

BAB II

LANDASAN TEOERI

2.1 Sistem

Dalam buku Erni Widarti, Joosten, Putu Yudia Pratiwi (2024:2) dengan judul Buku Ajar Pengantar Sistem Ekonomi Indonesia. Konsep dasar sistem, istilah “sistem” merujuk pada suatu entitas yang terdiri dari elemen atau komponen yang saling berhubungan dan berinteraksi secara kontinu atau berkesinambungan dengan lingkungan sekitarnya. Hal ini bertujuan untuk mencapai suatu tujuan tertentu yang terikat dan terpadu yang berlangsung terus-menerus. Setiap elemen atau komponen yang terdapat dalam sebuah sistem memiliki fungsi dan cara kerja masing-masing tetapi tetap beroperasi secara terpadu dalam satu kesatuan fungsi atau kerja. Tugas dan fungsi antara setiap elemen atau komponen tidak akan saling bertentangan atau berlawanan, karena semuanya saling bergantung dan saling membutuhkan satu sama lain untuk mencapai tujuan tertentu.

Sistem memiliki sekumpulan elemen yang saling terkait atau terpadu untuk mencapai suatu tujuan tertentu. Elemen-elemen sistem merujuk pada komponen atau bagian-bagian yang membentuk suatu sistem. Elemen sistem bekerja bersama-sama untuk mencapai tujuan sistem secara keseluruhan. Secara umum elemen sistem terdiri dari masukan, proses, keluaran, pengendalian, tujuan dan umpan balik.

1. *Input* (Masukan) Elemen ini mencakup semua sumber atau data yang dimasukkan ke dalam sistem untuk diproses. Input menjadi dasar bagi sistem untuk menghasilkan output.
2. *Proses* (Pemrosesan) Elemen sistem yang melakukan operasi atau transformasi pada input untuk menghasilkan *output*. Proses ini melibatkan aktivitas pemrosesan, analisis, dan manipulasi data.
3. *Output* (Keluaran) Elemen *output* merupakan luaran atau informasi yang dihasilkan oleh sistem setelah pemrosesan *input*. Output ini digunakan untuk pengambilan keputusan atau sebagai masukan untuk sistem lain.
4. Pengendalian (*Control*) Elemen sistem yang mengatur dan mengontrol operasi keseluruhan. Pengendali memastikan bahwa sistem berfungsi sesuai dengan tujuan yang ditetapkan.
5. Tujuan (*Goal*) Sasaran atau hasil yang ingin dicapai oleh sistem. Elemen sistem ini memberikan arah dan tujuan bagi semua elemen dalam suatu sistem.
6. Umpan Balik (*Feedback*) Elemen *feedback* merupakan suatu informasi yang diberikan kepada sistem setelah *output* dihasilkan. *Feedback* memungkinkan sistem untuk memantau kinerjanya dan membuat perubahan jika diperlukan.

Dengan adanya elemen-elemen sistem maka suatu sistem dapat dirancang dan dioperasikan dengan lebih baik untuk mencapai tujuan yang ditetapkan.

2.1.1 Karakteristik Sistem

Dalam buku Erni Widarti, Joosten, Putu Yudia Pratiwi (2024:6) dengan judul Buku Ajar Pengantar Sistem Ekonomi Indonesia. Karakteristik sistem merupakan atribut atau sifat-sifat yang melekat pada suatu sistem yang dapat membantu mendefinisikan dan memahami suatu sistem beroperasi. Berikut beberapa karakteristik umum sistem:

1. Komponen (*Component*)

Komponen adalah elemen-elemen atau bagian -bagian yang membentuk sistem dan beberapa diantaranya melakukan interaksi dengan membentuk satu kesatuan yang saling bekerja sama yang terdiri dari berbagai cabang sistem untuk mencapai tujuan tertentu. Contohnya dalam sistem informasi, komponen dapat mencakup perangkat keras, perangkat lunak, database, dan pengguna.

2. Lingkungan Sistem (*Environment*)

Lingkungan sistem mencakup semua elemen dari luar batas sistem yang berinteraksi dengan sistem dan dapat mempengaruhi atau dipengaruhi oleh sistem. Interaksi dengan lingkungan adalah kunci dalam pemahaman sistem. Faktor eksternal yang dapat mempengaruhi operasi sistem, sistem perubahan regulasi, dan teknologi baru.

3. Batasan Sistem (*Boundary*)

Sistem memiliki batasan yang memisahkan entitas dari lingkungannya. Batasan sistem menentukan apa yang termasuk dalam sistem dan apa yang

berada diluar sistem. Dengan adanya batasan sistem dapat membantu mengidentifikasi ruang lingkup operasional sistem.

4. Penghubung Sistem (*Interface*)

Penghubung sistem mencakup hubungan dan interaksi antara satu subsistem dengan subsistem lainnya. Melalui penghubung sistem membantu sumber-sumber daya mengalir dari subsistem ke subsistem lainnya. Keluaran (*output*) dari subsistem akan menjadi masukan (*input*) untuk subsistem lain melalui penghubung sistem. Dengan adanya penghubung sistem dapat memfasilitasi aliran informasi, energi, atau bahan di dalam sistem.

5. Masukan Sistem (*Input*)

Masukan sistem adalah segala sesuatu yang dimasukkan ke dalam sistem untuk diolah. Masukan dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). Contoh pada sistem komputer, program adalah *maintenance input* yang digunakan untuk mengoperasikan komputer, sedangkan data adalah *signal input* yang akan diolah menjadi informasi.

6. Keluaran Sistem

Keluaran sistem adalah hasil dari proses atau pengolahan masukan (*input*). Hasil masukan sistem yang diolah dan diklasifikasikan akan menjadi keluaran yang merupakan masukan bagi subsistem lain.

7. Pengolah Sistem (*process*)

Pengolah sistem merupakan suatu bagian dari pengolahan yang merubah masukan menjadi keluaran. Bagian ini bertanggung jawab untuk

memproses masukan menjadi keluaran. Pada pengolahan sistem melibatkan proses atau aktivitas yang terjadi dalam sistem.

8. Sasaran Sistem

Sasaran sistem adalah tujuan atau hasil yang diinginkan atau dicapai oleh sistem. komponen-komponen saling bekerja sama atau berkolaborasi untuk mencapai sasaran sistem tersebut. Sistem memiliki sasaran yang pasti dan bersifat deterministik. Suatu sistem dikatakan berhasil jika mengenai sasaran atau tujuan yang telah direncanakan.

2.1.2 Klasifikasi Sistem

Dalam buku Erni Widarti, Joosten, Putu Yudia Pratiwi (2024:8) dengan judul Buku Ajar Pengantar Sistem Ekonomi Indonesia. Klasifikasi sistem merupakan suatu bentuk kesatuan antara satu komponen dengan komponen lainnya dalam sistem. Sistem dapat diklasifikasikan berdasarkan berbagai kriteria tergantung konteksnya. Secara umum sistem dapat diklasifikasikan sebagai berikut:

1. Sistem Tertutup (*Close System*)

Sistem tertutup merupakan sistem yang tidak berinteraksi dengan lingkungannya dan tidak menerima *input eksternal*. Operasinya *independen* dan tidak terpengaruh oleh perubahan lingkungan *eksternal*.

2. Sistem Terbuka (*Open System*)

Sistem terbuka merupakan sistem yang selalu berhubungan dengan lingkungan luarnya untuk melakukan proses yang menghasilkan keluaran

(*output*). Sistem ini rentan terhadap perubahan lingkungan, dapat beradaptasi, memiliki batasan dan antarmuka dengan lingkungan.

3. Sistem Alamiah (*Natural System*)

Sistem alamiah merupakan sistem yang terdapat dalam alam dan melibatkan unsur-unsur alam seperti organisme hidup dan elemen fisik. Sistem ini beroperasi sesuai dengan hukum-hukum alam dan memiliki kompleksitas intrinsik.

4. Sistem Buatan Manusia (*Human Made System*)

Sistem yang dibuat atau dirancang oleh manusia untuk mencapai tujuan tertentu. Sistem ini bergantung pada teknologi dan desain manusia seperti mesin, perangkat lunak, bangunan, dll

5. Sistem Konseptual atau Abstrak (*Conceptual/Abstract System*)

Sistem yang terdiri dari konsep atau ide-ide tanpa keberadaan fisik yang nyata. Sistem ini digunakan untuk mewakili ide atau model dalam bentuk konseptual.

6. Sistem Fisik (*Physical System*)

Sistem yang memiliki keberadaan fisik atau nyata yang terdiri dari elemen-elemen yang dapat diidentifikasi. Sistem ini melibatkan komponen fisik seperti perangkat keras komputer, mesin atau infrastruktur lainnya.

7. Deterministik Sistem

Sistem yang perilakunya sepenuhnya dapat diprediksi dan ditentukan oleh kondisi awal atau input.

8. Probabilistik Sistem

Sistem dimana terdapat unsur ketidakpastian atau probabilitas dalam perilakunya. Keluaran atau output dalam sistem ini diungkapkan sebagai distribusi probabilitas dan prediksi bersifat statistik.

9. Sistem Artifisial (*Artificial System*)

Sistem yang dibuat oleh manusia dan melibatkan teknologi atau kecerdasan buatan. Contohnya termasuk perangkat lunak komputer, robot atau sistem *AI*.

10. Sistem Penjelasan Tingkah Laku (*Manned System*)

Sistem yang melibatkan intervensi atau partisipasi manusia dalam pengoperasiannya. Sistem ini memerlukan kehadiran atau kontrol manusia untuk fungsi yang optimal

2.2 Sistem Pengambilan Keputusan

Sistem Pendukung Keputusan biasanya dibangun untuk mendukung solusi atas suatu masalah atau untuk suatu peluang. SPK dapat juga didefinisikan sebagai sistem basis model yang terdiri dari prosedur-prosedur dalam pemrosesan data yang pertimbangannya untuk membantu manajer dalam mengambil keputusan [2].

Sistem Pengambilan keputusan merupakan suatu proses pemilihan alternatif terbaik dari beberapa alternatif secara sistematis untuk ditindaklanjuti (digunakan) sebagai suatu cara pemecahan masalah [3].

2.2.1 Komponen Sistem Pengambilan Keputusan

Adapun komponen-komponen dari SPK adalah sebagai berikut [3]:

1. Data Management

Termasuk *database*, yang mengandung data yang relevan untuk berbagai situasi dan diatur oleh *software* yang disebut *Database Management System* (DBMS).

2. Model Management

Melibatkan model finansial, statistik, *management science*, atau berbagai model kualitatif lainnya, sehingga dapat memberikan ke sistem suatu kemampuan analitis, dan manajemen *software* yang dibutuhkan.

3. Communication

User dapat berkomunikasi dan memberikan perintah pada DSS melalui subsistem ini. Ini berarti menyediakan antarmuka.

4. Knowledge Management

Subsistem *optional* ini dapat mendukung subsistem lain atau bertindak atau bertindak sebagai komponen yang berdiri sendiri.

2.3 Evaluasi Kinerja

Dalam buku Rina (2021:137) dengan judul Kinerja Karyawan, untuk melakukan evaluasi kinerja perlu dilakukan observasi terhadap unsur-unsur yang ada dalam setiap pekerjaan yang akan dijadikan sebagai sumber penilaian kinerja karyawan baik individu maupun secara tim atau kelompok. Pengukuran dari setiap penilaian yang dilakukan akan memberikan hasil evaluasi terhadap tingkat

prestasi karyawan berdasarkan dari pengamatan-pengamatan yang dilakukan, catatan perolehan prestasi yang dicapai oleh karyawan. Hasil evaluasi ini akan membantu para pemimpin organisasi atau perusahaan akan lebih cepat dalam mengambil keputusan yang berkaitan dengan kinerja karyawan. Berdasarkan hasil evaluasi tersebut manajemen akan menentukan kebijakan-kebijakan yang tujuannya dapat mengembangkan kinerja organisasi berdasarkan dari data kinerja sumber daya manusia khususnya karyawan. Evaluasi kinerja selalu membutuhkan suatu standar kinerja, tanpa adanya standar kinerja ini maka ukuran yang digunakan dalam melakukan perbandingan-perbandingan yang berkaitan dengan kinerja karyawan. Evaluasi kinerja tanpa memiliki standar kinerja maka evaluasi tersebut tidak akan memiliki nilai yang berfungsi bagi seorang pemimpin dalam mengambil suatu keputusan. Standar kinerja memiliki banyak fungsi yang diantaranya adalah dapat digunakan dalam menentukan besaran standar gaji karyawan, dengan standar tersebut beban pekerjaan dan tanggung jawab yang diemban oleh karyawan dapat dirumuskan oleh perusahaan.

Menurut [5] Evaluasi kinerja digunakan untuk kepentingan evaluasi tujuan dan sasaran, evaluasi rencana, evaluasi pengukuran kinerja, evaluasi hasil, evaluasi lingkungan, dan evaluasi proses kinerja.

1. Evaluasi tujuan dan sasaran, evaluasi terhadap tujuan dimaksudkan untuk mengetahui apakah tujuan yang telah ditetapkan sebelumnya dapat tercapai atau tidak. evaluasi terhadap sasaran untuk mengukur pencapaian sasaran terhadap target yang telah ditetapkan. Evaluasi tujuan dan sasaran memberikan umpan balik bagi proses perencanaan.

2. Evaluasi rencana, untuk menilai apakah hasil yang dicapai telah sesuai dengan apa yang direncanakan.
3. Evaluasi lingkungan, untuk menilai kondisi lingkungan yang dihadapi saat proses pelaksanaan kinerja, dan dapat mengakibatkan kesulitan dan kegagalan dalam mencapai hasil kinerja.
4. Evaluasi proses kinerja, untuk menilai apakah terdapat kendala dalam proses kinerja.
5. Evaluasi pengukuran kinerja, untuk menilai apakah pengukuran kinerja telah dilakukan dengan benar, sistem review dan coaching telah berjalan dengan semestinya, serta apakah metode yang digunakan dalam pengukuran sudah tepat dan benar dilakukan.
6. Evaluasi hasil, untuk menilai hasil kinerja organisasi, kelompok, maupun individu masing-masing pekerja

2.4 Metode TOPSIS

Menurut [6] Metode TOPSIS menggunakan prinsip bahwa alternatif yang terpilih harus mempunyai jarak terdekat dari solusi ideal positif dan jarak terjauh dari solusi ideal negatif, kemudian diurutkan berdasarkan nilai kedekatan relatif sehingga alternatif yang memiliki jarak terpendek dengan solusi ideal positif adalah alternatif terbaik, dengan kata lain alternatif yang memiliki nilai yang lebih besar itulah yang lebih baik untuk dipilih. Topsis adalah metode pengambilan keputusan multi kriteria dengan dasar alternatif yang dipilih memiliki jarak terdekat dengan solusi ideal positif dan memiliki jarak terjauh dari solusi ideal

negatif. Namun, alternatif yang mempunyai jarak terkecil dari solusi ideal positif, tidak harus mempunyai jarak terbesar dari solusi ideal *negative*.

Menurut [7] Metode *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS) adalah salah satu metode pengambilan keputusan multikriteria yang pertama kali diperkenalkan oleh Yoon dan Hwang. TOPSIS menggunakan prinsip bahwa alternatif yang terpilih harus mempunyai jarak terdekat dari solusi ideal positif dan terjauh dari solusi ideal negatif dari sudut pandang geometris dengan menggunakan jarak *Euclidean* untuk menentukan kedekatan relative dari suatu alternatif dengan solusi optimal. Solusi ideal positif didefinisikan sebagai jumlah dari seluruh nilai terbaik yang dapat dicapai untuk setiap atribut, sedangkan solusi negatif-ideal terdiri dari seluruh nilai terburuk yang dicapai untuk setiap atribut.

Dalam menggunakan metode TOPSIS terdapat prosedur-prosedur yang harus dilakukan, diantaranya adalah :

- a. Membuat matriks keputusan yang ternormalisasi
- b. Membuat matriks keputusan yang ternormalisasi terbobot
- c. Menentukan matriks solusi ideal positif dan solusi ideal negatif
- d. Menentukan jarak antara nilai setiap alternatif dengan matriks solusi ideal positif dan negatif.

Langkah-langkah yang harus dilakukan untuk memperoleh hasil penilaian dalam metode TOPSIS adalah :

1. Membangun normalized decision matrix

Elemen r_{ij} hasil dari normalisasi decision matrix R dengan metode *Euclidean length of a vector* adalah:

$$R_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}}$$

2. Membangun weighted normalized decision matrix

Solusi ideal positif A^+ dan solusi ideal negatif A^- dapat ditentukan berdasarkan rating bobot ternormalisasi (Y_{ij}) sebagai:

$$Y_{ij} = W_i R_{ij}$$

dengan $i=1,2,3, \dots, m$ dan $j=1,2,3, \dots, n$

3. Menentukan Matriks Solusi Ideal dan Matriks Solusi Ideal Negatif

Solusi ideal positif (A^+) dihitung berdasarkan:

$$A^+ = (Y_1^+, Y_2^+, Y_3^+, \dots, Y_n^+)$$

Solusi ideal negatif (A^-) dihitung berdasarkan:

$$A^- = (Y_1^-, Y_2^-, Y_3^-, \dots, Y_n^-)$$

4. Menentukan Jarak antara Nilai Setiap Alternatif dengan Matriks Solusi Ideal Positif dan Matrik Ideal Negatif.

Jarak antara alternatif A_i dengan solusi ideal positif dirumuskan sebagai:

$$D_i^+ = \sqrt{\sum_{j=1}^n (Y_{ij} - Y_i^+)^2}, i = 1, 2, 3, \dots, m$$

Jarak antara alternatif A_i dengan solusi ideal negatif dirumuskan sebagai:

$$D_i^- = \sqrt{\sum_{j=1}^n (Y_{ij} - Y_i^-)^2}, i = 1, 2, 3, \dots, m$$

5. Menentukan Nilai Preferensi untuk Setiap Alternatif

Kedekatan setiap alternatif terhadap solusi ideal dihitung berdasarkan rumus:

$$V = \frac{Di^-}{Di^- + Di^+}, i = 1, 2, 3, \dots, m$$

2.5 Alat Bantu Perancangan Sistem

Adapun alat bantu yang harus digunakan dalam perancangan atau pengembangan sistem yang digunakan dalam penelitian umumnya berupa gambaran atau diagram sebagai berikut :

2.5.1 ML (*Unified Modeling Language*)

Dalam buku Niqotaini, Bambang Saras Yulistiawan, Erly Krisnanik dan Rifka Dwi Amalia (2023:1) dengan judul analisa dan perancangan sistem informasi dengan *unified modelling language*, UML biasa disebutkan sebagai arsitektur untuk menyajikan visualisasi, mengkontruksi, memberikan dokumentasi, memberikan informasi proses software berupa model dan deskripsi. Unified modeling language membantu mengurangi masalah dengan memberikan metode

UML (*Unified Modeling Language*) adalah sebuah bahasa untuk menentukan, visualisasi, kontruksi, dan mendokumentasikan *artifact* (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak). *Artifact* dapat berupa model, deskripsi atau perangkat lunak dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem nonperangkat lunak lainnya.

UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, tetapi hampir dalam semua bidang yang membutuhkan pemodelan. [9].

2.5.2 Bagian-Bagian UML

Bagian-bagian utama dari UML adalah *view*, *diagram*, *model element*, dan *general mechanism*.

a. View

View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. View bukan melihat grafik, tetapi merupakan suatu abstraksi yang berisi sejumlah diagram. Beberapa jenis *view* dalam UML antara lain *use case view*, *logical view*, *component view*, *concurrency view*, dan *deployment view*.

b. Use Case View

Mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan sesuai yang diinginkan external actors. Actor yang berinteraksi dengan sistem yang dapat berupa user atau sistem lainnya. View ini digambarkan dalam use case diagram dan kadangkadang dengan *activity diagrams*. View ini digunakan terutama untuk pelanggan, perancang (*designer*), pengembang (*developer*), dan penguji sistem (*tester*).

c. Logical View

Mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis (*class*, *object*, dan *relationship*) dan kolaborasi dinamis yang terjadi ketika *object* mengirim pesan ke *object* lain dalam suatu fungsi tertentu. *View* ini digambarkan dalam *class diagrams* untuk struktur statis dan dalam *state*, *sequence*, *collaboration*, dan *activity diagram* untuk model dinamisnya. *View* ini digunakan untuk perancang (*designer*) dan pengembang (*developer*).

d. Component View

Mendeskripsikan implementasi dan ketergantungan modul. Komponen yang merupakan tipe lainnya dari *code module* diperlihatkan dengan struktur dan ketergantungannya juga alokasi sumber daya komponen dan informasi administratif lainnya. *View* ini digambarkan dalam *component view* dan digunakan untuk pengembang (*developer*).

e. Concurrency View

Membagi sistem ke dalam proses dan prosesor. *View* ini digambarkan dalam diagram dinamis (*state*, *sequence*, *collaboration*, dan *activity diagrams*) dan diagram implementasi (*component* dan *deployment diagrams*) serta digunakan untuk pengembang (*developer*), pengintegrasi (*integrator*), dan penguji (*tester*).

f. Deployment View

Mendeskripsikan fisik dari sistem seperti komputer dan perangkat (*nodes*) dan bagaimana hubungannya dengan lainnya. *View* ini digambarkan dalam

deployment diagrams dan digunakan untuk pengembang (*developer*), pengintegrasian (*integrator*), dan pengujian (*tester*).

g. Diagram

Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu view tertentu dan ketika digambarkan biasanya dialokasikan untuk view tertentu. Adapun jenis diagram antara lain:

1. Use Case Diagram

Dalam buku Niqotaini, Bambang Saras Yulistiawan, Erly Krisnanik dan Rifka Dwi Amalia (2023:2) *Usecase* juga menggambarkan fungsionalitas dalam sistem maupun kelas. Menjelaskan bagaimana interaksi sistem yang terjadi dengan dunia luar dan juga fungsionalitas perangkat lunak atau sistem yang dapat dilihat user. Umumnya usecase dibuat ketika awal pengembangan sebuah sistem atau perangkat lunak. *Usecase* menjelaskan serta menggambar apa sajakah yang dilakukan pada sistem serta siapakah yang dapat menerapkan fungsi-fungsi dalam sistem tersebut. *Usecase* juga memberikan penggambaran hubungan interaksi yang terjadi antar sistem dengan user.

Tabel 2.1 Simbol Use case

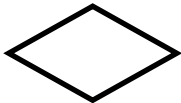
No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Orang, sistem atau proses ketika berinteraksi dengan usecase atau pun sistem yang dibuat.
3		<i>Generalization</i>	Relasi umum dan tertentu (generalisasi dan spesialisasi) antara dua buah usecase. Salah satunya mempunyai fungsi global apabila dibandingkan usecase lainnya. Arah panah mengarah ke usecase dimana dapat menjadi umum atau generalisasinya
4		<i>Include</i>	Menghubungkan usecase tambahan yang menuju usecase yang ditambahkan. Usecase yang ditambahkan akan membutuhkan usecase ini agar dapat berjalan untuk fungsinya atau persyaratan agar dapat menjalankan fungsi dalam usecase tersebut.
5		<i>Extend</i>	Menghubungkan dari usecase tambahan yang menuju ke usecase yang tambah. Usecase yang ditambahkan mampu sendiri berdiri walaupun tanpa usecase tambahan. Arah panah extend menuju kepada usecase yang ditambah.
6		<i>Association</i>	Menghubungkan antar aktor dan usecase dimana saling berinteraksi dalam usecase diagram atau aktor.

No	Gambar	Nama	Keterangan
			Asosiasi ini dapat diartikan sebagai link setiap elemen-elemen
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Fungsionalitas ataupun aksi-aksi yang ditampilkan sistem sebagai unit yang saling bertukar pesan antar unit maupun aktor.

2. Activity Diagram

Dalam buku Niqotaini, Bambang Saras Yulistiawan, Erly Krisnanik dan Rifka Dwi Amalia (2023:19) Fungsi activity diagram adalah menunjukkan urutan aktivitas seperti apa proses yang terjadi pada sebuah sistem dan memberikan informasi proses seluruh Activity diagram dapat dibuat berdasar satu atau lebih usecase. Diagram aktivitas ini memperjelas urutan kegiatan-kegiatan pada usecase diagram 20 yang dibuat sebelumnya. Menggunakan activity diagram juga mempermudah untuk mendapatkan gambaran yang lebih menyeluruh pada proses bisnis.

Tabel 2.2 Simbol Activity diagram

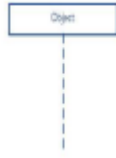
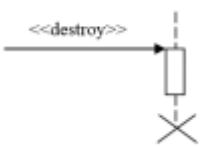
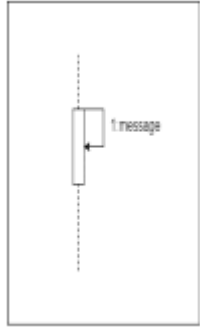
No	Gambar	Nama	Keterangan
1		<i>Activity</i>	Pada awalnya dengan kata kerja
2		Percabangan <i>/ Decision</i>	Proses memilih aktivitas lebih dari satu
3		<i>Initial Node</i>	Status awal pada diagram aktivitas. Umumnya activity diagram memiliki

No	Gambar	Nama	Keterangan
			satu status atau titik awal. Namun ada juga sebuah activity diagram memiliki lebih dari satu status atau titik awal
4		<i>Activity Final Node</i>	Status akhir, mempunyai status akhir. Simpul akhir ini menandakan bahwa urutan aktivitas telah berhenti atau selesai
5		<i>Swimlane</i>	Pemisahan terhadap organisasi yang bertanggung jawab dalam aktivitas. Pengelompokkan aktivitas didasarkan oleh aktivitas aktor dalam sebuah urutan yang sama

3. Sequence Diagram

Dalam buku Niqotaini, Bambang Saras Yulistiawan, Erly Krisnanik dan Rifka Dwi Amalia (2023:28) Sequence diagram juga menggambarkan aktivitas situs web selama menampilkan aliran pesan yang masuk pada objek web tersebut. Sequence diagram memberikan penggambaran perilaku objek dalam usecase yang dapat menggambarkan siklus hidup pada objek dan message yang dikirim dan diterima dari sebuah objek. Saat menggambarkan sequence diagram, metode yang dimiliki class dan objek yang terlibat harus perlu diperhatikan.

Tabel 2.3 Simbol *Sequence Diagram*

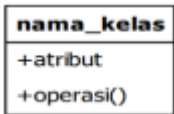
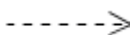
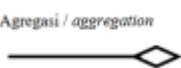
No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Orang, sistem atau proses ketika berinteraksi dengan usecase atau pun sistem yang dibuat.
2	 Garis hidup / <i>lifeline</i>	Garis Hidup (<i>Lifeline</i>)	Menyatakan kehidupan suatu objek
3		<i>Object</i>	Menyatakan objek yang berinteraksi pesan
4		Pesan tipe <i>create</i>	menyatakan suatu objek membuat sesuatu, arah panah mengarah pada objek yang dibuat
5		Pesan bertipe <i>destroy</i>	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy
6		Pesan bertipe <i>return</i>	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian .
7		Pesan bertipe <i>self</i>	Komponen ini berbentuk panah dan garis putus-putus yang memiliki fungsi untuk

No	Gambar	Nama	Keterangan
			menjelaskan relasi antar objeknya sendiri.

4. Class Diagram

Dalam buku Niqotaini, Bambang Saras Yulistiawan, Erly Krisnanik dan Rifka Dwi Amalia (2023:38) Kelas dan juga paket-paket yang berada dalam sebuah perangkat lunak ditampilkan dalam sebuah diagram yang disebut dengan class diagram. Gambaran dari sistem secara luas baik itu kelas maupun relasinya ditampilkan dalam sebuah class diagram. Sifat dari class diagram ini adalah statis dan memberikan penggambaran tentang relasi apa yang terjadi. Sistem berbasis objek umumnya menggunakan diagram seperti class diagram. Struktur class yang ada dalam sistem dapat di visualisasikan dalam diagram kelas. Diagram kelas mencakup sekumpulan interface, class, relasi, dan collaborations yang terdapat didalamnya. Dalam proses analisis, diagram kelas ini mengamati aturan serta tanggung jawab entitas dalam perilaku sistem. Peran diagram kelas pada tahap design adalah mengetahui struktur dari kelas-kelas yang terbentuk kedalam arsitektur sistem yang dibangun.

Tabel 2.4 Simbol *Class diagram*

No	Gambar	Nama	Keterangan
1		<i>Generalization</i>	Relasi antar kelas dengan makna generalisasipesialisasi
2		<i>Public Association</i>	Asosiasi lebih dari dua objek dapat dihindari.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Package</i>	Package merupakan sebuah bungkus dari satu atau lebih kelas
5		Antar muka / <i>Interface</i>	Sama dengan konsep interface dalam pemrograman berorientasi objek
6		<i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas
7		<i>Aggregation / Agrepgasi</i>	Relasi antar kelas dengan makna semua bagian (whole-part)
8		<i>Association</i>	Relasi dari objek satu dengan lainnya.

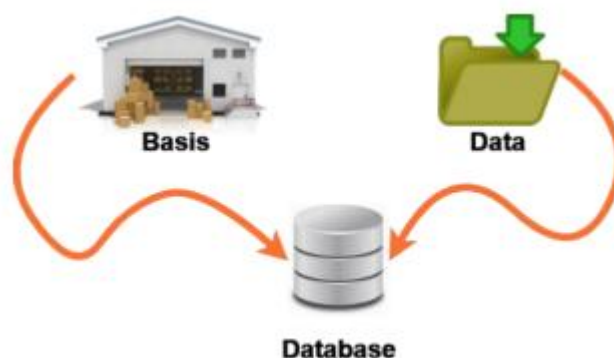
2.5.3 Tujuan Penggunaan UML

- Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
- Menyatukan praktik-praktik terbaik yang terdapat dalam pemodelan.
- Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah serta dimengerti secara umum.

- d. UML bisa juga berfungsi sebagai sebuah cetak biru (*blue print*) karena sangat lengkap dan detail. Dengan cetak biru ini maka akan bisa diketahui informasi secara detail tentang coding program atau bahkan membaca program dan menginterpretasikan kembali ke dalam bentuk diagram (*reverse engineering*).

2.6 Database

Dalam buku Erni Widarti, Joosten, Putu Yudia Pratiwi (2024:85) dengan judul Pengantar Sistem Informasi. *Database* atau basis data berasal dari 2 kata yaitu Basis dan Data. BASIS, dalam konteks ini, dapat dijelaskan sebagai pusat operasi atau gudang yang berfungsi sebagai tempat berkumpul atau bersarang suatu kelompok atau organisasi. Ini adalah lokasi sentral yang digunakan untuk kegiatan-kegiatan tertentu, seperti perencanaan, pengorganisasian, atau penyimpanan bahan atau informasi penting. Dengan kata lain, BASIS merupakan pusat atau markas yang menjadi titik fokus untuk kegiatan-kegiatan tertentu yang melibatkan kumpulan orang atau sumber daya.



Gambar 2.1 Konsep Database

DATA dapat dijelaskan sebagai representasi fakta dari dunia nyata yang mencerminkan suatu objek atau entitas, seperti manusia (seperti pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya. Representasi ini diwujudkan dalam berbagai bentuk, termasuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasi dari elemen-elemen tersebut. Dengan kata lain, data adalah informasi yang terstruktur yang dapat direpresentasikan dalam format yang dapat diolah dan dimengerti oleh sistem atau manusia.

Dalam buku Erni Widarti, Joosten, Putu Yudia Pratiwi (2024:86) dengan judul Pengantar Sistem Informasi. *Database* adalah kumpulan data yang terorganisir dan saling berhubungan secara logis, di mana setiap elemen atau entitas memiliki deskripsi atau atribut tertentu. Informasi dalam *database* disimpan bersama-sama dengan susunan tertentu yang dirancang untuk memenuhi kebutuhan informasi suatu organisasi.

Menurut Connolly dan Begg, dalam [10] *database* adalah suatu kumpulan data yang berhubungan secara logika dan secara deskripsi dari data-data yang dirancang untuk memenuhi kebutuhan informasi dalam suatu organisasi. *Database* menawarkan keuntungan penyimpanan data dengan format yang *independen* dan *fleksibel*.

2.6.1 Elemen-Element Database

Dalam buku Erni Widarti, Joosten, Putu Yudia Pratiwi (2024:88) dengan judul Pengantar Sistem Informasi. Element-elemen dalam *database* terdiri dari 3 jenis yaitu *tabel*, *record* dan *field*.

1. Tabel

Sebuah tabel dalam basis data adalah kumpulan data terkait yang diorganisir dalam format tabular, terdiri dari baris dan kolom. Setiap baris mewakili rekaman unik, dan setiap kolom mewakili bidang atau atribut tertentu. Tabel-tabel ini merupakan elemen dasar dari basis data relasional, di mana mereka berfungsi sebagai sarana utama untuk menyimpan dan mengorganisir data. Tabel digunakan untuk menyimpan berbagai jenis informasi, seperti detail karyawan, inventaris produk, atau pesanan pelanggan. Struktur sebuah tabel ditentukan oleh kolom-kolomnya dan tipe data yang mereka *contain*. Selain itu, tabel dapat memiliki metadata terkait, seperti batasan dan kunci, untuk menegakkan integritas data dan hubungan.

2. Record

Sebuah rekaman dalam basis data adalah kumpulan item data terkait yang disimpan dalam sebuah tabel. Seringkali disebut sebagai baris dalam sebuah tabel dan berisi informasi spesifik tentang entitas tertentu, seperti orang, barang, atau objek. Setiap rekaman terdiri dari serangkaian kolom, masing-masing berisi data tertentu. Dalam basis data relasional, sebuah rekaman adalah kelompok data terkait yang disimpan dalam struktur yang sama. Istilah "rekaman" dan "baris" sering digunakan secara bergantian untuk merujuk pada konsep yang sama dalam konteks basis data. Rekaman menyediakan cara praktis untuk menyimpan dan mengambil data dari basis data, dan mereka dapat dengan mudah dibuat, dimodifikasi,

dan dihapus tanpa memengaruhi data lain dalam basis data. Selain itu, sebuah kunci primer biasanya digunakan untuk mengidentifikasi secara unik setiap rekaman dalam sebuah tabel.

3. Field

Dalam basis data, sebuah *field* adalah bagian data spesifik yang disimpan dalam sebuah rekaman atau baris dari sebuah tabel. *Field* juga dikenal sebagai kolom atau atribut. *Fields* digunakan untuk menentukan struktur sebuah tabel dan untuk menyimpan jenis data tertentu, seperti nama, tanggal, atau angka. Setiap *field* memiliki nama, tipe data, dan panjang, yang menentukan jumlah karakter atau digit maksimum yang dapat disimpan dalam *field* tersebut. *Fields* juga dapat memiliki properti tambahan, seperti batasan dan kunci, untuk menegakkan integritas data dan hubungan. Penggunaan *fields* memungkinkan penyimpanan, pengambilan, dan manipulasi data yang efisien dalam suatu sistem basis data.

2.6.2 Jenis-Jenis Database

Dalam buku Erni Widarti, Joosten, Putu Yudia Pratiwi (2024:90) dengan judul Pengantar Sistem Informasi. Berdasarkan hasil pencarian, terdapat beberapa jenis *database* yang umum digunakan, antara lain:

1. *Relational Database*:

Jenis *database* yang paling umum digunakan, di mana data disimpan dalam tabel yang saling berhubungan.

2. *Hierarchical Database:*

Database dengan struktur hirarkis, di mana data disusun dalam bentuk pohon.

3. *Object Oriented Database:*

Database yang menggunakan konsep pemrograman berorientasi objek untuk menyimpan data.

4. *Network Database:*

Database yang menggunakan struktur jaringan untuk menyimpan dan mengakses data.

5. *NoSql Database:*

Basis data *NoSQL (Not Only SQL)* adalah jenis basis data yang dirancang untuk mengelola dan menyimpan data yang tidak terstruktur atau semi-terstruktur. Berbeda dengan basis data relasional, basis data *NoSQL* tidak memanfaatkan struktur tabel yang kaku dan menggunakan model data yang lebih fleksibel. Ini memungkinkan penyimpanan dan pengelolaan data yang tidak memerlukan skema yang telah ditentukan sebelumnya

Setiap jenis *database* memiliki fungsinya masing-masing, dan pemilihan jenis *database* yang tepat sangat bergantung pada kebutuhan dan tujuan penggunaannya.

2.7 MySQL

MySQL ialah *software* sistem manajemen DBMS yang *multiuser*. Karena sifatnya yang *open source* dan memiliki kemampuan menampung kapasitas yang

sangat besar, maka *MySQL* menjadi *database* yang sangat populer di kalangan programmer web. *MySQL* merupakan sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, *multiuser* serta menggunakan perintah standar SQL (*Structured Query Language*) [11].

MySQL adalah salah satu jenis *database* yang banyak digunakan untuk membuat aplikasi berbasis web yang dinamis. *MySQL* termasuk jenis RDBMS (*Relational Database Management Sistem*). *MySQL* ini mendukung Bahasa pemrograman PHP. *MySQL* juga mempunyai *query* atau bahasa SQL (*Structured Query Language*) yang simple dan menggunakan *escape character* yang sama dengan PHP. *MySQL* adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan *MySQL*, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial [12].



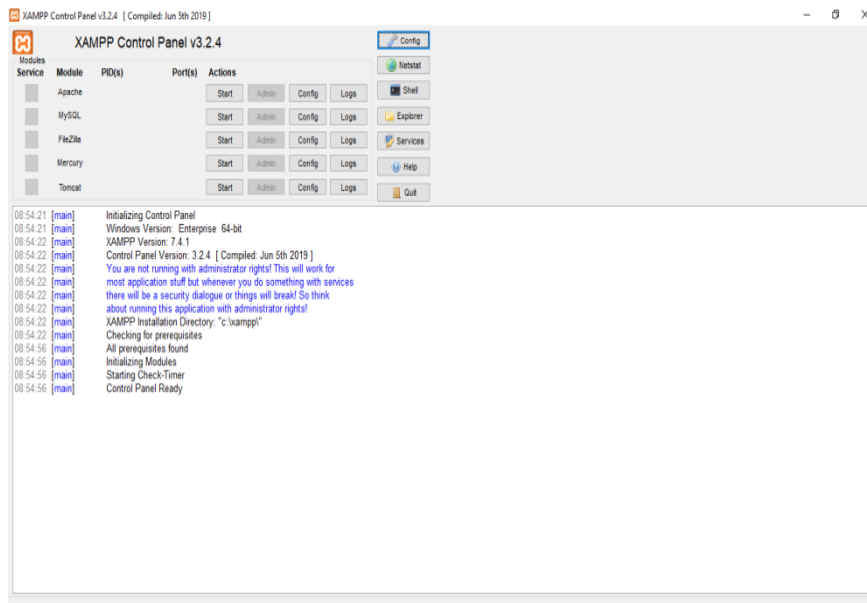
Gambar 2.2 MySQL

2.8 Xampp

Menurut [13] XAMPP merupakan paket PHP berbasis *open source* yang dikembangkan oleh sebuah komunitas *Open Source*. Dalam menggunakan XAMPP sudah disediakan berbagai kebutuhan sehingga tidak perlu melakukan penginstalan program lainnya. Beberapa paket yang sudah disediakan adalah *Phpmyadmin*, *Filezilla*, PHP, MySQL, dan *Apache*.

XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP, dan Perl. XAMPP adalah *tool* yang menyediakan paket perangkat lunak dalam satu buah paket. Dalam paket XAMPP sudah terdapat Apache (*web server*), MySQL (*database*), PHP (*server side scripting*), Perl, FTP server, PhpMyAdmin dan berbagai pustaka bantu lainnya. Dengan menginstal XAMPP maka anda tidak perlu lagi melakukan instalasi dan melakukan konfigurasi *web server* Apache, PHP, dan MySQL secara manual. XAMPP akan otomatis menginstalasi dan mengonfirmasi untuk anda. [14].

Dari beberapa pengertian diatas dapat ditarik kesimpulan bahwa, Xampp adalah sebuah *software* yang didalamnya terdapat *database* sehingga memudahkan dalam pengoperasiannya.



Gambar 2.3 tampilan XAMPP

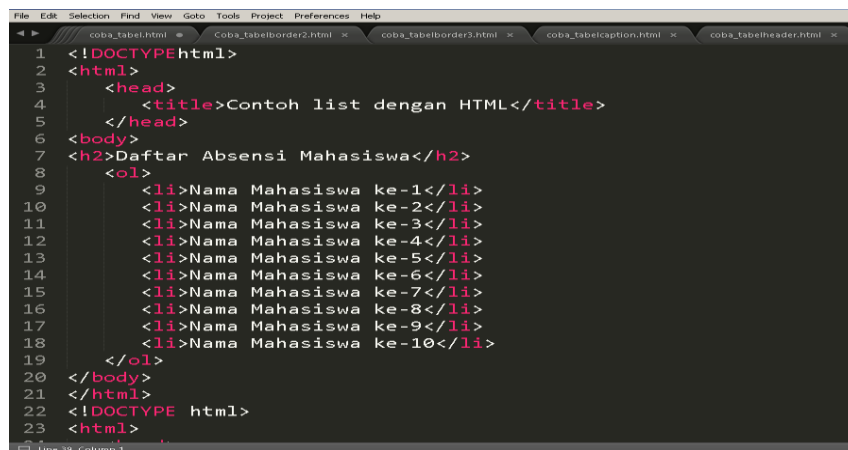
2.9 PHP

PHP merupakan salah satu bahasa pemrograman *opensource* yang digunakan untuk membuat website. PHP adalah singkatan dari *Hypertext Pre-processor*, yang sebelumnya disebut *Personal Home Pages*. PHP adalah bahasa pemrograman umum yang berarti php dapat disematkan ke dalam kode HTML, atau dapat digunakan dalam kombinasi dengan berbagai sistem template web sistem manajemen konten web, dan kerangka kerja web. [15].

Menurut [16] PHP merupakan bahasa pemrograman untuk *script web server – side*. PHP memiliki beberapa kelebihan yang tidak dimiliki oleh bahasa - bahasa sejenisnya, yaitu:

- 1) Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.

- 2) *Web Server* yang mendukung PHP dapat ditemukan dimana – mana dari mulai IIS sampai dengan apache, dengan konfigurasi yang relatif
- 3) Dalam sisi pengembangan lebih mudah, karena banyaknya *milis - milis* dan *developer* yang siap membantu dalam pengembangan.
- 4) Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena referensi yang banyak.
- 5) PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (*linux, unix, windows*) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah - perintah sistem



```

1  <!DOCTYPEhtml>
2  <html>
3    <head>
4      <title>Contoh list dengan HTML</title>
5    </head>
6    <body>
7      <h2>Daftar Absensi Mahasiswa</h2>
8      <ol>
9        <li>Nama Mahasiswa ke-1</li>
10       <li>Nama Mahasiswa ke-2</li>
11       <li>Nama Mahasiswa ke-3</li>
12       <li>Nama Mahasiswa ke-4</li>
13       <li>Nama Mahasiswa ke-5</li>
14       <li>Nama Mahasiswa ke-6</li>
15       <li>Nama Mahasiswa ke-7</li>
16       <li>Nama Mahasiswa ke-8</li>
17       <li>Nama Mahasiswa ke-9</li>
18       <li>Nama Mahasiswa ke-10</li>
19     </ol>
20   </body>
21 </html>
22 <!DOCTYPE html>
23 <html>

```

Gmabar 2.4 Tampilan PHP

2.10 Website

Menurut Dina Amelia dalam [17] Website dapat diartikan sebagai suatu kumpulan-kumpulan halaman yang menampilkan berbagai macam informasi teks, data, gambar diam maupun bergerak, data animasi, suara, video maupun gabungan dari semuanya, baik itu yang bersifat statis maupun yang dinamis,

dimana membentuk satu rangkaian bangunan yang saling berkaitan dimana masing-masing dihubungkan dengan jaringan halaman atau *hyperlink*. Definisi secara umum, website adalah kumpulan dari berbagai macam halaman situs yang terangkum di dalam sebuah domain atau subdomain, yang berada di dalam WWW (*World Wide Web*) dan tentunya terdapat di dalam Internet. Halaman website biasanya berupa dokumen yang ditulis dalam *format Hyper Text Markup Language* (HTML).

Jenis-jenis *website* berdasarkan sifatnya adalah [18]:

1. *Website* Dinamis

Website dinamis memiliki isi kontennya sudah termanajemen dengan kata lain sudah menerapkan sistem *Content Management System* (CMS) untuk informasi yang disampaikan. Seseorang bisa mengubah konten dari halaman tertentu menggunakan *browser*. Permintaan dari pengguna dapat diproses oleh *server* yang kemudian ditampilkan dalam isi yang berbeda-beda menurut alur programnya.

2. *Website* Statis

Website statis adalah *website* dimana penggunanya tidak bisa mengubah konten dari *web* tersebut secara langsung menggunakan *browser*. Interaksi yang terjadi antara pengguna dan *server* hanyalah seputar pemrosesan *link* saja. Halaman-halaman *web* tersebut tidak memiliki *database*. Data dan informasi yang ada pada *web* statis tidak berubah-ubah, kecuali diubah sintaksnya. Dokumen *web* yang dikirim kepada *client* akan sama isinya dengan apa yang ada di *web server*.

Dari beberapa pengertian diatas dapat ditarik kesimpulan bahwa, *Website* adalah kumpulan dari beberapa halaman – halaman situs yang berada di dalam *World Wide Web (WWW)* yang berisis informasi dalam bentuk data digital baik berupa text, gambar, video, audio, dan animasi lainnya yang di sediakan melalui jalur koneksi internet.

2.11 Knowledge Discovery In Database

Knowledge Discovery In Database (KDD) merupakan metode yang digunakan untuk mencari pengetahuan atau informasi yang belum diketahui dari sebuah *database* [19]. *Knowledge Discovery In Database (KDD)* merupakan keseluruhan proses non-trivial untuk mencari dan mengidentifikasi pola (pattern) dalam data, dimana pola yang ditemukan bersifat sah, baru, dapat bermanfaat dan dapat dimengerti. Teknologi dan Informasi adalah suatu hal yang lumrah bagi masyarakat. Pesatnya perkembangan teknologi dan informasi tentunya memberikan pengaruh besar bagi masyarakat dan juga pastinya bagi pengusaha atau pebisnis dalam kalangsungan bisnisnya, terutama yang berhubungan dengan data. Tidak sedikit segala sesuatu yang berkaitan dengan bisnis, teknologi dan informasi tentunya selalu menggunakan data sebagai sumber informasinya.

2.12 Metodologi Penelitian

Dalam Kamus Besar Bahasa Indonesia (KBBI), metodologi adalah ilmu tentang metode; uraian tentang metode. Sedangkan pengertian penelitian dalam Kamus Besar Bahasa Indonesia (KBBI), penelitian adalah kegiatan pengumpulan,

pengolahan, analisis, dan penyajian data yang dilakukan secara sistematis dan objektif untuk memecahkan suatu persoalan atau menguji suatu hipotesis untuk mengembangkan prinsip-prinsip umum. Metodologi penelitian merupakan cara untuk mengetahui hasil dari sebuah permasalahan yang spesifik dari suatu penelitian. Metodologi penelitian ini dilakukan dengan cara sistematis yang akan digunakan sebagai acuan dalam penelitian.

Manfaat metodologi penelitian adalah:

1. Memudahkan pekerjaan peneliti agar sampai pada tahap pengambilan keputusan atau kesimpulan-kesimpulan.
2. Untuk mengatasi berbagai keterbatasan yang ada, misalnya keterbatasan waktu, biaya, tenaga, etik, dan lain-lain.
3. Kesimpulan yang diambil oleh peneliti dapat terpercaya.
4. Kesimpulan yang diambil dapat digunakan untuk memecahkan permasalahan.