

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Konsep Dasar Penelitian**

##### **2.1.1. Penelitian Terkait**

Pada penelitian yang dilakukan oleh (Oviana, 2019), membahas tentang sistem yang dapat memprediksi apakah pasien tersebut positif atau negatif batu ginjal. Penelitian ini menerapkan teknik data mining untuk menemukan pengetahuan baru dari sekumpulan data yang ada sehingga dapat melakukan prediksi penyakit batu ginjal. Prediksi dilakukan dengan menerapkan prinsip-prinsip algoritma Jaringan Syaraf Tiruan Backpropagation dengan atribut yang didapatkan dari rekam medis pasien. Konfigurasi terbaik yang diterapkan pada arsitektur jaringan adalah dengan 12 neuron input layer, 120 neuron hidden layer, dan 1 neuron output layer dengan nilai learning rate sebesar 0.2 dan 0.5, serta batas error 0.025 dengan fungsi aktivasi sigmoid biner yang menghasilkan akurasi sebesar 98,5507% dan nilai error MSE 0,014493.

Sementara pada penelitian (Laksito, Sismoro, & Triyanto, 2019), membahas tentang ketersediaan pangan yang cukup dan merata merupakan salah satu pilar perwujudan ketahanan pangan. Setiap tahunnya produksi padi di Kabupaten Bantul selalu berubah-ubah. Metode Regresi Linear Berganda adalah metode peramalan yang menggunakan lebih dari dua faktor yang dapat mempengaruhi hasil sehingga dapat menemukan hasil yang maksimal. Dengan metode Regresi Linear Berganda ini di dapatkan mean absolute deviation (MAD) 0,101 dengan data pelatihan dari tahun 2009 –2017. Persamaan Regresi

Linear berganda yang didapatkan yaitu  $Y = 8307,561443282 + 5,9294543706657x_1 + 118,28063200866x_2 + 175,71009241484x_3$ .

Pada penelitian (Sinaga, Jalaluddin, & Solikhun, 2020), membahas tentang jumlah siswa baru pada SMK Swasta Abdi Sejati pada tahun yang akan datang sehingga dapat dilakukan antisipasi untuk memenuhi dan meningkatkan segala aspek kebutuhan proses belajar dan mengajar di sekolah. Data penelitian diperoleh dari Sekolah SMK Swasta Abdi Sejati Kerasaan. Sumber data lima (5) tahun dari tahun 2014 sampai dengan tahun 2018. Penelitian ini menggunakan 5 model arsitektur yaitu 3-22-1, 3-23-1, 3-24-1, 3-26-1 dan 3-28-1. Dari kelima model arsitektur yang digunakan di peroleh satu model arsitektur terbaik 3-22-1 dengan tingkat keakurasian 75%, hal ini karena model ini memiliki MSE pengujian yang lebih kecil dibandingkan model 3-24-1, kendati sama-sama memiliki hasil akurasi yang sama. Berdasarkan model arsitektur terbaik ini akan digunakan untuk memprediksi jumlah siswa baru pada SMK Swasta Abdi Sejati untuk 2 tahun yang akan datang, yakni tahun 2019 hingga tahun 2020.

### **2.1.2. Penerimaan Mahasiswa Baru**

Menurut (Ilyas, Marisa, & Purnomo, 2018), penerimaan mahasiswa baru merupakan salah satu proses yang ada di instansi pendidikan seperti universitas yang berguna untuk menyaring calon mahasiswa yang terpilih sesuai kriteria yang ditentukan oleh universitas tersebut. Pada umumnya proses penerimaan mahasiswa baru dilakukan melalui tahapan pendaftaran, seleksi berkas, dan pengumuman penerimaan mahasiswa.

### **2.1.3. Pengertian Sistem**

Menurut B, Marshall Romney, dan Steinbart, (2015) Sistem adalah rangkaian dari dua atau lebih komponen-komponen yang saling berhubungan, yang berinteraksi untuk mencapai suatu tujuan. Sebagian besar sistem terdiri dari subsistem yang lebih kecil yang mendukung sistem yang lebih besar.

Berdasarkan pengertian diatas dapat disimpulkan bahwa sistem adalah kumpulan dari komponen-komponen yang saling berkaitan satu dengan yang lain untuk mencapai tujuan dalam melaksanakan suatu kegiatan pokok perusahaan.

### **2.1.4. Sistem Prediksi**

Prediksi adalah suatu proses memperkirakan secara sistematis tentang sesuatu yang paling mungkin terjadi di masa depan berdasarkan informasi masa lalu dan sekarang yang dimiliki, agar kesalahannya (selisih antara sesuatu yang terjadi dengan hasil perkiraan) dapat diperkecil. Prediksi tidak harus memberikan jawaban secara pasti kejadian yang akan terjadi, melainkan berusaha untuk mencari jawaban sedekat mungkin yang akan terjadi (Herdianto, 2013).

Pengertian Prediksi sama dengan ramalan atau perkiraan. Menurut kamus besar bahasa Indonesia, prediksi adalah hasil dari kegiatan memprediksi atau meramal atau memperkirakan nilai pada masa yang akan datang dengan menggunakan data masa lalu. Prediksi menunjukkan apa yang akan terjadi pada suatu keadaan tertentu dan merupakan input bagi proses perencanaan dan pengambilan keputusan.

### 2.1.5. Regresi Linier

Pengertian Regresi secara umum adalah sebuah alat statistik yang memberikan penjelasan tentang pola hubungan (model) antara dua variabel atau lebih. Analisis Regresi digunakan untuk membangun persamaan dan menggunakan persamaan tersebut untuk membuat perkiraan (prediction)(Winarso, 2017). Karena merupakan prediksi, maka nilai prediksi tidak selalu tepat dengan nilai riilnya. Dalam Analisis Regresi dikenal 2 jenis variabel yaitu :

1. Variabel Respon disebut juga Variable Dependen yaitu variabel yang keberadaannya dipengaruhi oleh variabel lainnya dan dinotasikan variabel Y.
2. Variabel Prediktor disebut juga Variable Independent yaitu variabel yang bebas (tidak dipengaruhi oleh variabel lainnya) dan dinotasikan X.

#### 2.1.5.1. Regresi Linier Sederhana

Regresi Linier Sederhana untuk mendapatkan hubungan matematis dalam bentuk suatu persamaan antara variabel tak bebas tunggal dengan variabel bebas tunggal. Regresi linier sederhana hanya memiliki satu peubah X yang dihubungkan dengan satu peubah tidak bebas Y. Bentuk umum dari persamaan regresi linier untuk populasi adalah:

$$Y = \alpha + bX + e$$

Keterangan:

Y = Variabel tak bebas (dependent variable)

X = Variabel bebas (independent variable)

$\alpha$  = Konstanta (intercept)

$b$  = Parameter Koefisien Regresi Variabel Bebas

$e$  = Pengamatan Variabel gangguan atau eror

Menentukan koefisien persamaan  $a$  dan  $b$  dapat dengan menggunakan metode kuadrat terkecil, yaitu cara yang dipakai untuk menentukan koefisien persamaan  $a$  dan  $b$  dari jumlah pangkat dua (kuadrat) antara titik – titik dengan garis regresi yang dicari yang terkecil.

Dengan demikian, dapat ditentukan:

$$\alpha = \frac{(\sum Y_i)(\sum X_1^2) - (\sum X_i)(\sum X_i Y_i)}{n \sum X_1^2 - (\sum X_i)^2}$$
$$b = \frac{n \cdot \sum X_1 Y - (\sum X_1)(\sum Y)}{n \sum X_i - (\sum X_i)^2}$$

#### 2.1.6. Backpropagation

Algoritma Backpropagation (BP) merupakan salah satu algoritma Jaringan Saraf Tiruan (JST) yang terkenal dan ekstensif yang sering digunakan untuk prediksi dan perkiraan waktu, yang juga menentukan hasil untuk fungsi non-linear (Fardhani, A. A., Natalia Simanjuntak, D. I., & Wanto, 2018). Algoritma backpropagation digunakan untuk latihan. Pada algoritma ini sistem pemrosesan informasi mempunyai karakteristik menyerupai jaringan saraf manusia. Representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut sehingga tercipta sebagai suatu generalisasi model matematika dari pemahaman manusia (human cognition) yang didasarkan atas asumsi pemrosesan informasi terjadi pada elemen sederhana yang disebut neuron. Dalam backpropagation, fungsi aktivasi menggunakan beberapa kondisi yaitu, terus-menerus, kemudahan perbedaan dan fungsi tidak menurun. Metode pelatihan backpropagation

melibatkan feedforward dari pola pelatihan input, perhitungan dan backpropagation dari kesalahan dan penyesuaian bobot dalam sinapsis. Ciri khas backpropagation melibatkan tiga lapisan: lapisan input, dimana data diperkenalkan ke jaringan; hidden layer, dimana data diproses; dan lapisan output, di mana hasil dari masukan yang diberikan oleh lapisan input.

Pelatihan Backpropagation meliputi 3 fase:

1. fase propagasi maju (feedforward) pola pelatihan masukan. Pola masukan dihitung maju mulai dari layer masukan hingga layer keluaran dengan fungsi aktivasi yang ditentukan;
2. fase propagasi mundur (backpropagation) dari error yang terkait. Selisih antara keluaran dan target merupakan kesalahan yang terjadi. Kesalahan tersebut dipropagasi mundur, dimulai dari garis yang berhubungan langsung dengan unit-unit di layer keluaran;
3. fase modifikasi bobot.

Ketiga tahapan tersebut diulangi terus-menerus sampai mendapatkan nilai error yang diinginkan. Setelah training selesai dilakukan, hanya tahap pertama yang diperlukan untuk memanfaatkan jaringan syaraf tiruan tersebut. Kemudian, dilakukan pengujian terhadap jaringan yang telah dilatih. Pembelajaran algoritma jaringan syaraf membutuhkan perambatan maju dan diikuti dengan perambatan mundur.

#### **2.1.6.1. Fase Propagasi Maju**

Selama propagasi maju, sinyal masukan ( $x_1$ ) dipropagasikan ke layer tersembunyi menggunakan fungsi aktivasi yang ditentukan. Keluaran dari unit tersembunyi ( $Z_1$ ) tersebut selanjutnya dipropagasi maju lagi ke layer

tersembunyi berikutnya dengan fungsi aktivasi yang telah ditentukan. Dan seterusnya hingga menghasilkan keluaran jaringan ( $y_k$ ).

Berikutnya, keluaran jaringan ( $y_k$ ) dibandingkan dengan target yang harus dicapai ( $t_k$ ). Selisih ( $t_k - y_k$ ) adalah kesalahan yang terjadi. Jika kesalahan ini lebih kecil dari batas toleransi yang ditentukan, maka iterasi dihentikan. Jika kesalahan masih lebih besar dari batas toleransi, maka bobot setiap garis dari jaringan akan dimodifikasi untuk mengurangi kesalahan.

#### **2.1.6.2. Fase Propagasi Mundur**

Berdasarkan kesalahan  $t_k - y_k$  dihitung faktor  $\delta_k$  ( $k = 1, \dots, m$ ) yang dipakai untuk mendistribusikan kesalahan di unit  $Y_k$  ke semua unit tersembunyi yang terhubung langsung dengan  $Y_k$ .  $\delta_k$  juga dipakai untuk mengubah bobot garis yang berhubungan langsung dengan unit keluaran.

Dengan cara yang sama, dihitung faktor  $\delta_j$  di setiap layer tersembunyi sebagai dasar perubahan bobot semua garis yang berasal dari unit tersembunyi di layer di bawahnya. Dan seterusnya hingga semua faktor  $\delta$  di unit tersembunyi yang terhubung langsung dengan unit masukan dihitung.

#### **2.1.6.3. Fase Propagasi Bobot**

Setelah semua faktor  $\delta$  dihitung, bobot semua garis dimodifikasi bersamaan. Perubahan bobot suatu garis didasarkan atas faktor  $\delta$  neuron di layer atasnya. Sebagai contoh, perubahan bobot garis yang menuju ke layer keluaran didasarkan atas yang ada di unit keluaran.

Ketiga fase tersebut diulang-ulang hingga kondisi penghentian dipenuhi. Umumnya kondisi penghentian yang sering dipakai adalah jumlah iterasi atau kesalahan. Iterasi akan dihentikan jika jumlah iterasi yang

dilakukan sudah melebihi jumlah maksimum iterasi yang ditetapkan, atau jika kesalahan yang terjadi sudah lebih kecil dari batas toleransi yang ditetapkan.

#### 2.1.6.4. Prosedur Pelatihan

Seperti halnya jaringan syaraf yang lain, pada jaringan feedforward (umpan maju) pelatihan dilakukan dalam rangka perhitungan bobot sehingga pada akhir pelatihan akan diperoleh bobot-bobot yang baik. Selama proses pelatihan, bobot-bobot diatur secara iteratif untuk meminimumkan error (kesalahan) yang terjadi. Error (kesalahan) dihitung berdasarkan rata-rata kuadrat kesalahan (MSE). Rata-rata kuadrat kesalahan juga dijadikan dasar perhitungan unjuk kerja fungsi aktivasi. Sebagian besar pelatihan untuk jaringan feedforward (umpan maju) menggunakan gradien dari fungsi aktivasi untuk menentukan bagaimana mengatur bobot-bobot dalam rangka meminimumkan kinerja. Gradien ini ditentukan dengan menggunakan suatu teknik yang disebut backpropagation.

Pada dasarnya, algoritma pelatihan standar backpropagation akan menggerakkan bobot dengan arah gradien negatif. Prinsip dasar dari algoritma backpropagation adalah memperbaiki bobot-bobot jaringan dengan arah yang membuat fungsi aktivasi menjadi turun dengan cepat.

Langkah-langkah yang dilakukan pada prosedur pelatihan adalah:

- **Langkah 0:** Inisialisasi bobot keterhubungan antara neuron dengan menggunakan bilangan acak kecil (-0.5 sampai +0.5).
- **Langkah 1 :** Kerjakan langkah 2 sampai langkah 9 selama kondisi berhenti yang ditentukan tidak dipenuhi.

- **Langkah 2** : Kerjakan langkah 3 sampai langkah 8 untuk setiap pasangan pelatihan.

### Propagasi maju

- **Langkah 3** : Setiap unit masukan ( $x_i$ ,  $i = 1, \dots, n$ ) menerima sinyal masukan  $x_i$ , dan menyebarkannya ke seluruh unit pada lapisan tersembunyi
- **Langkah 4** : Setiap unit tersembunyi ( $x_i$ ,  $i = 1, \dots, p$ ) jumlahkan bobot sinyal masukannya :

$$z_{in_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij}$$

$v_{0j}$  = bias pada unit tersembunyi  $j$  aplikasikan fungsi aktivasinya untuk menghilangkan sinyal keluarannya,  $z_j = f(z_{in_j})$ , dan kirimkan sinyal ini keseluruh unit pada lapisan diatasnya (unit keluaran).

- **Langkah 5** : tiap unit keluaran ( $y_k$ ,  $k = 1, \dots, m$ ) jumlahkan bobot sinyal masukannya :

$$y_{in_k} = w_{0k} + \sum_{j=1}^p z_j w_{jk}$$

$w_{0k}$  = bias pada unit keluaran  $k$  dan aplikasikan fungsi aktivasinya untuk menghitung sinyal keluarannya,  $y_k = f(y_{in_k})$ .

### Propagasi balik

- **Langkah 6** : tiap unit keluaran ( $y_k$ ,  $k = 1, \dots, m$ ) menerima pola target yang saling berhubungan pada masukan pola pelatihan, hitung kesalahan informasinya,

$$\delta_k = (t_k - y_k) f'(y_{in_k})$$

hitung koreksi bobotnya (digunakan untuk mempengaruhi wjk nantinya),

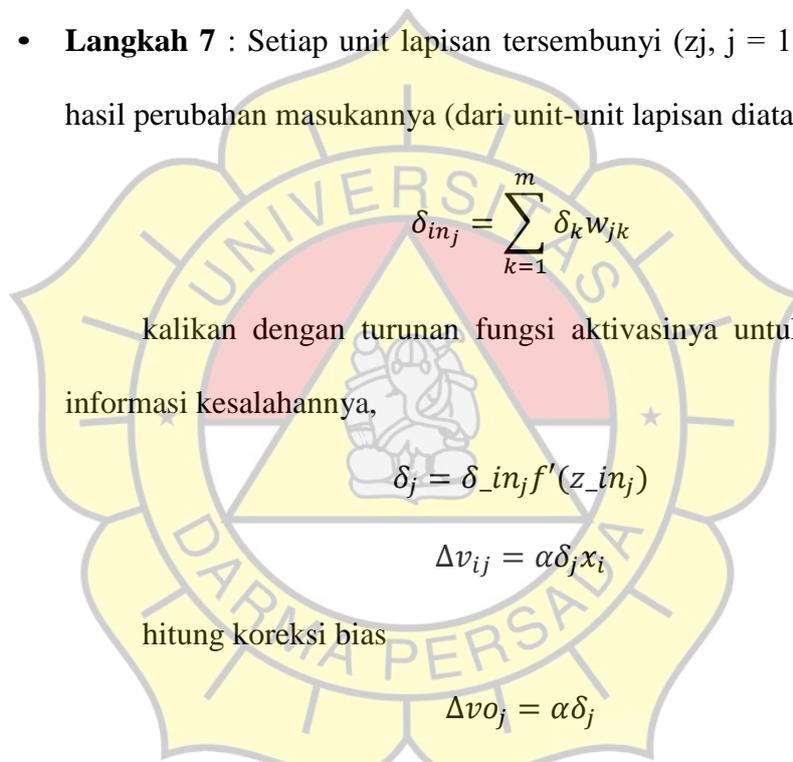
$$\Delta w_{jk} = \alpha \delta_k z_j$$

hitung koreksi biasnya (digunakan untuk mempengaruhi wo\_k nantinya)

$$\Delta wo_k = \alpha \delta_k$$

dan kirimkan  $\delta_k$  ke unit-unit pada lapisan dibawahnya,

- **Langkah 7** : Setiap unit lapisan tersembunyi ( $z_j$ ,  $j = 1, \dots, p$ ) jumlah hasil perubahan masukannya (dari unit-unit lapisan diatasnya),



$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk}$$

kalikan dengan turunan fungsi aktivasinya untuk menghitung informasi kesalahannya,

$$\delta_j = \delta_{in_j} f'(z_{in_j})$$

$$\Delta v_{ij} = \alpha \delta_j x_i$$

hitung koreksi bias

$$\Delta vo_j = \alpha \delta_j$$

- **Langkah 8** : Update bobot dan bias pada hubungan antar lapisan

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk}$$

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij}$$

- **Langkah 9** : Tes kondisi terhenti

Backpropagation secara garis besar terdiri dari dua fase, fase maju dan fase mundur. Selama fase maju algoritma ini memetakan nilai masukan untuk mendapatkan keluaran yang diharapkan. untuk

menghasilkan keluaran pola maka didapatkan dari rekapitulasi bobot masukan dan dipetakan untuk fungsi aktivasi jaringan.

### 2.1.7. *Unified Modeling Language (UML)*

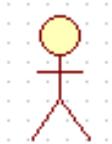
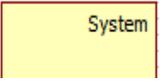
*Unified Modeling Language (UML)* merupakan sistem arsitektur yang bekerja dalam *OOAD (Object-Oriented Analysis/Design)* dengan satu bahasa yang konsisten untuk menentukan, visualisasi, mengkontruksi, dan mendokumentasikan *artifact* (sepotong informasi yang digunakan atau dihasilkan dalam suatu proses (Nugroho, 2015)). Tujuan *UML* diantaranya adalah :

1. Memberikan model yang siap pakai, pemodelan *visual* yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
2. Memberikan contoh bahasa pemodelan yang bebas dari berbagai Bahasa pemrograman dan proses rekayasa.
3. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan *UML* terdiri dari beberapa diagram pemodelan berikut ini.

#### 1. *Use Case Diagram*

*Use Case Diagram* merupakan model diagram *UML (Unified Modeling Language)* yang digunakan untuk menggambarkan kesepakatan fungsional yang diharapkan dari sebuah system ((Nugroho, 2015)). Diagram ini menjelaskan manfaat suatu sistem jika dilihat menurut pandangan orang yang berada diluar sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar. Berikut adalah beberapa simbol yang digunakan di use case:

**Tabel 2. 1 Komponen Use Case Diagram (Nugroho, 2015)**

Simbol	Nama Elemen	Keterangan
	Aktor	Aktor adalah seseorang atau sesuatu yang berperan untuk berinteraksi dengan sistem tetapi tidak memiliki kontrol akan <i>use case</i> .
	UseCase	Simbol UseCase menggambarkan fungsionalitas dari sistem, sehingga pengguna paham mengenai manfaat dari sistem yang dibangun
	Asosiasi	Penghubung antar elemen ( aktor / UseCase ) di dalam system
	Generalisasi	Sebuah elemen yang dihasilkan dari pewarisan elemen lain
	Batasan system	Area yang digunakan untuk menempatkan usecase sebagai batasan apa yang dilakukan system

## 2. Activity Diagram

*Activity Diagram* memodelkan alur proses dan hubungan antar proses dalam suatu sistem informasi dan juga berisikan tentang scenario yang ada dalam sistem tersebut (Nugroho, 2015). Berikut komponen – komponen yang ada didalam sebuah model *Activity Diagram*:

**Tabel 2. 2 Komponen Activity Diagram (Nugroho, 2015)**

Simbol	Nama Elemen	Keterangan
	Aktifitas	Notasi yang menggambarkan aktor dari proses dalam aliran pekerjaan
	Transisi	Notiasi yang digunakan memperlihatkan aliran kontrol dari satu aktifitas ke aktifitas lain
	<i>Decision</i>	Notasi yang menandakan adanya kontrol cabang berdasarkan <i>decision point</i>
	Awalan & Akhiran	Notasi yang menandakan awal dan akhir dari sebuah aktifitas system

## 3. Sequence Diagram

*Sequence Diagram* adalah diagram yang menjelaskan interaksi antar objek yang disusun berdasarkan waktu proses berlangsung.

Diagram ini digunakan untuk menggambarkan tahap demi tahap yang harus dilakukan oleh pengguna sistem untuk menghasilkan sesuatu dari *use case diagram* yang sudah dibuat (Nugroho, 2015). Berikut adalah beberapa komponen yang sangat berpengaruh dalam pembentukan sebuah *Sequence Diagram*:

**Tabel 2. 3 Komponen Sequence Diagram (Nugroho, 2015)**

Simbol	Nama Elemen	Keterangan
	Objek	Komponen yang menjadikan sebuah objek dalam membuat diagram
	<i>Stimulus</i>	Untuk menandakan hubungan komunikasi antar objek
	<i>Self Stimulus</i>	Fungsi sama dengan <i>Stimulus</i> , tetapi pesan yang disampaikan dikirimkan untuk objek itu sendiri
	<i>Focus Control</i>	Sebagai tempat untuk hasil input atau output dari sebuah proses yang dilakukan oleh objek ataupun aktor yang ada dalam sistem

### 2.1.8. Python

Menurut Ibrahim, (2015), Python adalah bahasa pemrograman interpretatif, berorientasi objek dan semantik yang dinamis. Python memiliki high-level struktur data, *dynamic typing* dan *dynamic binding*. Python memiliki

sintaks sederhana dan mudah dipelajari untuk penekanan pada ke-mudahan membaca dan mengurangi biaya perbaikan program. Python mendukung modul dan paket untuk mendorong kemandirian program dan *code reuse*. Interpreter Python dan standard library-nya tersedia secara gratis untuk semua platform dan dapat secara bebas disebar. Bahasa pemrograman ini dibuat oleh Guido van Rossum.

### 2.1.9. MySQL

*MySQL* adalah *Relational Database Management System (RDBMS)* yang didistribusikan secara gratis dibawah lisensi *GPL (General Public License)*. Dimana setiap orang bebas untuk menggunakan *MySQL*, namun tidak boleh dijadikan produk turunan yang bersifat *closed source* atau komersial. *MySQL* sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu *SQL (Structured Query Language)*. *SQL* adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis (Kadir, 2014). Keandalan suatu sistem *database (DBMS)* dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah *SQL*, yang dibuat oleh user maupun program-program aplikasinya. *MySQL* biasanya digunakan atau diinstall bersamaan dengan *XAMPP* sehingga untuk melihat isi tabel bisa menggunakan *PHPmyAdmin*.

### 2.1.10. Evaluasi Performance

Untuk menguji hasil klasifikasi pada sistem yang telah dibangun, maka dibutuhkan suatu metode perhitungan evaluasi performansi yaitu root mean

square error (RMSE). RMSE digunakan untuk membandingkan metode-metode estimasi yang digunakan, yaitu untuk menentukan metode estimasi yang paling akurat. Menurut Kolen & Brennan yang dikutip (Rahman, A. Y., & Istiadi, 2020) RMSE dapat ditentukan dengan persamaan berikut:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n |\hat{Y}_i - Y_i|^2}{n}}$$

