



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1 Pendidikan

Pendidikan pendidikan berasal dari kata “*paedagogie*” dari bahasa Yunani, terdiri dari kata “*paes*” artinya anak dan “*agogos*” artinya membimbing. Jadi *paedagogie* berarti bimbingan yang diberikan kepada anak. Dan menurut kamus besar Bahasa Indonesia menyatakan bahwa Pendidikan adalah proses pengubahan sikap dan tata laku seseorang atau kelompok orang dalam usaha mendewasakan manusia melalui upaya pengajaran dan latihan, proses perbuatan, cara mendidik. Dapat disimpulkan bahwa pendidikan adalah usaha sadar dan terencana untuk memberikan bimbingan atau dukungan dalam pengembangan potensi fisik dan mental yang diberikan oleh orang dewasa pada siswa untuk mencapai kedewasaan serta mencapai tujuan agar peserta didik mampu melaksanakan tugas hidupnya secara mandiri [2].

UMMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA

2.2 Guru atau Pendidik

Salah satu komponen proses Pendidikan adalah pendidik atau Guru. Secara umum seorang guru adalah seorang pendidik yang bertanggung jawab atas pertumbuhan dan perkembangan seorang siswa termasuk aspek mental (kognitif dan emosional) dan fisik dan kemudian memerintahkan siswa untuk maju dan mengikuti ajaran guru dengan keyakinan yang sesuai [2]. Dan menurut UU RI No. 14 tahun 2005 mengartikan dosen dan guru memiliki tugas utama mendidik, mengajar, membimbing, mengarahkan, melatih, menguji, dan mengevaluasi siswa prasekolah pada sistem pendidikan formal, pendidikan menengah pertama, dan pendidikan dasar.

2.3 Penilaian Kinerja Karyawan / Pegawai

Penilaian Karyawan / Pegawai pada dasarnya adalah upaya penilaian terhadap kinerja karyawan. Secara umum hal ini dapat dipahami sebagai upaya untuk mengukur kinerja setiap karyawan di perusahaan. Hal ini berkaitan dengan produktivitas pekerja dan efisiensi kerja dalam melakukan pekerjaan tertentu sesuai dengan uraian tugas yang diberikan perusahaan kepada pekerja yang bersangkutan. Selain itu hasil pengukuran kinerja karyawan atau evaluasi karyawan ini pada umumnya akan digunakan untuk meninjau upaya perusahaan dalam meningkatkan produktivitas dan efisiensi yang dilakukan secara berkesinambungan [6].

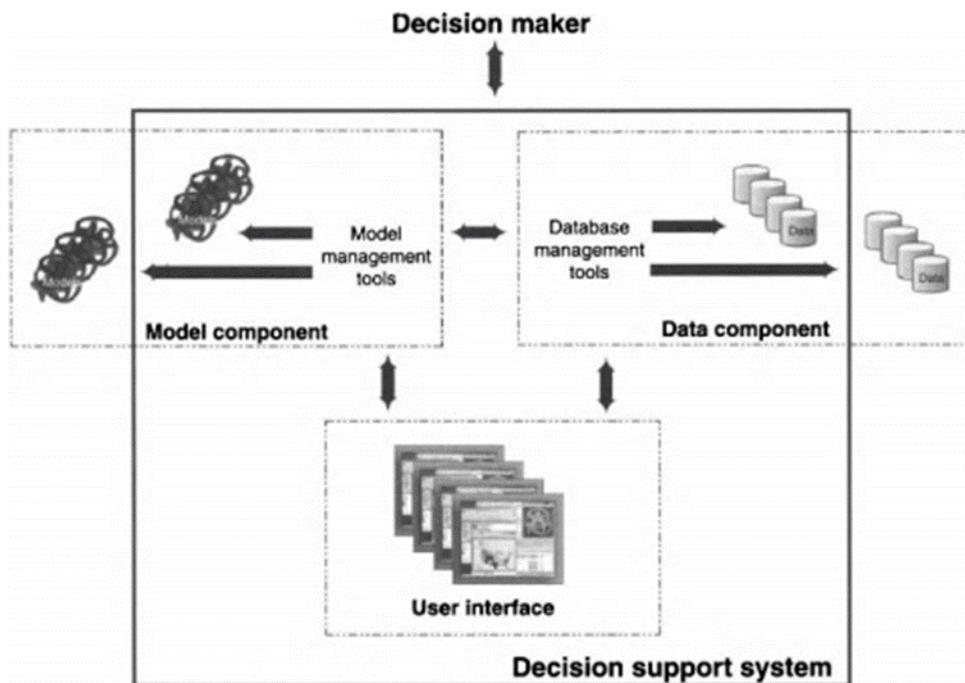
2.4 Pengambilan keputusan (*Decision Making*)

Robins berpendapat bahwa "*decision making is which on choses between two or more alternative*". Berdasarkan pendapat di atas, dapat dipahami bahwa esensi dari pengambilan keputusan adalah memilih dua atau lebih pilihan untuk melakukan tindakan tertentu secara individu atau kelompok. Keputusan adalah proses memilih tindakan tertentu dari serangkaian kemungkinan tindakan alternatif. Pengambilan keputusan merupakan upaya untuk menciptakan suatu peristiwa dan membentuk masa depan (pada saat pemilihan dan peristiwa selanjutnya) [13].

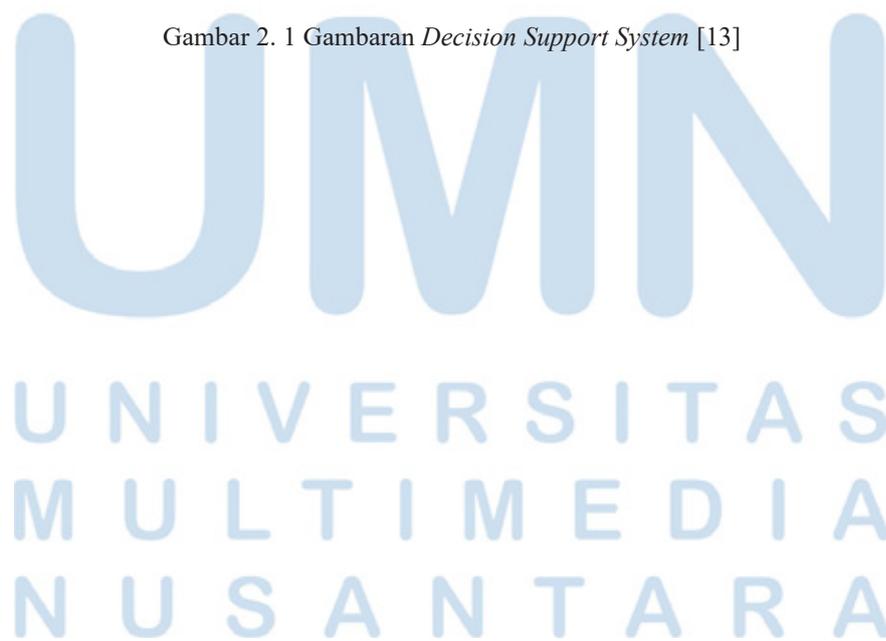
2.5 Sistem Pendukung Keputusan (*Decision Support System* atau DSS)

Sistem Pendukung Keputusan adalah Sistem informasi terkomputerisasi yang mengambil pendekatan untuk membuat berbagai alternatif keputusan untuk membantu pihak-pihak tertentu dalam mengatasi masalah dengan menggunakan data dan model. Pengambilan keputusan merupakan hasil dari proses memilih dari berbagai alternatif tindakan yang dapat dipilih melalui mekanisme tertentu dalam rangka pengambilan keputusan yang terbaik. DSS (*Decision Support System*) hanya menyediakan keputusan alternatif dan disajikan kepada pengguna untuk pengambilan keputusan. [14].

Komponen-komponen Sistem Pendukung Keputusan terdiri dari 3 komponen utama atau subsistem yaitu *Subsistem Data (Database)*, *Subsistem Model (Model Subsistem)*, *Subsistem Dialog (User Sistem Interface)* [15].



Gambar 2. 1 Gambaran *Decision Support System* [13]



2.6 *Multiple Attribute Decision Making (MADM)*

Multiple attribute decision making (MADM) mengacu pada pembuatan keputusan dihadapi beberapa, biasanya bertentangan, atribut. masalah untuk pengambilan keputusan beberapa atribut adalah kejadian umum di setiap aspek. Setiap masalah memiliki banyak atribut. Sebuah pengambil keputusan harus menghasilkan atribut yang relevan untuk setiap masalah pengaturan. kalimat "atribut" dapat disebut sebagai "sasaran" atau "Kriteria. Jumlah atribut bisa antara beberapa sampai sekitar 350. Misalnya, seseorang dapat menggunakan harga, jarak tempuh bensin, keamanan, ruang kaki, pengerjaan, dan gaya untuk mengevaluasi mobil, Meskipun mungkin ada lebih dari 100 faktor yang harus dipertimbangkan saat memilih lokasi untuk pabrik [16].

2.7 *Simple Additive Weighting (SAW)*

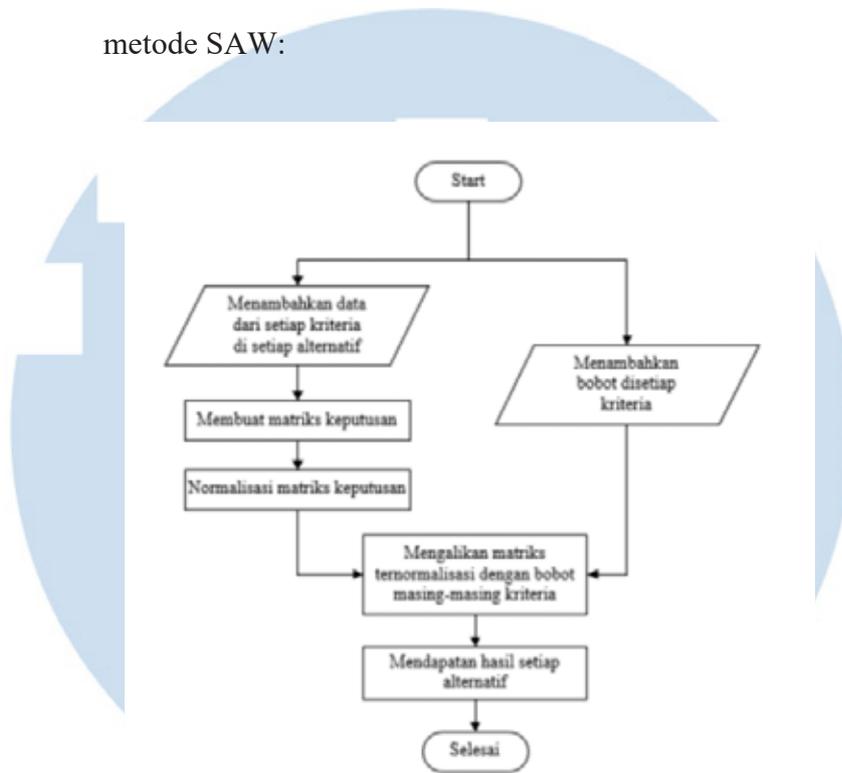
Konsep dasar metode SAW adalah mencari jumlah bobot evaluasi kinerja setiap pilihan untuk semua kriteria. Metode SAW membutuhkan proses normalisasi matriks keputusan dalam skala yang dapat dibandingkan dengan semua alternatif evaluasi yang ada. Metode ini mengharuskan pengambil keputusan untuk menentukan bobot masing-masing atribut. Skor total alternatif diperoleh dengan menjumlahkan semua hasil perkalian antara skor dan bobot masing-masing atribut. Metode SAW memiliki dua atribut yaitu *cost* dan *benefit*. [17].

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

Benefit adalah kriteria yang menguntungkan untuk perhitungan dimana semaking tinggi nilainya semakin baik untuk penilaian dan *cost* adalah kriteria yang merugikan untuk perhitungan dimana semakin tinggi nilainya semakin buruk untuk penilaiannya [17]. Jadi pemilihan *cost* dan *benefit* tergantung dari kriterianya ada berapa *cost* dan *benefitnya*, namun pada perhitungan SAW harus memiliki kriteria *cost* dikarenakan jika semua kriteria adalah *benefit* akan memberikan hasil peringkat yang lebih besar sehingga untuk proses peringkat menggunakan SAW diharuskan memiliki kriteria *benefit* dan *cost* [28]. Contoh dalam penilaian kinerja guru untuk kriteria kehadiran merupakan kriteria *benefit* dikarenakan semakin tinggi nilai kehadiran semakin baik untuk penilaian kinerja sedangkan kriteria keterlambatan semakin tinggi nilainya semakin buruk untuk penilaian kinerja sehingga disebut kriteria *cost*. Kemudian sebelum perhitungan SAW diperlukannya bobot dari tiap kriteria, pembobotan kriteria dalam metode SAW ditentukan oleh yang akan mengambil keputusan dan disesuaikan dengan kebutuhan [29].

Konsep dasar SAW adalah mencari penjumlahan terbobot dari rating kinerja ternormalisasi (R) pada setiap alternatif pada semua bobot atribut. Metode SAW membutuhkan proses normalisasi matriks ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada [17].

Berikut merupakan gambaran langkah – langkah dari proses metode SAW:



Gambar 2. 2 Flowchart tahapan proses SAW [17]

1) Pertama menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu C_i . Berikut merupakan contoh soal yang telah dibuat:

Nama	IPK	Penghasilan Ortu	Jml Tanggungan	Rumah(km)
Tania	3,62	2.500.000	1	30
Leo	3,45	4.000.000	2	77
Monika	4,00	3.500.000	2	100
Salim	3,20	6.500.000	3	52

Gambar 2. 3 Contoh soal metode SAW yang dibuat

2) Lalu dilakukannya pembagian *cost* dan *benefit* untuk kriterianya, kriteria *benefit*nya adalah IPK, penghasilan orang tua perbulan, dan tanggungan sedangkan untuk kriteria *cost*nya adalah lokasi rumah. Dengan

pembobotan IPK 30%, lokasi rumah 20%, Jml Tanggungan 25%, dan Penghasilan orang tua per bulan 25%.

3) Membuat matriks keputusan berdasarkan kriteria (C_i) kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut *benefit* ataupun atribut *cost*) sehingga diperoleh matriks ternormalisasi R.

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\max_i x_{ij}} & \text{Jika } j \text{ merupakan atribut } \textit{benefit} \\ \frac{\min_i x_{ij}}{x_{ij}} & \text{Jika } j \text{ merupakan atribut } \textit{cost} \end{cases} \quad (2.1)$$

Rumus 2. 1 Rumus Normalisasi SAW

Berdasarkan rumus 2.1 yang merupakan rumus normalisasi SAW maka untuk kriteria *benefit* nilainya akan dibagi dengan nilai tertinggi didalam dan untuk kriteria *cost* nilainya akan dibagi dengan nilai terendah dari kriteria tersebut.

Berikut hasil normalisasi berdasarkan rumus 2.1

Nama	IPK	Penghasilan Ortu (Rp)	Jml Tanggungan	Rumah(km)
Tania	3,62 / 4,00 = 0,905	2500000 / 6500000 = 0,384	1 / 3 = 0,333	30 / 30 = 1
Leo	3,45 / 4,00 = 0,8625	4000000 / 6500000 = 0,615	2 / 3 = 0,666	30 / 77 = 0,389
Monika	4,00 / 4,00 = 1	3500000 / 6500000 = 0,538	2 / 3 = 0,666	30 / 100 = 0,3
Salim	3,2 / 4,00 = 0,8	6500000 / 6500000 = 1	3 / 3 = 1	30 / 52 = 0,576

Gambar 2. 4 Hasil Normalisasi

4) Hasil akhir diperoleh dari proses pengurutan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan vektor bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik (A_i) sebagai solusi.

Berikut proses dan rumus perangkkingan dari hasil normalisasi

$$V_i = \sum_{j=1}^n W_j r_{ij} \quad (2.2)$$

Rumus 2. 2 Rumus Peringkat SAW

Hasil perhitungan dari rumus peringkat SAW

$$V_{Tania}: (0.905 \times 30\%) + (0.384 \times 25\%) + (0.3333 \times 25\%) + (1 \times 20\%) = 0.650825 \quad (4)$$

$$V_{Leo}: (0.8625 \times 30\%) + (0.615 \times 25\%) + (0.6666 \times 25\%) + (0.389 \times 20\%) = 0.65695 \quad (3)$$

$$V_{Monika}: (1 \times 30\%) + (0.538 \times 25\%) + (0.6666 \times 25\%) + (0.3 \times 20\%) = 0.66115 \quad (2)$$

$$V_{Salim}: (0.8 \times 30\%) + (1 \times 25\%) + (1 \times 25\%) + (0.576 \times 20\%) = 0.8552 \quad (1)$$

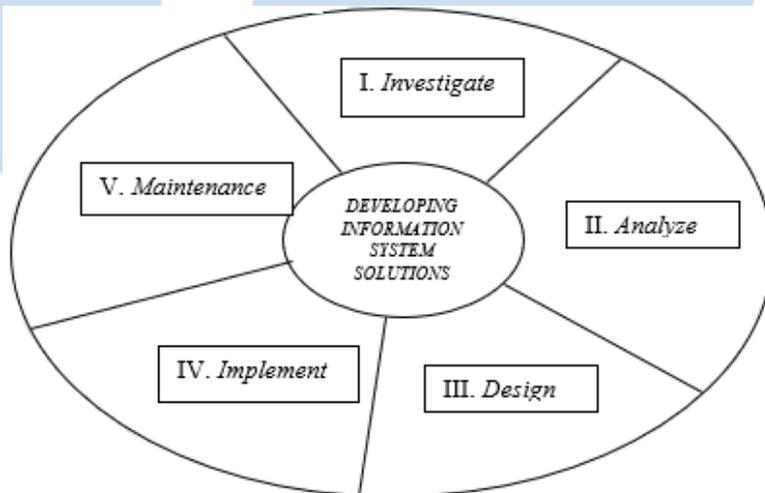
U M N

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

2.8 Software Development Lifecycle (SDLC)

Software development lifecycle (SDLC) adalah merupakan gambaran atau perencanaan dari suatu usaha atau kegiatan yang bertujuan membuat sistem yang dapat berjalan seperti roda berputar, yang melalui beberapa tahapan umum seperti menyelidiki, analisis, desain, implementasi, dan *maintenance* [18].

Berikut merupakan gambaran dari SDLC:



Gambar 2. 5 Tahapan SDLC [18]

2.9 Model Prototype

Metode *prototyping* adalah metode pengembangan *software* yang berupa model fisik dari bagaimana sistem berfungsi sebagai versi awal *software*. Metode *prototype* akan menghasilkan *prototype* dari sistem sebagai perantara pengembangan *software* dan agar pengguna dapat berinteraksi langsung dalam proses pengembangan *software* [19].

Berikut gambaran dari tahapan model *prototype* yang dipakai dalam penelitian ini [20]



Gambar 2. 6 Tahapan SDLC metode *Prototype* [20]

Penjelasan singkat dari langkah – langkah model *Prototype* [20]

1. *Planning*

Langkah pertama adalah *planning*. Pada tahap ini akan memutuskan sistem seperti apa yang akan dibuat dan apa saja yang diperlukan agar proses *development* dapat berhasil dilakukan [20].

2. *Analysis*

Setelah *Planning*, tahap selanjutnya adalah *Analysis* dimana pada tahap ini dilakukannya analisis mengenai permasalahan yang dialami pengguna dan mengumpulkan informasi mengenai fitur apa saja yang perlu dibuat agar sistem dapat menyelesaikan permasalahan yang ditemukan [20].

3. Design

Pembuatan desain dilakukan berdasarkan hasil *Analysis* [20], desain sistem dibuat menggunakan beberapa *Unified Modeling Language* (UML) seperti *Activity Diagram*, *Class Diagram*, dan *Use Case Diagram*

4. System Prototype

Model sistem dipersiapkan dan fase *coding* dilakukan pada fase ini. Seluruh sistem akan dirancang berdasarkan fitur dan kebutuhan sistem yang didapat dari hasil fase *Analysis*. Pada tahap ini sistem yang dibuat sudah dapat dipakai sesuai dengan kegunaan dan kebutuhan dari pengguna [20].

5. Test

Di fase ini sistem dicoba pada perangkat keras yang telah dipilih berdasarkan fungsionalitas yang telah dibuat. Interaksi sistem dengan sistem lain pada perangkat keras diperiksa agar sistem dapat berjalan sesuai dengan kebutuhan atau tidak [20].

U M I N
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

2.10 Unified Modelling Language (UML)

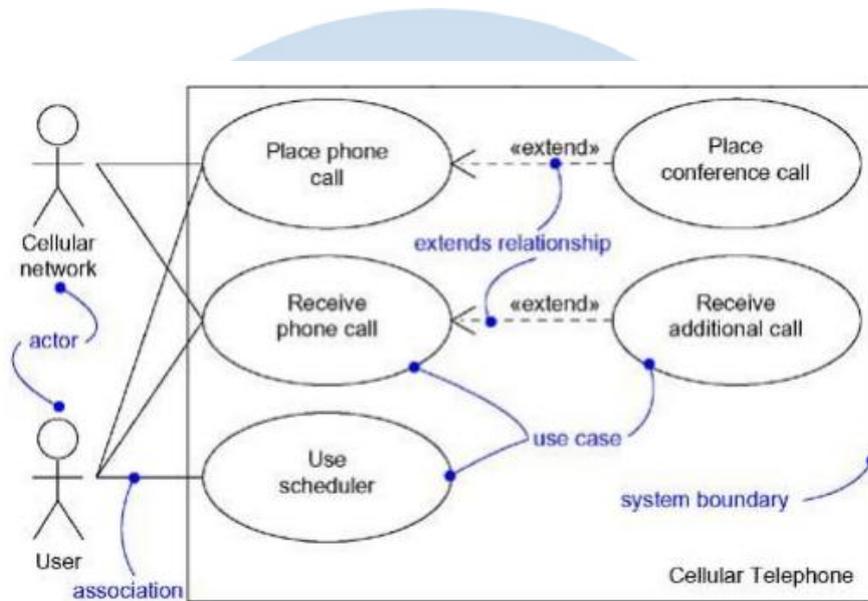
Unified Modeling Language (UML) adalah Bahasa standar untuk membuat gambar desain perangkat lunak. UML dapat digunakan untuk memvisualisasikan, mendefinisikan, membangun, dan mendokumentasikan fungsionalitas sistem. UML adalah bahasa yang sangat ekspresif yang menangani semua tampilan yang diperlukan untuk mengembangkan dan mengimplementasikan sistem UML. Ada tiga komponen utama untuk ini. Komponen dasar UML, aturan yang menentukan bagaimana komponen tersebut digabungkan, dan beberapa mekanisme umum yang berlaku untuk seluruh bahasa. [21].

2.11 Use Case Diagram

Use case diagram adalah salah satu dari lima diagram di UML untuk memodelkan aspek dinamis sistem. *Use case diagram* adalah pusat pemodelan perilaku sistem, subsistem, atau kelas. Masing-masing menunjukkan satu set kasus penggunaan dan aktor serta hubungannya.

Use case diagram penting untuk memvisualisasikan, mendefinisikan, dan mendokumentasikan perilaku elemen. Agar sistem, subsistem, dan kelas dapat diakses dan dimengerti dengan memberikan pandangan eksternal tentang bagaimana elemen-elemen ini dapat digunakan dalam konteks [21].

Berikut gambar dari penggunaan *use case* diagram:



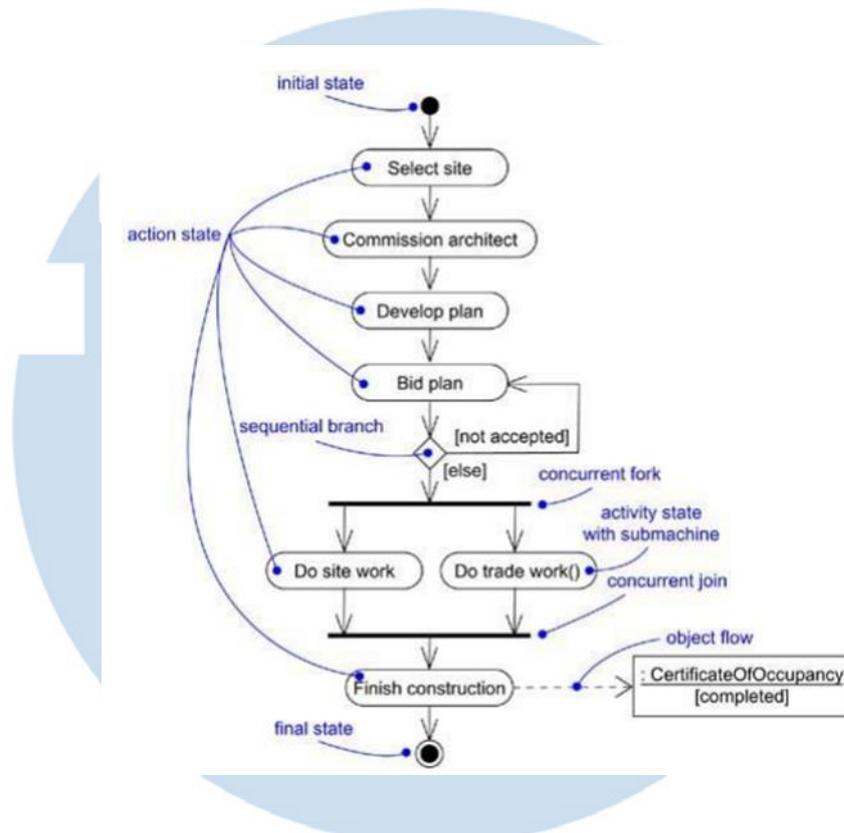
Gambar 2. 7 Contoh penggunaan *Use Case Diagram* [21]

2.12 Activity Diagram

Activity diagram adalah salah satu dari lima diagram di UML untuk pemodelan aspek dinamis sistem. *Activity diagram* pada dasarnya adalah diagram alur, yang menunjukkan aliran kendali dari aktivitas ke aktivitas. *Activity diagram* berguna untuk memodelkan aspek dinamis dari suatu sistem. *Activity diagram* dapat memvisualisasikan, menentukan, membangun, dan mendokumentasikan dinamika suatu objek, atau dapat digunakan untuk memodelkan aliran kendali suatu operasi [21].

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

Berikut contoh penggunaan *activity diagram*



Gambar 2. 8 Contoh penggunaan *Activity Diagram* [21]

Activity diagram menunjukkan aliran dari aktivitas ke aktivitas.

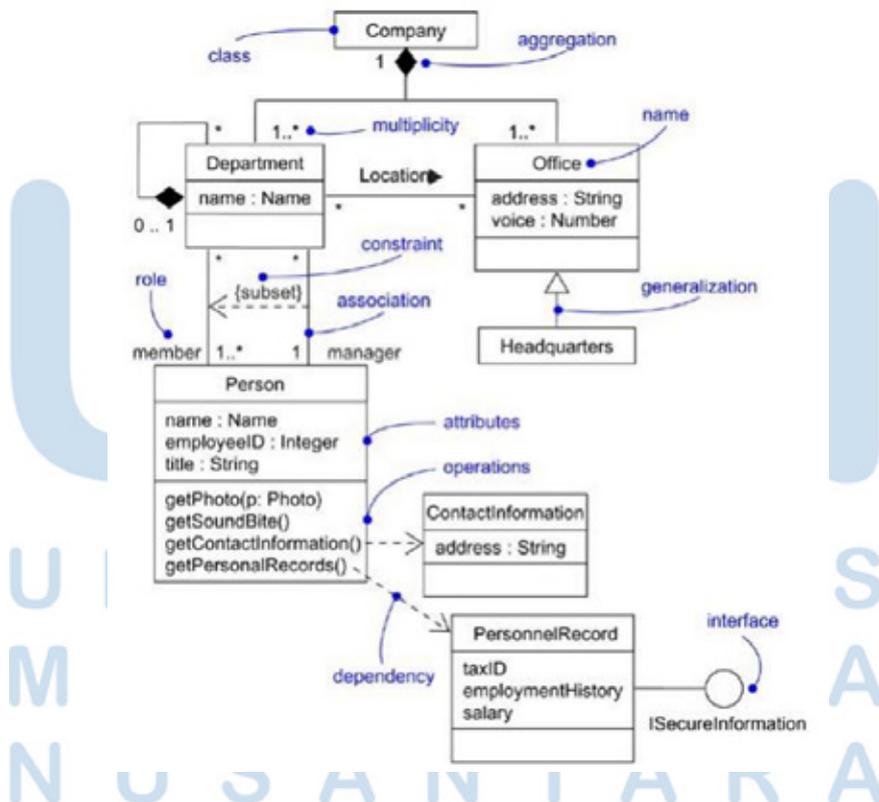
Aktivitas pada akhirnya menghasilkan tindakan yang terdiri dari file yang dapat dieksekusi secara komputasi yang menghasilkan perubahan status sistem atau pengembalian nilai. Tindakan termasuk menjalankan operasi lain, mengirim sinyal, membuat atau menghancurkan objek. [21].

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

2.13 Class Diagram

Class diagram adalah diagram yang paling umum digunakan dalam pemodelan sistem berorientasi objek. *Class diagram* menunjukkan satu set kelas, antarmuka, kolaborasi, dan hubungan mereka. Diagram kelas digunakan untuk memodelkan tampilan desain statis sistem. Ini termasuk pemodelan kosakata sistem, pemodelan kolaboratif, atau pemodelan skema. *Class diagram* digunakan tidak hanya untuk memvisualisasikan, mendefinisikan, dan mendokumentasikan model struktural, tetapi juga untuk membuat sistem yang dapat dieksekusi melalui rekayasa maju dan mundur. [21]

Berikut gambar penggunaan *Class diagram*.



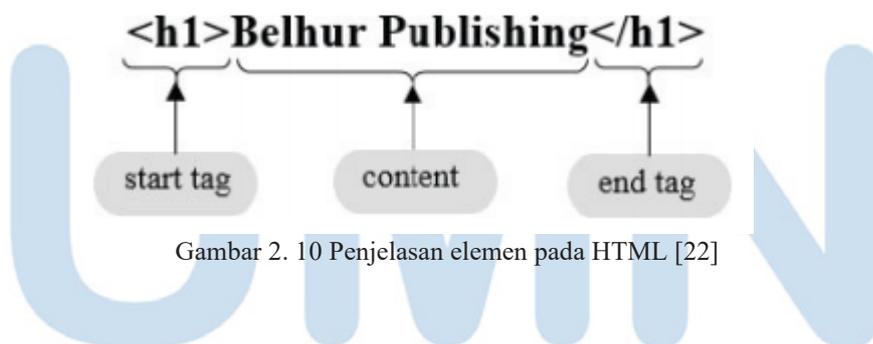
Gambar 2. 9 Contoh penggunaan *Class Diagram* [23]

2.14 HTML

HTML adalah singkatan dari *Hypertext Markup Language*. *Hypertext* mengacu pada kemampuan untuk membuat tautan ke halaman *Web* lain. *Markup* artinya digunakan untuk membuat halaman dengan teks yang diformat, gambar, dan sumber daya lain yang disematkan di halaman. HTML dibuat oleh Tim Berners-Lee pada tahun 1990. Pengkodean HTML didasarkan pada sekumpulan simbol markup yang ditempatkan di HTML dokumen atau halaman *Web*. Simbol *markup* ini mengidentifikasi elemen struktural atau *tag* yang memberi tahu *browser Web* (dan agen pengguna lainnya) cara merender dan menampilkan halaman *Web*. Halaman *Web* adalah file teks biasa yang hanya terdiri dari teks dan elemen HTML [22].

TagName adalah nama elemen, dan konten adalah konten elemen.

Perhatikan contoh berikut:



Gambar 2. 10 Penjelasan elemen pada HTML [22]

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

2.15 CSS

CSS adalah bahasa sederhana yang mendefinisikan konstruksi gaya untuk memformat tampilan halaman *Web*. CSS adalah bahasa presentasi, berbeda dengan HTML, yang merupakan bahasa semantik. Dengan kata lain, Anda tidak akan menemukan pernyataan *loop* atau fungsi di CSS. Ini adalah bahasa pelengkap yang sempurna untuk HTML dan dirancang terutama untuk bekerja dengan HTML.

CSS berisi dua bagian: selektor dan deklarasi. Selektor dapat berupa tag HTML apa pun (seperti *p*, *h1*, atau *div*) atau pemilih yang ditentukan pengguna seperti *class* atau *id*. Kita akan membahas pemilih kelas dan *id* nanti di bab ini. Deklarasi adalah daftar pasangan nilai properti yang terkait dengan pemilih. Properti adalah konstruksi apa pun seperti ukuran *font*, warna, atau posisi dengan nilai masing-masing seperti 12 piksel, merah, atau relatif [22].



2.16 PHP

PHP dapat digunakan untuk melakukan apa saja yang dapat dilakukan oleh program *Common Gateway Interface* lainnya, termasuk mengambil dan menggunakan data masukan, membuat konten halaman dinamis, dan memperbarui data yang disimpan di *server*. Salah satu fitur PHP yang paling signifikan adalah dukungannya untuk berbagai macam *database*. Menulis aplikasi *Website* dinamis yang dapat mengambil dan menyimpan data sangat mudah dilakukan dengan PHP.

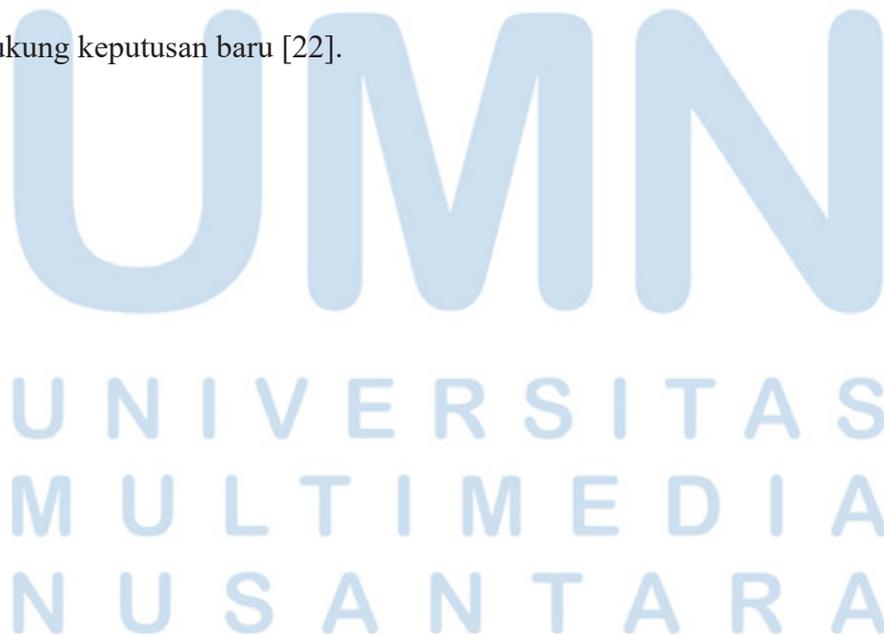
Dukungan PHP terhadap standar *Open Database Connectivity* (ODBC) berarti dapat digunakan untuk menyambungkan ke *database* apa pun yang mendukung ODBC, termasuk *Microsoft Access* dan *SQL Server*. PHP dapat digunakan untuk menyelesaikan kebutuhan aplikasi bisnis, meramaikan aplikasi yang ada, membuat antarmuka pengguna, atau membuat aplikasi mudah diakses oleh pengguna jarak jauh tanpa memerlukan perangkat lunak atau perangkat keras tambahan [22].



2.17 Database

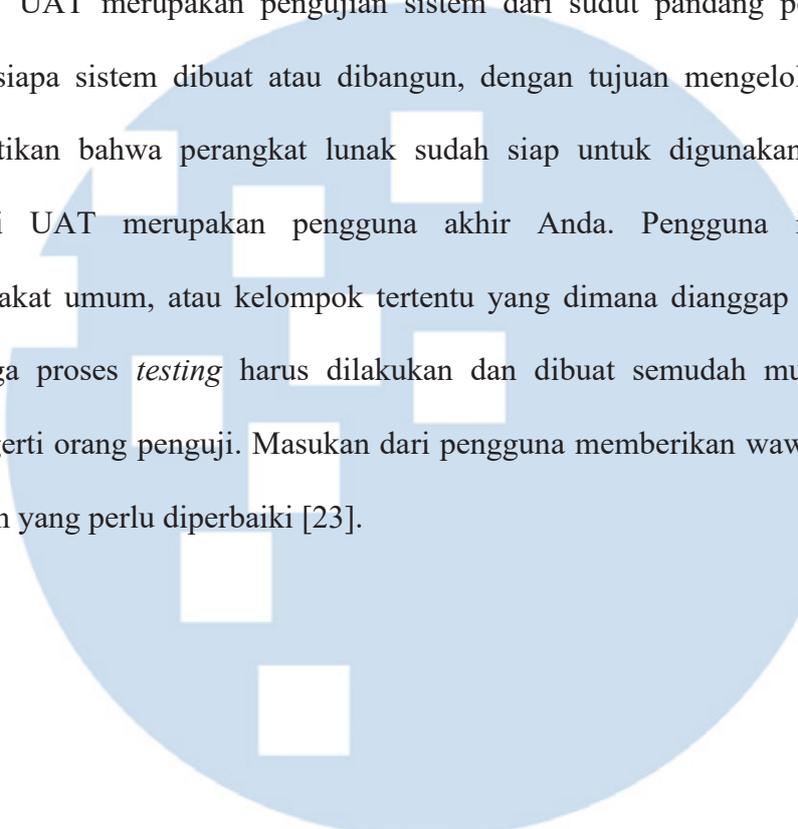
Sistem *database* adalah alat yang menyederhanakan tugas-tugas dalam mengelola data dan mengekstraksi informasi yang berguna secara tepat. *Database* menganalisis dan memandu aktivitas atau tujuan bisnis suatu organisasi. *Database* adalah tempat penyimpanan data pusat dalam sistem informasi organisasi dan penting untuk mendukung fungsi organisasi, memelihara data untuk fungsi-fungsi ini dan membantu pengguna menafsirkan data dalam pengambilan keputusan.

Manajer menggunakan pengetahuan yang berasal dari *database* untuk keunggulan kompetitif, misalnya, untuk menentukan pola pembelian pelanggan, pelacakan penjualan, dukungan manajemen hubungan pelanggan (CRM), belanja *online*, manajemen hubungan karyawan, menerapkan sistem pendukung keputusan (DSS), mengelola inventaris dan sebagainya. Untuk memenuhi kebutuhan organisasi yang berubah, struktur *database* harus fleksibel untuk memberi akses pada data baru dan mengakomodasi hubungan baru untuk mendukung keputusan baru [22].



2.18 User Acceptance Test (UAT)

UAT merupakan pengujian sistem dari sudut pandang pengguna dan untuk siapa sistem dibuat atau dibangun, dengan tujuan mengelola risiko dan memastikan bahwa perangkat lunak sudah siap untuk digunakan dan dirilis. Penguji UAT merupakan pengguna akhir Anda. Pengguna ini biasanya masyarakat umum, atau kelompok tertentu yang dimana dianggap masih awam sehingga proses *testing* harus dilakukan dan dibuat semudah mungkin untuk dimengerti orang penguji. Masukan dari pengguna memberikan wawasan tentang masalah yang perlu diperbaiki [23].

A large, light blue watermark logo of Universitas Multimedia Nusantara (UMMN) is centered on the page. It features a stylized 'U' and 'M' inside a circle, with the letters 'U', 'M', and 'N' arranged vertically.

UMMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA

2.19 Penelitian Terdahulu

Tabel 2. 1 Tabel Penelitian Terdahulu

No	Nama Journal, Vol, No, Tahun	Penulis	Hasil Penelitian	Kontribusi di Penelitian ini
1.	Sensors, 17, 7, 2020	Navin Kumar, Aritri Debnath	Penelitian bertujuan untuk menyelesaikan masalah dalam mencari jaringan yang memiliki lalu lintas, kecepatan, konektivitas, dan aksesibilitas yang baik untuk sinyal data internet <i>mobile</i> .	Penelitian ini memberikan gambaran bagaimana DSS yang menggunakan metode SAW untuk menyelesaikan suatu masalah.
2.	IJNMT, 8, 1, 2020	Harya Bima Dirgantara, Henri Septanto.	Penelitian ini membuat game anak-anak berbasis <i>website</i> dengan menggunakan proses SDLC <i>prototyping</i> .	Penelitian ini memberikan informasi bagaimana proses SDLC <i>prototyping</i> membentuk sistem berbasis <i>website</i> .
3.	Jurnal Teknik Informatika dan Sistem Informasi, Vol 5, No. 1, 2018	Petrus Sokibi, Apriyanto Noer Setiawan.	Pengguna Sistem Pendukung Keputusan dalam menentukan penilaian kinerja karyawan yang terbaik, dapat membantu dan mempermudah perusahaan dalam menilai kinerja karyawannya.	Penelitian ini memberikan gambaran kriteria dan bentuk data yang seperti apa jika ingin menggunakan DSS dengan SAW untuk menentukan penilaian kinerja pegawai.
4.	IJNMT, 7, 2 2020	Eko Budi Setiawan, Angga Setiyadi, Wildan Agung Prakoso	Dalam jurnal ini penulis membuat sistem pengambil keputusan menggunakan metode <i>Simple Additive Weighting</i> untuk menentukan lokasi yang cocok untuk membuka UMKM.	Penelitian ini memberikan informasi mengenai bagaimana pembuatan <i>prototype</i> sistem pengambilan keputusan menggunakan metode SAW