

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Pengertian Sistem**

Menurut (Eko Maidiansyah, 2021), sistem adalah kumpulan elemen yang saling berhubungan dan berinteraksi dalam satu kesatuan untuk menjalankan suatu proses pencapaian suatu tujuan utama.

Sedangkan menurut (Jimi Asmara, 2019), sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersamasama untuk melakukan suatu kegiatan tertentu untuk mencapai tujuan tertentu.

#### **2.2 Pengertian Pendukung Keputusan**

Menurut (Zulfahmi, 2019), dukungan komputerisasi dalam hal pengambilan keputusan telah masuk ke segala bidang. Salah satunya adalah Sistem Pendukung Keputusan pemilihan siswa berprestasi yang berbasis komputer.

Sedangkan menurut (Irvan Sulistiya Putra, 2019), sistem pendukung keputusan diarancang untuk mendukung seluruh tahap pengambilan keputusan mulai dari mengidentifikasi masalah, memilih data yang relevan, dan menentukan pendekatan yang digunakan dalam proses pengambilan keputusan sampai mengevaluasi pemilihan alternatif-alternatif yang ada.

### **2.3 Pengertian Pelanggan**

Menurut (M.Munir, 2020), pelanggan adalah pihak yang membeli atau menggunakan produk/jasa yang ditawarkan.

Sedangkan menurut (Retno Karunia Putri, 2022), pelanggan adalah seseorang yang terbiasa untuk membeli barang pada sebuah toko tertentu. Pelanggan juga sering disebut dengan klien, nasabah atau juga pasien tergantung pada sifat industri atau budaya.

### **2.4 Pengertian Metode Vikor**

Menurut (Putu Citra Darmika Dewi, 2021), metode vikor merupakan suatu metode untuk menangani permasalahan multi kriteria yang kompleks dengan fokus utamanya pada ranking dan seleksi dari sebuah alternatif. Selain itu, metode ini juga memiliki kelebihan dalam kompromi alternatif

Sedangkan menurut (Fajar Muharam, 2022), vikor yaitu salah satu metode yang digunakan dalam sistem pendukung keputusan. Fokus metode Vikor ini adalah dengan dengan menjadikan patokan kriteria dan bobot setiap data alternatif, dan utilitas yang sesuai dan ukuran *regret* untuk setiap pilihan yang telah ditentukan.

### **2.5 Pengertian *Android***

Menurut (Faradika, 2020),*Android* adalah aplikasi sistem operasi untuk telepon seluler yang berbasis *Linux*. *Android* menyediakan *platform* terbuka bagi para

pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak.

Sedangkan Menurut (Rizal Fadilla, 2023), *Android* merupakan salah satu sistem operasi seluler baru yang semakin canggih dan didesain untuk mengakomodasi pengembangan aplikasi pada perangkat seluler, perkembangan sistem operasi *Android* dimulai pada awal tahun 2003 dan dikembangkan oleh *Android Inc.*

## **2.6 Pengertian *Android Studio***

Menurut (Sri Mulyati, 2019), *Android Studio* adalah lingkungan pengembangan terintegrasi resmi untuk sistem operasi *android*, yang dibangun di perangkat lunak *Jet Brains'IntelliJ IDEA* dan dirancang khusus untuk pengembangan *Android*, Ada beberapa *tools* yang harus diinstal di aplikasi *Android Studio* diantaranya *Android SDK* dan *AVD*.

### **2.7.1 *Android Software Development Kit (SDK)***

Menurut (Dicky Taruna, 2021), *Android* *SDK* mencakup perangkat tools pengembangan yang komprehensif. *Android* *SDK* terdiri dari *debugger*, *libraries*, *handset emulator*, dokumentasi, dengan menggunakan *plugin* *Android Development Tools* (ADT). Dengan ini, pengembang dapat menggunakan IDE untuk mengedit dokumen Java dan XML serta menggunakan peralatan *command line* untuk menciptakan, membangun, melakukan debug aplikasi *Android*, dan pengendalian perangkat *Android*.

### **2.7.2 *Android Virtual Device (AVD)***

Menurut (Maiyana, 2018), *Android Virtual Device* merupakan *emulator* untuk menjalankan program aplikasi *Android* yang kita buat. AVD ini selanjutnya digunakan sebagai tempat untuk test dan menjalankan aplikasi *Android* tanpa harus menggunakan perangkat *Android* yang sebenarnya. Sebelum menggunakan AVD harus menentukan karakteristiknya, misalkan dalam menentukan versi *Android*, jenis dan ukuran layar dan besarnya memori. AVD bisa dibuat sebanyak yang kita inginkan.

## **2.7 JAVA**

Menurut (Muhammad Aland, 2019), Java adalah bahasa pemrograman yang berkembang yang berorientasi objek. Teknologi *object-oriented* memandang *software* sebagai sebuah interaksi antarbagian dalam sebuah sistem dan menggambarkan bagian tersebut ke dalam satu objek yang memiliki sifat/*property*/data dan kemampuan untuk melakukan suatu tugas tertentu.

## **2.8 XML**

Menurut Safaat (Mamay Syani, 2018), *Extensible Markup Language (XML)* didesain untuk mampu menyimpan data secara ringkas dan mudah diatur dan diolah.

## 2.9 XAMPP

Menurut (Nurfitra Efriansyah Lim, 2023), XAMPP merupakan *software* yang digunakan untuk menjalankan sebuah *website* dengan basis PHP dengan menggunakan MySQL sebagai pengolah data utama di *local computer*.

## 2.10 *My Structure Query Language* (MySQL)

Menurut (Najmi Nilfaidah, 2021), MySQL adalah salah satu jenis *database server* yang sangat terkenal. Kepopulerannya disebabkan karena bersifat *free* (anda tidak perlu membayar untuk menggunakannya) pada berbagai *platform* (kecuali pada *Windows*, yang bersifat *software* atau anda perlu membayar setelah melakukan evaluasi dan memutuskan digunakan untuk keperluan produksi).

## 2.11 DATABASE

### 2.12.1 Pengertian *Database*

Menurut (Alvin Dwi Hardiansyah dan Catur Nugrahaeni Puspita Dewi, 2020), basis data adalah kumpulan data yang saling berhubungan secara logis dan didesain untuk mendapatkan data yang dibutuhkan oleh suatu organisasi.

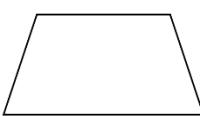
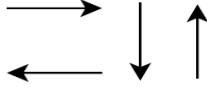
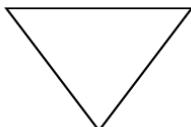
### 2.12.2 *Database Management System* (DBMS)

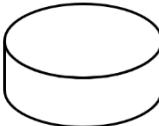
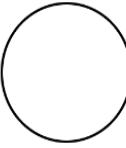
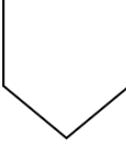
Menurut (Yanuardi, 2018), DBMS (*Database Management System*) atau dalam bahasa Indonesia sering disebut sebagai Sistem Manajemen Basis Data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data.

## 2.12 Aliran Sistem Informasi (ASI)

Menurut (Maydianto, 2021), Aliran sistem informasi merupakan alat bantu sistem yang digunakan untuk merancang sebuah sistem dari awal sampai akhir perancangan.

**Tabel 2. 1** Aliran Sistem Infromasi

| No | Gambar  | Nama            | Keterangan   |
|----|---|-----------------|--|
| 1  |    | Proses Manual   | Merupakan proses manual pada diagram alir.   |
| 2  |  | Proses Komputer | Merupakan kegiatan dari operasi program komputer.  |
| 3  |  | Simbol Dokumen  | Merupakan dokumen untuk mengambarkan input atau ouput baik untuk proses manual, mekanik atau komputer. |
| 4  |  | Garis Alir      | Merupakan arus dari proses.  |
| 5  |  | Arsip           | Merupakan arsip data yang dihasilkan.  |

|   |   |             |   |
|---|---|-------------|---|
| 6 |  | Penyimpanan | Merupakan media penyimpanan dari proses <i>entry</i> data dan proses komputerisasi. |
| 7 |  | Penghubung  | Penyambungan proses pada halaman yang sama.   |
| 8 |  | Penghubung  | Penyambungan proses pada halaman yang berbeda.                                      |

### 2.13 *Unified Modelling Language (UML)*

Menurut (Rifky Faridzi Daffa, 2022), *Extensible Markup Language (XML)* adalah bahasa markup yang diciptakan oleh konsorsium *World Wide Web (W3C)*. Bahasa ini berfungsi untuk menyederhanakan proses penyimpanan dan pengiriman data antar *server*.

#### 2.15.1 *Use Case Diagram*

Menurut (Eko Mardiyansyah, 2023), *Use case diagram* mendeskripsikan kelakuan sistem dari sudut pandang pengguna, untuk membantu memahami kebutuhan. Aktor adalah orang, proses, atau sistem lain yang mengaktifkan fungsi dari sistem target.

**Tabel 2. 2 *Use Case Diagram***

| No | Gambar | Nama | Keterangan |
|----|--------|------|------------|
|----|--------|------|------------|

|   |  |                       |   |
|---|--|-----------------------|---|
| 1 |  | <i>Actor</i>          | Merupakan proses atau sistem lain yang berinteraksi dengan sistem informasi akan dibuat diluar sistem informasi yang akan dibuat sendiri. |
| 2 |  | <i>Use Case</i>       | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .              |
| 3 |  | <i>Association</i>    | Apa yang menghubungkan antara objek satu dengan objek lainnya.  |
| 4 |  | <i>Extend</i>         | Menspesifikasi bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.              |
| 5 |  | <i>Generalization</i> | Merupakan suatu garis yang berfungsi untuk mewariskan struktur data dan objek induk kepada anak yang dituju.                              |
| 6 |  | <i>Include</i>        | Menspesifikasi bahwa <i>use case</i> sumber secara jelas.   |

### 2.15.2 Activity Diagram

Menurut (Eko Maidiansyah, S. F. 2021), *Activity diagram* ialah suatu cara untuk menggambarkan aliran kerja sebuah sistem yang ada pada perangkat lunak

**Tabel 2. 3 Activity Diagram**

| No | Gambar | Nama                | Keterangan                          |
|----|--------|---------------------|-------------------------------------|
| 1  |        | <i>Initial Node</i> | Menggambarkan titik awal aktifitas. |

|   |  |                            |  |
|---|--|----------------------------|--|
| 2 |  | <i>Actifity</i>            | Merupakan gambar aktivitas yang ada pada sistem.     |
| 3 |  | <i>Decision</i>            | Merupakan pilihan untuk pengambilan suatu keputusan. |
| 4 |  | <i>Join</i>                | Penggabungan dimana lebih dari aktifitas.            |
| 5 |  | <i>Actifity Final Node</i> | Menggambarkan titik akhir aktifitas.                 |

### 2.15.3 Class Diagram

Menurut (Sahrurun & Sularno, 2023), *Class Diagram* sering digunakan untuk menampilkan kelas, tabel, *field*, dan operasi dari program ke *database* yang digunakan. Penjelasan yang lebih rinci dari tugas seorang aktor.

**Tabel 2. 4 Class Diagram**

| No | Gambar | Nama                    | Keterangan  |
|----|--------|-------------------------|---|
| 1  |        | <i>Generalization</i>   | Hubungan dimana objek anak ( <i>descendene</i> ) berbagi perilaku dan struktur data dari objek yang ada diatas nya objek induk ( <i>ancestor</i> ). |
| 2  |        | <i>Nary Association</i> | Upaya untuk menghindari asosiasi dengan lebih dari dua objek.   |

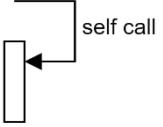
|            |  |                      |  |            |              |   |
|------------|--|----------------------|--|------------|--------------|---|
| 3          | <table border="1"> <tr><td>Nama Kelas</td></tr> <tr><td>+atribut</td></tr> <tr><td>-operasi()</td></tr> </table> | Nama Kelas           | +atribut   | -operasi() | <i>Class</i> | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama. |
| Nama Kelas |  |                      |  |            |              |   |
| +atribut   |  |                      |  |            |              |   |
| -operasi() |  |                      |  |            |              |   |
| 4          |  | <i>Collaboration</i> | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .                               |            |              |   |
| 5          |  | <i>Realization</i>   | Operasi yang benar-benar dilakukan oleh suatu objek.   |            |              |   |
| 6          |  | <i>Dependency</i>    | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri. |            |              |   |
| 7          |  | <i>Association</i>   | Apa yang menghubungkan objek satu dan lainnya.   |            |              |   |

#### 2.15.4 *Sequence Diagram*

Menurut (Khozin Yuliana, 2018), *Sequence Diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu, untuk menggambarkan *diagram sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

**Tabel 2. 5 Sequence Diagram**

| No. | Gambar | Nama        | Keterangan  |
|-----|--------|-------------|---|
| 1   |        | Actor       | Mengambarkan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem.   |
| 2   |        | Lifeline    | Menyatakan kehidupan suatu objek.   |
| 3   |        | Objek       | Antar muka yang sering berinteraksi   |
| 4   |        | Waktu Aktif | Menyatakan objek dalam keadaan aktif dan berintegrasi, semua yang berhubungan dengan waktu aktif ini adalah sebuah tahapan yang akan dilakukan di dalamnya. |
| 5   |        | Message     | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.   |
| 6   |        | Message     | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi.  |

|   |   |                     |  |
|---|---|---------------------|--|
| 7 |  | <i>Self Message</i> | Spesifikasi dari komunikasi untuk menunjsukan kegiatan memuat proses informasi pada aktifitas sendiri. |
|---|---|---------------------|--|