

## BAB II LANDASAN TEORI

### 2.1 Tinjauan Pustaka

Tinjauan pustaka adalah rangkuman tertulis mengenai artikel dari jurnal, buku dan dokumen lain yang menjelaskan teori serta informasi dari masa lalu maupun masa sekarang. Berikut ini adalah beberapa penelitian dari penelitian sebelumnya, pada sebagai berikut:

**Tabel 2. 1** Tinjauan Pustaka

No. Literatur	Peneliti Tahun	Judul Penelitian
Literatur 1	(Canggih Ajika Pamungkas 2019)	Aplikasi penghitung jarak koordinat berdasarkan <i>latitude</i> dan <i>longitude</i> dengan metode <i>euclidean distance</i> Dan metode <i>haversine</i>
Literatur 2	(Miftahuddin, Umaroh, and Karim 2020)	Perbandingan metode perhitungan jarak <i>euclidean</i> , <i>haversine</i> , dan <i>manhattan</i> dalam penentuan posisi karyawan.
Literatur 3	(Zulmiyah, Teknik, and Politeknik 2018)	Program perhitungan luas lahan segi empat menggunakan bahasa c++ berbasis <i>codeblocks</i>
Literatur 4	(Kusumo 2020)	Perhitungan Luas Permukaan Dan <i>Volume</i> Daratan Indonesia Menggunakan Data <i>Digital Elevation Model</i>
Literatur 5	(Nishom 2019)	Perbandingan Akurasi <i>Euclidean Distance</i> , <i>Minkowski Distance</i> , dan <i>Manhattan Distance</i> pada Algoritma <i>KMeans Clustering</i> berbasis <i>Chi-Squar</i> .
Literatur 6	(Andrew 2021)	Pemanfaatan <i>Global Positioning System</i> (GPS) Untuk Menghitung Panjang Dan Luas Lahan

### **2.1.1 Literatur 1 (Canggih Ajika Pamungkas 2019)**

Pada penelitian ini (Canggih Ajika Pamungkas 2019) mencari perbandingan antara 2 algoritma *Euclidean Distance* dan *metode Haversine*. Dan keluaran yang dihasilkan dari implementasi kedua metode, yaitu *Euclidean Distance* dan *Haversine*, menghasilkan nilai yang sama untuk jarak pendek. Hal ini disebabkan oleh sifat perhitungan pada jarak yang relatif dekat, di mana kelengkungan bumi belum memberikan perbedaan yang signifikan pada hasil perhitungan kedua metode tersebut.

### **2.1.2 Literatur 2 (Miftahuddin et al. 2020)**

Dalam penelitian yang berjudul “Perbandingan metode perhitungan jarak *euclidean*, *haversine*, dan *manhattan* dalam penentuan posisi karyawan”. Hasil pengujian pengambilan data koordinat dengan menggunakan fitur GPS pada *smartphone*, mendapatkan waktu rata-rata dikirimnya data koordinat sampai tersimpan di database sistem adalah 0,9 detik. Dari tabel perbandingan kinerja pada Tabel 5 dapat disimpulkan bahwa metode penghitungan jarak yang dapat diterapkan pada sistem identifikasi lokasi pegawai adalah metode *Euclidean* dan metode *Haversine*. Hal ini disebabkan selisih jarak rata-rata dari perhitungan yang menghasilkan hasil sebenarnya kurang dari 1. 0,5 meter. Di sisi lain, metode perhitungan *Manhattan* tidak sesuai untuk digunakan dalam studi kasus *Employee Location Identifier*, karena perbedaan jarak rata-rata dengan perhitungan sebenarnya adalah 6,67 meter, yang dapat menjadi titik awal terjadinya penipuan. Kedua, masih terdapat perbedaan krusial antara *Euclidean* dan *Haversine*, dimana selisih kesalahannya kecil.

Dengan menerapkan aplikasi di perimeter, *Haversine* mencapai akurasi pengambilan keputusan yang lebih akurat. Hasil dari setiap perhitungan jarak mungkin disebabkan oleh ide dari masing-masing metode perhitungan. Penyimpangan terbesar terjadi di *Manhattan*. Konsep penghitungan jarak *Manhattan* adalah dengan menerapkan konsep mencari perbedaan murni antar data. Ini tidak cocok untuk penghitungan jarak menggunakan variabel koordinat lintang dan bujur. Kalkulus *Euclidean* menerapkan konsep *Pythagoras*, sedangkan kalkulus *Haversine* menerapkan konsep penghitungan jarak pada permukaan bola tanpa mempertimbangkan kemiringan. Hasil penelitian menunjukkan bahwa Haversin memiliki tingkat akurasi yang lebih tinggi dibandingkan *Euclidean*. Hasil analisis tersebut menunjukkan bahwa kemiringan permukaan tetap perlu diperhatikan saat menghitung jarak di atas permukaan.

### **2.1.3 Literatur 3 (Zulmiyah et al. 2018)**

Dari penelitian yang dilakukan oleh (Zulmiyah et al. 2018) pada perhitungan luas lahan menggunakan program berbasis *C++* di *Codeblocks* dengan *dataset* tertentu menunjukkan bahwa hasil perhitungan lebih akurat dibandingkan metode manual. Selain itu, penggunaan program ini terbukti lebih efisien dan menghemat waktu, sehingga sangat praktis dalam aplikasinya.

### **2.1.4 Literatur 4 (Kusumo 2020)**

Luas permukaan memiliki perbedaan dengan luas area. Luas permukaan merupakan representasi sesungguhnya dari suatu permukaan yang dipengaruhi oleh bentuk topografinya, sedangkan luas area merupakan representasi permukaan yang diproyeksikan pada bidang datar. Permukaan daratan yang bergelombang apabila diletakkan berimpit dengan proyeksinya pada bidang datar maka akan

terbentuk sebuah ruang. Ruang diantara permukaan bergelombang dengan bidang datar disebut sebagai volume daratan. Luas permukaan dan volume daratan dapat dihitung dengan menggunakan data *Digital Elevation Model* (DEM). Luas permukaan memiliki peran penting dalam analisis permukaan seperti mengestimasi ketersediaan area untuk habitat hewan dan tumbuhan. Volume daratan dapat dimanfaatkan untuk mengetahui volume ruang penyimpanan air tanah pada akuifer yang terdapat pada lapisan-lapisan batuan di bawah permukaan. Luas permukaan dan volume daratan Indonesia dihitung dengan menggunakan data DEMNAS yang memiliki sistem koordinat WGS 84, datum vertikal EGM 2008 dan resolusi spasial 8,25 meter. Data DEMNAS diproyeksikan pada sistem proyeksi *Universal Transverse Mercator* (UTM). Luas permukaan dan volume daratan Indonesia dihitung dengan menggunakan metode *surface volume* pada perangkat lunak *Arcmap*. Berdasarkan uji metode menggunakan DEM simulasi dengan membandingkan hasil hitungan manual dengan hasil hitungan metode *surface volume*, diperoleh hasil perbedaan hitungan luas permukaan setelah dirata-rata adalah sebesar 1%, sedangkan perbedaan hitungan *volume* adalah sebesar 0,025%.

#### **2.1.5 Literatur 5 (Nishom 2019)**

Perbandingan akurasi metode pengukuran jarak (*euclidean*, *manhattan*, dan *minkowski*) untuk pelabelan klaster status disparitas kebutuhan Guu telah dilakukan dan memberikan nilai atau tingkat akurasi yang tinggi, yaitu 84.47% (untuk metode *euclidean distance*), 83.85% (untuk metode *manhattan distance*), dan 83.85% (untuk metode *minkowski*). Dengan demikian, dapat disimpulkan bahwa metode *euclidean* merupakan metode terbaik untuk diterapkan dalam

algoritma *KMeans Clustering*. Selanjutnya, klaster label klaster dapat digunakan mengidentifikasi status disparitas Guru untuk masing-masing sekolah di Kota Tegal. Berdasarkan pelabelan pada klaster tersebut, maka dapat diketahui bahwa sekolah dengan kondisi ketersediaan Guru yang masih sangat kurang (kategori label disparitas TINGGI) dan perlu mendapatkan perhatian lebih adalah SMP Atmaja Wacana, SMKN 3 Tegal, SMAS Muhammadiyah, SMAS Pancasakti Tegal, SMKS Muhammadiyah 1 Kota Tegal, dan SMP IC Bias Assalam.

#### **2.1.6 Literatur 6 (Andrew 2021)**

Pada penelitian ini (Andrew 2021) mengungkapkan bahwa penggunaan *Global Positioning System* (GPS) telah mengalami perkembangan yang signifikan, terutama dalam hal pengukuran lahan. GPS memungkinkan pengukuran panjang dan luas lahan dilakukan dengan lebih cepat, presisi, dan efisien dibandingkan metode konvensional. Dalam penerapannya, GPS menyediakan data koordinat secara langsung yang dapat digunakan untuk menentukan batas-batas dan luas area tertentu. Metode ini tidak hanya mengurangi kesalahan manusia dalam pengukuran, tetapi juga menghemat biaya dan waktu. Andrew juga menyatakan bahwa teknologi GPS memiliki potensi besar untuk diterapkan dalam berbagai bidang, seperti pertanian, perencanaan tata kota, dan pengelolaan sumber daya alam. Metode yang digunakan pada penelitian ini yaitu menggunakan metode pengembangan *System Development Life Cycle* (SDLC), dengan pendekatan model *prototype*. Pada penelitian yang dilakukan oleh Andrew dapat disimpulkan bahwa pemanfaatan GPS pada *smartphone* dapat digunakan untuk menghitung luas tanah dengan bantuan aplikasi tambahan. Penggunaan aplikasi GPS pada *smartphone* ini tidak hanya mempermudah proses perhitungan, tetapi juga

menghemat biaya selama proses berlangsung. Tingkat akurasi dari hasil perhitungan ini mencapai 83%, sehingga cukup andal dalam mengukur luas tanah. Namun, keberhasilan proses ini sangat bergantung pada konektivitas internet yang stabil untuk memastikan data yang diperoleh akurat dan tepat waktu.

## **2.2 Implementasi**

Implementasi adalah suatu kegiatan dari sebuah rencana yang dibuat secara terperinci untuk mencapai suatu tujuan. Implementasi mulai dilakukan apabila seluruh perencanaan sudah dianggap sempurna. Implementasi menurut teori Jones bahwa :

*“Those Activities directed toward putting a program into effect”*(Proses mewujudkan program hingga memperlihatkan hasilnya). Jadi

Implementasi adalah tindakan yang dilakukan setelah suatu kebijakan ditetapkan. Implementasi merupakan cara agar sebuah kebijakan dapat mencapai tujuannya.

## **2.3 Algoritma**

Pengertian umum dari suatu algoritma adalah urutan dari sejumlah langkah logis dan sistematis untuk memecahkan suatu masalah tertentu. Beberapa ahli menganggap algoritma sebagai urutan langkah-langkah yang harus diikuti dalam matematika atau perhitungan untuk memecahkan masalah lain, terutama komputer.

Algoritma menurut (Kani, 2020, 1.19) adalah suatu upaya dengan urutan operasi yang disusun secara logis dan sistematis untuk menyelesaikan suatu masalah untuk menghasilkan suatu *output* tertentu.

## 2.4 Manhattan Distance

*Manhattan distance* atau jarak *Manhattan* sering juga disebut *CityBlock Distance*. Adalah metode pengukur jarak antara dua titik dalam sistem koordinat. Pada dasarnya *Manhattan Distance* dihitung dengan menjumlah selisih nilai antara koordinat x dan y dari kedua titik tersebut. *Manhattan Distance* sering digunakan dalam berbagai bidang, seperti dalam algoritma pencarian jarak terdekat, sistem rekomendasi, perencanaan *route*, dan pengolahan citra. *Manhattan Distance* dihitung sebagai jumlah dari perbedaan mutlak antara dua *vector*. *Manhattan Distance* juga dapat diterapkan pada perhitungan jarak antara dua titik berdasarkan data koordinat geografis seperti *latitude* (garis lintang) dan *longitude* (garis bujur) (Buaton, Sundari, and Maulita 2016) Dalam hal ini, perhitungan *Manhattan Distance* berfokus pada perbedaan antara nilai *latitude* dan *longitude* dari kedua titik. Adapun rumus dari *Manhattan distance* adalah :

$$d(x,y) = \sum_{i=1}^n |x_i - y_i|. (1 \text{ derajat bumi})$$

$d(x,y)$  = Jarak

x = koordinat lokasi 1

y = koordinat lokasi 2

$\sum_{i=1}^n$  =notasi sigma, yang menunjukkan penjumlahan dari elemen-elemen  $|x_i|$  untuk  $i$  mulai dari 1 hingga  $n$ .

1 derajat Bumi= 111,322 km

Berikut adalah contoh untuk mengilustrasikan perhitungan *Manhattan Distance* antara dua titik berdasarkan *latitude* dan *longitude*:

**Tabel 2. 2** Koordinat  
(Miftahuddin et al. 2020)

Lokasi	<i>Latitude</i>	<i>Longitude</i>
Titik A	-6,8971795	107,63647
Titik B	-6,897536468	107,6363645

$$d(x, y) = \sum_{i=1}^n |\text{koordinat 1} - \text{koordinat 2}|. (1 \text{ derajat bumi})$$

Sehingga,

$$d(x, y) = \left( \frac{|-6,8971795 - (-6,897536468)| +}{107,63647 - 107,6363645} \right) \cdot 111,322 \text{ km}$$

$$d = 0,04846 \text{ km} \longrightarrow d = 48,46 \text{ m}$$

jika dikonversikan kedalam satuan meter maka  $d = 48,46$  meter

## 2.5 Perhitungan

Perhitungan menurut kamus umum bahasa Indonesia karangan poerwadarminta sebagai berikut:

“perhitungan adalah penjumlahan / penentuan total sesuatu atau pembayaran untuk sebuah jasa dan ongkos antaran”. (Eviana et al. 2022)

## 2.6 Luas Tanah

Luas tanah adalah ukuran dari total luas area yang dimiliki oleh suatu properti atau bangunan, zona yang ditempati oleh bangunan maupun yang tidak ada bangunan. Luas tanah juga dapat diukur dalam satuan meter persegi ( $m^2$ ) atau



hektar (ha). Meter persegi dapat dipakai dalam pengukuran luas bangunan, lahan atau ruang. Sedangkan hektar adalah satuan ukuran luas yang bias digunakan untuk mengukur lahan pertanian, kehutanan, atau lahan - lahan yang luas. Satu hektar setara dengan 10.000 meter persegi ( $m^2$ ). Luas tanah juga bias lebih besar dari luas bangunan, terutama jika banyak area yang tidak ada terbangun seperti halaman, taman, atau area terbuka lainnya. Sebaliknya, luas bangunan biasanya lebih kecil daripada luas tanah, karena tidak mencakup keseluruhan area yang dimiliki (Pradnyawati and Cipta 2021).

## 2.7 CodeIgniter

Menurut Budi Raharjo (2015:3), “ *CodeIgniter* adalah *framework web* untuk bahasa pemrograman PHP yang dibuat oleh Rick Ellis pada tahun 2006, penemu dan pendiri EllisLab. EllisLab adalah suatu tim kerja yang berdiri pada tahun 2002 dan bergerak dibidang pembuatan *software* dan *tool* untuk para mengembang *web*”.

*CodeIgniter* memiliki banyak fitur yang membantu pengembang (*developer*) PHP untuk dapat membuat aplikasi *web* secara mudah dan cepat. Dibandingkan dengan *framework web* PHP lainnya, harus diakui bahwa *CodeIgniter* memiliki desain yang lebih sederhana dan bersifat fleksibel. *CodeIgniter* mengizinkan para pengembang untuk menggunakan *framework* secara parsial atau secara keseluruhan.

## 2.8 Visual Studio Code

Visual Studio Code adalah kode editor yang dikembangkan oleh Microsoft untuk Windows, Linux dan MacOS. Ini termasuk dukungan debugging, GIT kontrol yang disematkan, penyorotan sintaks, penyelesaian kode cerdas, cuplikan,

dan kode refactoring. Menurut Lardinois, 2015) Visual Studio Code didasarkan pada Elektron, kerangka kerja yang digunakan untuk menyebarkan aplikasi Node.js untuk desktop yang berjalan pada Blinklayout. Meskipun menggunakan kerangka Elektron, Visual Studio Code tidak menggunakan Atom dan menggunakan komponen editor yang sama (diberi kode nama "*Monaco*") yang digunakan dalam Visual Studio Team Services yang sebelumnya disebut Visual Studio Online.

## 2.9 Pengertian Website

Pengertian website atau disingkat web, dapat diartikan sekumpulan halaman yang terdiri atas beberapa laman yang berisi informasi dalam bentuk data digital, baik berupa teks, gambar, video, audio, dan animasi lainnya yang disediakan melalui alur koneksi internet”.

Menurut Lukmanul Hakim (2004), *Website* merupakan fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen dalam *website* disebut dengan *webpage* dan *link* dalam *website* dapat digunakan oleh pengguna untuk beralih dari satu halaman ke halaman (*hypertext*) lain baik antar halam yang disimpan di server yang sama maupun dalam server yang ada di seluruh dunia. Halaman (*page*) dapat di akses atau di baca melalui *browser* seperti Google Chrome, Mozilla Firefox dan lain sebagainya.

## 2.10 HTML (HyperText Markup Language)

HTML ialah singkatan dari *Hypertext Markup Language*. HTML ialah bahasa markup yang tujuannya untuk menandai isi dari suatu dokumen agar tidak dipahami sebagai teks biasa, dapat dihubungkan satu sama lain, dan di pahami oleh perangkat lunak pembaca yang bias dipahami, yang di sebut *web browser*.

HTML dimodelkan menggunakan SGML (*Standar Generalized Markup Language*), merupakan standar deklaratif untuk mendeskripsikan struktur juga atribut dokumen. HTML ialah kumpulan dari kode kode yang biasa digunakan untuk menentukan teks dalam dokumen sebagai elemen dokument dan biasa ditautkan satu sama lain.

Adapun perbedaan dokumen *web* dengan dokumen *html*. Dokumen *web* merupakan dokumen yang berisi informasi yang di tampilkan di setiap situs *web*, baik di internet maupun intranet, pada umumnya dengan menggunakan *browser web*. Dokumen *web* dikenal juga dengan dokumen HTML ialah *file* teks biasa, yang elemen teks yang ada di dalamnya mempunyai kode *element*, yang disebut dengan *tag*. Dengan adanya kode *element* yang ada pada teks, maka *browser* dapat menampilkan dokumen sesuai dengan definisi dari kodenya (Sidik 2019).

Beberapa tugas utama dari HTML saat membangun sebuah *website* antara lain :

1. Menentukan tata letak situs *web*.
2. Pemformatan teks dasar, seperti pengaturan *paragraph* juga pemformatan *font*.
3. Buat daftar
4. Buat *table*
5. Sisipkan gambar, video, dan audio.
6. Membuat *link*.

Membuat formulir.

## 2.11 CSS (Cascade Style Sheet)

CSS (*Cascade Style Sheet*) ialah suatu bahasa yang digunakan dalam menjelaskan *style* dari suatu dokumen HTML harus ditampilkan dengan

menggunakan definisi yang di buat secara terpisah dari kode HTML nya. Dengan adanya CSS maka pengembangan dari *web* tidak lagi menggunakan atribut dari suatu *element*, karena sudah di gantikan dengan CSS. Penggunaan atribut, walaupun dengan atribut *style* diharapkan bias dihindari.

Dengan CSS kita dapat memisahkan komponen atau elemen yang digunakan untuk menampilkan informasi dengan informasinya sendiri. Sebelum ada CSS isi dari dokumen HTML tercampur antara elemen dokumen dengan komponen untuk bagaimana menampilkannya, terlalu banyak atribut yang harus dituliskan dengan *tag* elemen.

*Style* atau gaya di dalam CSS berhubungan dengan atribut elemen dokumen HTML. Memberikan *style* atau gaya kepada elemen elemen dokumen agar dokumen dapat ditampilkan secara menarik di halaman *browser web*. Cara memberikan *style* ke dalam elemen dokumen :

1. *Inline*
2. *Internal*
3. *Eksternal*

Ketiga cara pemberian *Style* di atas biasa dilakukan secara kombinasi, tidak harus memilih satu cara di dalam satu dokumen *web* (Sidik 2019).

## **2.12 PHP**

*Hypertext Preprocessor* (PHP) ialah *script* pemograman yang terletak dan dieksekusi di server. Salah satunya ialah untuk menerima, mengolah, juga menampilkan data dari dan sebuah situs (Turnip 2021).

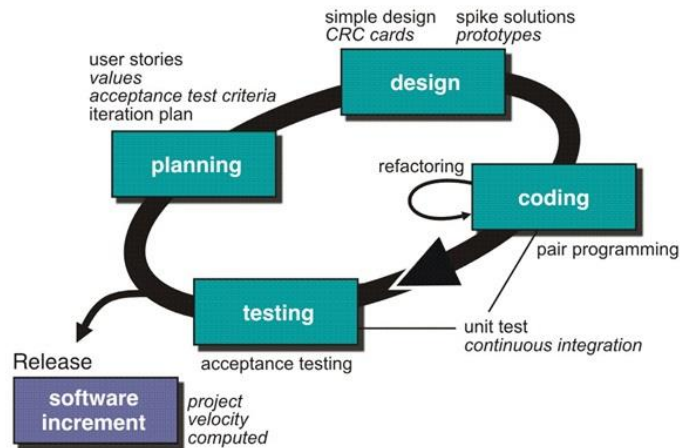
### 2.13 MySQL

Basis data (*Database*) ialah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga bisa diperiksa menggunakan suatu program komputer guna memperoleh suatu informasi dari basis data tersebut. Untuk mengelola database diperlukan dalam suatu perangkat lunak yang disebut DBMS (*Database Management System*). DBMS merupakan suatu sistem perangkat lunak yang memungkinkan pengguna untuk membuat, memelihara, mengontrol, juga mengakses database secara praktis juga efisien. Sedangkan RDBMS (*Relationship Database Management System*) merupakan salah satu dari jenis DBMS yang mendukung adanya *relationship* atau hubungan antar table (Wartarius, Arifin, Zainal, and Junaidi 2020).

### 2.14 Extreme Programming

Teknik pemrograman *extreme* adalah metodologi yang berkembang dan muncul sebagai *respons* terhadap masalah yang dihadapi dalam pengembangan perangkat lunak dengan menggunakan model pengembangan tradisional. Model pengembangan tradisional mengacu pada perencanaan, analisis, dan perencanaan *system*, namun karena skalanya yang besar, setiap fase membutuhkan banyak waktu, sehingga XP menawarkan *alternative*.

XP menawarkan fase – fase lebih pendek yang berulang dibagian berbeda tergantung *focus* yang ingin dicapai (Akbar 2017) .



**Gambar 2. 1 Extreme Programming (XP)**

### 2.15 Unified Modeling Language (UML)

*Unified Modelling Language* (UML) merupakan bahasa berbasis *grafis/gambar* untuk membayangkan, menentukan, merakit, dan melaporkan kerangka kemajuan pada pemrograman berbasis OO (*Object Oriented*). UML juga menyediakan spesifikasi untuk menyusun sistem cetak biru, yang menggabungkan konsep skala bisnis, kompilasi kelas dalam dialek pemrograman eksplisit, skema *database*, dan komponen yang diperlukan perangkat lunak sistem (Zufria 2013).

UML pertama kali diperkenalkan pada tahun 1990, ketika Gady Booch dan Ivar Jacobson dan James Rumbaugh mulai merangkul ide dan kemampuan tambahan dari teknologi masing-masing lalu mencoba membuat strategi terikat, yang kemudian diberi nama UML. Pada bulan Oktober 1994, ketika Rumbaugh bergabung dengan Booch di *Rational Software Corporation*, upaya untuk memajukan UML. Proyek pertama mereka ialah menggabungkan metode Booch dengan OMT (*Object Modeling Technology*). UML versi 0.8 dirilis pada Oktober 1995. Saat mengirimkan formulir UML 0.8, Jacobson juga menambahkan

*Reasonable* untuk memperluas UML agar menyertakan OOSE. Hasilnya ialah adaptasi dari UML versi 0.9 yang dikirimkan pada bulan Juni 1996. Sepanjang tahun 1996, Rumbaugh, Booch, dan Jacobson juga menerima banyak kontribusi dari komunitas rekayasa perangkat lunak. Saat ini, banyak asosiasi perangkat lunak melihat UML sebagai sebuah sistem untuk bisnis mereka. Tujuan dari *Unified Modelling Language* (UML) diantaranya adalah (Zufria 2013) :

1. Menyediakan bahasa tampilan *visual* siap pakai yang berterima kasih karena telah membuat dan memperdagangkan model dengan cara yang sederhana dan secara umum dapat dibenarkan.
2. Menyediakan bahasa tampilan yang bebas dari berbagai dialek pemrograman dan siklus desain.
3. Sertakan prosedur yang ditentukan dalam demonstrasi.

Rosa dan Salahuddin menyatakan bahwa UML (*Unified Modeling Language*) merupakan salah satu prinsip bahasa yang digunakan di dunia modern untuk mendeskripsikan kebutuhan, menganalisis, mendesain, juga mendeskripsikan arsitektur dalam pemrograman berorientasi objek (PBO). Berikut ini adalah jenis jenis diagram UML (*Unified Modelling Language*)

### **2.15.1 Bangun Dasar UML**

Dalam memahami UML, kita memerlukan model konseptual dari "bahasa" ini. Metode UML memiliki 3 struktur dasar untuk menggambarkan sistem/perangkat lunak yang sedang dikembangkan, yaitu:

1. Benda (*Things*)
2. Relasi (*Relationship*)
3. Diagram

Setiap konstruksi dasar bisa diterapkan ke dalam seluruh fase pengembangan sistem. Ketiga jenis bangunan ini bisa saling melengkapi. secara umum, UML digunakan di dalam pengembangan sistem/perangkat lunak berorientasi objek, karena metode UML ini memiliki kelebihan sebagai berikut (Zufria 2013):

1. *Uniformity*

Dengan metode UML, pengembang hanya perlu menggunakan satu metode, mulai dari tahap analisis hingga desain. Hal ini tidak dapat dilakukan dalam pendekatan pengembangan terstruktur. Dengan perkembangan aplikasi GUI (*graphical user interface*) saat ini, UML juga memungkinkan kita untuk merancang komponen antarmuka pengguna dan desain perangkat lunak dan desain *database* secara terintegrasi.

2. *Understandibility*

Dengan menggunakan metode ini, kode yang dihasilkan dapat diatur ke dalam kelas-kelas yang terkait dengan masalah yang sebenarnya, agar lebih mudah dipahami oleh siapapun.

3. *Stability*

Kode program yang dihasilkan lebih stabil dari waktu ke waktu karena sangat dekat dengan masalah sebenarnya yang ada di situs.

4. *Reusability*




Menggunakan pendekatan berorientasi objek, kode dapat digunakan kembali, yang sangat mempercepat waktu pengembangan perangkat lunak.

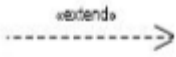

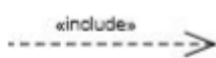


### 2.15.2 Use Case Diagram

*Use case* akan menggambarkan tampilan luar dari sistem yang akan di modelkan. Model *use case* dapat dideskripsikan dalam *use case* diagram, tetapi perlu diingat bahwa diagram berbeda dari model karena model lebih luas dari diagram. *Use case* harus mampu menggambarkan urutan aktor yang menghasilkan nilai yang terukur (Suendri 2019). Simbol yang digunakan dalam *Use case* diagram yaitu:

**Tabel 2. 3 Use Case Diagram**

No	Simbol	Nama	Keterangan
1		Aktor	Proses <i>user</i> , atau sebuah sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat pada luar sistem informasi yang akan dibuat itu sendiri, sehingga walaupun simbol dari aktor adalah gambar orang, biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i> .
2		<u>Use case</u>	Fungsionalitas yang sudah disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau <i>actor</i> yang biasanya dinyatakan dengan menggunakan kata kerja pada awal frase nama <i>use case</i> .
3		<i>Association</i>	Komunikasi yang terjadi antara <i>actor</i> dan <i>use case</i> yang berkontribusi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i> .


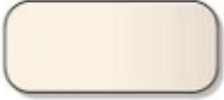

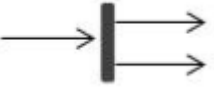


4		<i>Extend</i>	Hubungan <i>use case</i> tambahan ke dalam <i>use case</i> dimana <i>use case</i> yang telah ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan mempunyai nama depan yang sama dengan <i>use case</i> yang di tambahkan.
5		<i>Generalizati on</i>	Relasi generalisasi dan spesialisasi (umum-khusus) antara satu atau lebih <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya.
6		<i>Include</i>	Hubungan <i>use case</i> tambahan dalam sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan sehingga memerlukan <i>use case</i> ini untuk terjalankannya fungsional atau sebagai syarat dijalankan <i>use case</i> ini.

Sumber : (Rosa & Salahuddin, 2013)

### 2.15.3 Activity Diagram

*Activity Diagram* adalah diagram yang menunjukkan aktivitas sistem dalam bentuk serangkaian tindakan, bagaimana setiap tindakan dimulai, dan keputusan yang mungkin terjadi sebelum tindakan itu berakhir. *Activity Diagram* jugat menggambarkan proses dari beberapa tindakan pada saat yang bersamaan. *Activity Diagram* adalah aktivitas, objek, status, transisi status, dan peristiwa. Diagram alur kerja aktivitas menggambarkan perilaku sistem dari aktivitas (Suendri 2019) Berikut simbol ictivity diagram, yaitu :

Tabel 2. 4 Simbol *Activity Diagram*

No	Simbol	Nama	Keterangan
1		<i>Initial</i>	Pada Status awal aktivitas sistem, dimana sebuah diagram aktivitas memiliki sebuah status awal.
2		<i>Activity</i>	Aktivitas sistem yang dilakukan, biasanya diawali dengan kata kerja.
3		<i>Decision</i>	Asosiasi percabangan yang dimana lebih dari satu aktivitas digabungkan menjadi satu.
4		Join Node	Asosiasi penggabungan yang dimana lebih dari satu aktivitas lebih dari satu.
5		Final	Pada status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas hanya memiliki sebuah status satu.
6		<i>Simline</i>	Tidak menggabungkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

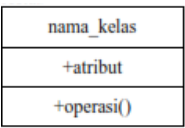


Sumber : (Rosa & Salahuddin, 2018)

#### 2.15.4 Class Diagram

*Class* diagram menjelaskan tentang jenis objek di dalam sistem dan berbagai hubungan statis yang ada di antara mereka. Diagram kelas juga menunjukkan atribut dan operasi kelas dan batasan yang terkandung dalam hubungan objek-objek ini . *Class* diagram ini mencakup tiga bidang utama, yaitu:

1. Nama, kelas harus mempunyai sebuah nama.
2. Atribut, adalah perangkat yang terpasang ke kelas. Nilai suatu kelas hanya dapat diproses dalam rentang atributnya.
3. Operasi, adalah sebuah proses yang dapat dieksekusi oleh kelas, dapat dieksekusi di kelas itu sendiri, juga dapat dieksekusi di kelas lain.

**Tabel 2. 5 Class Diagram**

No	Simbol	Nama	Keterangan
1		Kelas	Kelas yang terdapat pada struktur system
2		Antar muka	Sama dengan konsep antarmuka didalam pemrograman berorientasi objek
3		Antar muka	Hubungan antarkelas yaitu dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4		Asosiasi berarah	Hubungan antarkelas dengan makna kelas yang satu, dapat digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
5		Generalisasi	Hubungan antar kelas memiliki makna generalisasi spesialisasi (umum khusus)
6		<i>dependency</i>	Hubungan antar kelas memiliki makna kebergantungan antarkelas

Sumber : (Rosa & Salahuddin, 2018)

### 2.16 Pengujian ISO 25010

Menurut (Lamada et al. 2020), Di antara berbagai sudut pandang pengujian, ISO 9126 dan ISO 25010 adalah standar dari pengujian perangkat lunak internasional. Standar ISO 25010 dikembangkan untuk menggantikan standar ISO 9126 berdasarkan perkembangan ICT (*information and Communication Teknologi*). Standar ISO 25010 juga memiliki delapan karakteristik, yaitu kesesuaian fungsional, keandalan, efisiensi, ketersediaan, keamanan, kompatibilitas, pemeliharaan, dan portabilitas. Berikut adalah penjelasan dari delapan karakteristik ISO 25010 yaitu :

1. *Functionality Suitability*, adalah karakteristik yang menggunakan alat penelitian berupa *test case* dengan skala Guttman. Skala Guttman digunakan sebagai pemberi jawaban pasti atas masalah yang ingin di angkat.
2. *Performance Efficiency*, digunakan untuk menguji tingkat kinerja aplikasi yang sedang dikembangkan.
3. *Compatibility*, adalah kemampuan dari suatu komponen atau sistem dalam bertukar informasi.
4. *Usability*, ini dilakukan dengan menganalisis umpan balik pengguna menggunakan lima pilihan.
5. *Realibility*, digunakan untuk menguji keandalan atau sebuah keterpercayaan sistem.
6. *Security*, adalah pengecekan sejauh mana sistem atau produk menyediakan layanan melindungi dari akses, penggunaan, modifikasi, gangguan, dan pengungkapan yang berbahaya.

7. *Maintainability*, menggunakan alat ukur yang telah diuji oleh peneliti langsung di lapangan kegiatan, sesuai dengan alat uji yang disebutkan Land, pengujian terdiri dari 3 aspek, yaitu alat ukur, perhitungan konsistensi dan kesadaran.

*Portability*, adalah sejauh mana sebuah sistem atau produk bisa dipindahkan dari satu ruang ke ruang lainnya.