nya yang jarang bermasalah.

Kesimpulannya MySQL adalah singkatan dari *My Sequel* yang bisa didefinisikan sebuah software atau perangkat lunak dengan sistem manajemen basis data SQL atau DBMS yang multithread dan multi-user dengan jumlah instalasi sekitar 6 juta diseluruh dunia. Sederhananya MySQL merupakan sebuah perangkat lunak pada *Relational Database Mnagement Sytem* (RDMS) didalam sebuah manajemen *database* sebagai basis data.