

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1. Sistem Tanya Jawab**

Sistem Tanya Jawab (Question Answering System) merupakan sebuah sistem pada komputer yang dapat menjawab pertanyaan menggunakan bahasa alami yang biasa dipakai manusia secara otomatis. Jawaban yang diperoleh dari sistem berisi informasi yang berasal dari sebuah sumber basis data [6]. Sebuah sistem tanya jawab, diharapkan dapat membantu orang untuk mendapatkan informasi lebih mengenai hal tertentu tanpa harus bertanya kepada orang lain. Tujuan utama dari sistem jawab otomatis adalah untuk mencari jawaban yang tepat menggunakan bahasa alami yang ditentukan oleh manusia [7]. Saat ini sistem tanya jawab telah dikembangkan dalam berbagai bidang seperti pada asisten pribadi, pelayanan konsumen yang diterapkan pada pelayanan jasa guna untuk mendapatkan informasi dan memenuhi kebutuhan konsumen.

#### **2.2. Preprocessing**

*Preprocessing* adalah proses pengolahan data raw sebelum diolah pada tahap selanjutnya. Pada penelitian ini data yang digunakan adalah data pertanyaan berupa text dari hasil wawancara dari kuesioner yang tidak terstruktur. Oleh karena itu, proses ini sangat dibutuhkan karena akan digunakan untuk mengubah data yang tidak terstruktur menjadi terstruktur (normalisasi) agar dapat diproses pada tahap selanjutnya. Pada penelitian ini tahapan dalam *preprocessing* terdiri dari *case folding*, *cleansing* (penghilangan simbol), *tokenizing*, *stopword removal*, dan *stemming*.

### **2.2.1. Case Folding**

Case Folding adalah proses menyeragamkan semua teks ke dalam case sama menjadi huruf kecil (lowercase) atau huruf besar (uppercase) . Pada proses ini, tahap casefolding menyeragamkan huruf dalam teks menjadi huruf kecil (lowercase) [8].

### **2.2.2. Cleansing**

Cleansing merupakan proses membersihkan kata-kata yang tidak diperlukan untuk mengurangi noise. Kata yang dihilangkan adalah url, hashtag (#), username (@username), dan email. Selain itu juga tanda baca seperti titik(.), koma(,), dan tanda baca lainnya akan dihilangkan [8]. Pada penelitian ini, proses cleansing untuk menghilangkan simbol seperti tanda baca dan emoticon dalam teks.

### **2.2.3. Tokenizing**

Tokenizing adalah proses untuk memisahkan kata dalam teks menjadi beberapa urutan kata yang dipisahkan melalui spasi atau karakter lainnya [8]. Pada penelitian, spasi digunakan untuk memisahkan kata dalam teks.

### **2.2.4. Stemming**

Stemming adalah proses menjadikan kata yang memiliki imbuhan ke bentuk kata dasarnya. Contohnya adalah kata “membayar” diubah menjadi “bayar” [8].

### **2.2.5. Stopword Removal**

Remove Stopword merupakan proses penghilangan kata tidak penting yang terdapat dalam stoplist. Kata-kata yang dihilangkan tersebut menunjukkan kurang relevansinya dengan teks dan didefinisikan dalam stopword list [7].

### 2.2.6. TF-IDF

Metode TF-IDF merupakan metode untuk menghitung bobot setiap kata yang paling umum digunakan pada information retrieval. Metode ini juga terkenal efisien, mudah dan memiliki hasil yang akurat [9]. Metode ini akan menghitung bobot setiap term dalam sebuah dokumen dengan rumus:

$$IDF_{t_i} = \log\left(\frac{D}{df_i}\right) \quad 2.1$$

Dimana:

D : total keseluruhan dokumen,

df<sub>i</sub> : banyaknya dokumen yang mengandung kata ke -i yang dicari.

Kemudian algoritma yang digunakan untuk menghitung bobot (w) term dari masing-masing dokumen terhadap term menggunakan persamaan [9].

$$w_i = tf_i * IDF_{t_i} \quad 2.2$$

Dimana :

tf<sub>i</sub> : term frekuensi/ frekuensi kata

W<sub>i</sub> : bobot dokumen ke-d terhadap term ke-t

### 2.3. Relevance Vector Machine (RVM)

Relevance Vector Machine (RVM) diperkenalkan oleh Michael E. Tipping pada tahun 2000. RVM merupakan metode pembelajaran mesin yang diadaptasi dari Bayesian Framework. Selain itu RVM memiliki kemiripan dengan Support Vector Machine (SVM) dalam hal model fungsinya. Seperti SVM, RVM dikembangkan untuk analisis binary [10].

RVM termasuk ke dalam supervised learning. Sama halnya dengan metode supervised learning yang lainnya, RVM juga membutuhkan data latih yang terdiri dari himpunan vektor  $\{x_n\}_{n=1}^N$  yang dipetakan terhadap target  $\{t_n\}_{n=1}^N$ ,

dimana target berupa sebuah nilai untuk regresi dan label kelas untuk klasifikasi. Tujuan dari supervised learning adalah melatih suatu model menggunakan sejumlah data latih, sehingga diharapkan suatu masukan  $x$  dapat diprediksi nilai atau kelasnya seakurat mungkin [11]. Prediksi dengan menggunakan metode RVM dapat dihitung dengan menggunakan persamaan:

$$y(x; w) = \sum_{i=1}^M w_i \varphi_i(x) + w_0 = w^T \varphi(x) + w_0 \quad 2.3$$

di mana  $w$  adalah vektor bobot,  $\varphi_i(x)$  adalah fungsi kernel terhadap data  $x$ , dan  $w_0$  merupakan *bias*.

Fungsi kernel yang digunakan adalah kernel RBF (Radial Basis Function). Fungsi ini dihitung dengan menggunakan persamaan:

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right) \quad 2.4$$

dimana  $x$  dan  $x'$  merupakan data masukan.

Klasifikasi dalam kasus ini adalah model prediksi mengambil bentuk kombinasi linear dari fungsi basis/kernel yang diubah oleh fungsi logistic sigmoid [11].

$$y(x, w) = \sigma(x; w) = \sigma(w^T \varphi(x)) \quad 2.5$$

dimana  $\sigma(\cdot)$  adalah fungsi logistic sigmoid yang didefinisikan dengan persamaan.

$$\sigma(-y) = \frac{1}{1 + \exp(-y)} \quad 2.6$$

Berdasarkan definisi dari distribusi Bernoulli, likelihood terdefiniskan sebagai berikut:

$$p(t|w) = \prod_{i=1}^N \sigma\{y(x_n; w)\}^{t_n} \{y(x_n; w)\}^{1-t_n} \quad 2.7$$

untuk target  $t_n \in \{0, 1\}$ .

Persamaan likelihood dilengkapi dengan sebuah prior terhadap parameter (bobot) dalam bentuk

$$p(w|a) = \prod_{i=1}^N \frac{\sqrt{a_i}}{2\pi} \exp\left(-\frac{a_i w_i^2}{2}\right) \quad 2.8$$

dimana  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)^T$  merupakan hyperparameter yang diperkenalkan untuk mengontrol kekuatan dan prior terhadap parameter bobot yang diasosiasikannya,  $N$  merupakan banyak kalimat, dan  $w$  merupakan parameter bobot.

Untuk nilai  $\alpha$  tertentu, distribusi posterior bobot terhadap data dapat dihitung menggunakan aturan Bayes, dengan persamaan:

$$p(w|t, a) = \frac{p(t|w)p(a)}{p(t|a)} \quad 2.9$$

dimana  $p(t|w)$  adalah likelihood,  $p(w|\alpha)$  adalah prior, dan  $p(t|\alpha)$  adalah evidence.

Parameter bobot model tidak dapat diperoleh dengan cara analitik, sehingga aproksimasi Laplacian digunakan. Sejak  $p(w|t, \alpha)$  secara linear proporsional  $p(t|w) \times p(w|\alpha)$ , dapat dimungkinkan untuk mencari maksimum dari persamaan

$$\begin{aligned} \ln p(w|t, \alpha) &= \ln \{ p(t|w)p(w|\alpha) \} + \ln p(t|\alpha) \\ &= \sum_n^N \{ t_n \ln y_n + (1-t_n) \ln (1-y_n) \} + \frac{1}{2} w^T A w \end{aligned} \quad 2.10$$

untuk parameter bobot paling mungkin  $w'$  dengan  $y_n = \sigma \{ y(x_n; w) \}^{t_n}$  dan  $A = \text{diag}(\alpha_1, \alpha_2, \dots, \alpha_n)$  yang terdiri dari nilai  $\alpha$  yang diinisialisasi. Parameter  $\sigma$  adalah parameter bebas, penelitian sebelumnya memberi nilai 0,5. Untuk mencari parameter  $w'$  pada proses reestimasi, metode IRLS dapat digunakan.

Fungsi logistic log-likelihood dapat diturunkan sebanyak dua kali untuk memperoleh matriks Hessian dengan persamaan

$$\begin{aligned} H &= \nabla \nabla \ln p(w|t, \alpha) \\ &= \varphi_n^T B \varphi_n + A \end{aligned} \quad 2.11$$

dimana  $\phi$  merupakan matriks kernel, dan  $B = \text{diag}(\beta_1, \beta_2, \dots, \beta_n)$  merupakan matriks diagonal yang elemen matriksnya diperoleh dengan persamaan

$$\beta_n = \sigma\{y(x_n; w')\} [1 - \sigma\{x_n; w'\}] \quad 2.12$$

$$\beta_n = y_n(1 - y_n) \quad 2.13$$

dimana  $w'$  merupakan parameter bobot yang akan direestimasi. Proses reestimasi parameter bobot  $w$  didapat dengan persamaan

$$w' = w_n + \lambda \Delta \quad 2.14$$

dimana  $\lambda$  merupakan koefisien estimasi, dan  $\Delta$  merupakan vektor yang diperoleh dari persamaan

$$\Delta = H^{-1} \cdot g \quad 2.15$$

dimana  $H^{-1}$  adalah invers dari matriks hesian dan  $g$  adalah gradien negatif yang diperoleh dari penurunan fungsi logistic log-likelihood sebanyak satu kali dengan persamaan

$$\begin{aligned} g &= -\nabla \ln p(w|t, \alpha) \\ &= \varphi_n^T e - Aw_n \end{aligned} \quad 2.16$$

Vektor  $e$  didapat dengan menghitung selisih antara vektor  $t$  dan vektor  $y$ . Berikut persamaan yang digunakan.

$$e = t - y \quad 2.17$$

Setelah estimasi parameter bobot  $w$  dilakukan, selanjutnya *hyperparameter*  $\alpha_j$  dilakukan reestimasi dengan menggunakan persamaan

$$\alpha_j^{new} = \frac{\gamma_j}{w_j^2} \quad 2.18$$

dimana  $\gamma_j$  didefinisikan dengan persamaan

$$\gamma_j = 1 - a_j \Sigma_{jj} \quad 2.19$$

dimana  $\Sigma_{jj}$  merupakan elemen diagonal matriks kovarians yang didapat dari persamaan

$$\begin{aligned}\Sigma &= -(\nabla \nabla \ln p(w|t, a))^{-1} \\ &= H^{-1}\end{aligned}\quad 2.20$$

dimana  $H^{-1}$  merupakan invers dari matriks Hessian. Selain dengan cara menggunakan persamaan, reestimasi hyperparameter  $\alpha_j$  juga dapat dilakukan dengan cara *hybrid update* dengan menggunakan persamaan

$$\alpha_j = \frac{\gamma_j}{w_j^2}; \text{ jika } i < \frac{i_{max}}{2} \quad 2.21$$

$$\alpha_j = \gamma_j \left( \frac{w_j^2}{\gamma_j} - \Sigma_{jj} \right); \text{ jika } i \geq \frac{i_{max}}{2} \quad 2.22$$

dimana  $i_{max}$  merupakan jumlah iterasi maksimum pada pelatihan.

Proses pelatihan akan berhenti jika nilai  $\delta$  kurang dari nilai pembandingnya dan setiap parameter dianggap telah optimum. Nilai pembanding minimum yang digunakan adalah  $10^{-6}$ . Berikut persamaan untuk menghitung  $\delta$ .

$$\delta_j = |\ln(\alpha_j) - \ln(\alpha_j^{lama})| \quad 2.23$$

### 2.3.1. Multiclass RVM

Pada dasarnya RVM dikembangkan untuk persoalan klasifikasi biner (dua kelas). Namun dalam banyak kasus dibutuhkan lebih dari dua kelas atau kategori. Jumlah kelas yang lebih dari dua tidak dapat menggunakan RVM biasa, perlu cara lain untuk menggunakan RVM sebagai pengklasifikasi multiclass. Diantara caranya adalah menggunakan metode multiclass yang biasa digunakan pada Support Vector Machine (SVM), yaitu Metode Satu Lawan Satu (One Against One) dan Metode Satu Lawan Semua (One Against All) [12].

### 2.3.1.1. Metode Satu Lawan Semua (One Against All)

Pada metode satu lawan semua, klasifikasi k-kelas dilakukan dengan menentukan k fungsi pemisah, dimana k adalah banyaknya kelas. Fungsi pemisah ( $p$ ) akan dilatih menggunakan semua data dari kelas-i dengan label 1 dan semua data dari kelas lain dengan label 0. Contohnya pada kasus tiga kelas, ketika melatih  $p^1$ , semua data kelas 1 akan diberi label 1, sedangkan semua data dari kelas lain (kelas 2 dan kelas 3) akan diberi label 0. Kemudian dilanjutkan melatih  $p^2$  dengan menggunakan semua data kelas 2 dan diberi label 1, sedangkan semua data kelas lain diberi label 0. Begitu pula dengan yang lainnya hingga kelas k [12].

### 2.3.1.2. Metode Satu Lawan Satu (One Against One)

Dengan metode satu lawan satu, terlebih dahulu perlu menentukan  $k(k-1)/2$  fungsi pemisah. Setiap fungsi akan dilatih menggunakan data dari dua kelas. Misal, dalam kasus tiga kelas terdapat 3 fungsi pemisah, yaitu  $p^{12}$ ,  $p^{13}$ ,  $p^{23}$ . Ketika melatih  $p^{12}$ , semua data dari kelas 1 diberi label 1 dan semua data dari kelas 2 diberi label 0. Pendekatan yang sama dipakai untuk melatih  $p^{13}$  dan  $p^{23}$  [11].

Setelah fungsi pemisah didapat, maka selanjutnya adalah melakukan pengujian. Salah satu metode yang dapat dilakukan adalah menggunakan *max-voting*. Metode ini menghitung jumlah *voting* untuk setiap nilai prediksi, jumlah *voting* paling banyak akan dianggap sebagai kelas prediksi [11]. Misalnya pada fungsi pemisah  $p^{12}$ , nilai prediksi termasuk kelas 1, maka *voting* untuk kelas 1 ditambah satu. Begitu juga sebaliknya, jika nilai prediksi termasuk kelas 2, maka *voting* untuk kelas 2 ditambah satu.

Berdasarkan penelitian yang dilakukan oleh Hsu dan Lin mengenai perbandingan metode multiclass pada SVM, metode satu lawan satu memiliki akurasi yang lebih baik dibanding metode satu lawan semua [12]. Penelitian tersebut dilakukan dengan cara melakukan uji coba menggunakan kernel linear



dan RBF pada beberapa kasus yang memiliki beragam jumlah data dan kelas. Hasil penelitian tersebut menunjukkan bahwa pada sebagian besar kasus metode satu lawan satu lebih unggul dibandingkan metode satu lawan semua. Oleh karena itu, pada penelitian ini akan digunakan metode satu lawan satu untuk mengatasi masalah multiclass.

#### **2.4. Akurasi**

Untuk mengetahui performansi RVM dapat dilakukan dengan cara menghitung akurasi. Akurasi adalah jumlah prediksi benar dibagi jumlah data. Berikut rumus untuk menghitung akurasi [11].

$$Akurasi = \frac{Jumlah\ Prediksi\ Benar}{Jumlah\ Data} * 100\% \quad 2.24$$

#### **2.5. Cosine Similarity**

Cosine similarity adalah sebuah metode untuk mengukur kemiripan kata. Semakin besar nilai dari perhitungannya, maka semakin mirip dengan kata yang dibandingkan.

#### **2.6. Pemodelan Sistem**

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

Unified Modeling Language (UML) merupakan sebuah bahasa yang digunakan untuk memodelkan suatu sistem perangkat lunak berorientasi objek. UML sudah banyak digunakan oleh industri-industri perangkat lunak untuk merancang, menggambarkan, dan mendokumentasikan sistem perangkat lunak. UML sangat cocok digunakan untuk mendesain perangkat lunak yang dibuat menggunakan bahasa pemrograman berorientasi objek, seperti C++, Java, PHP, dan Python. Walaupun UML pada umumnya digunakan pada bahasa pemrograman berorientasi objek, namun bisa juga digunakan pada bahasa pemrograman prosedural [13].

Berikut ini merupakan tujuan merancang dengan menggunakan UML.

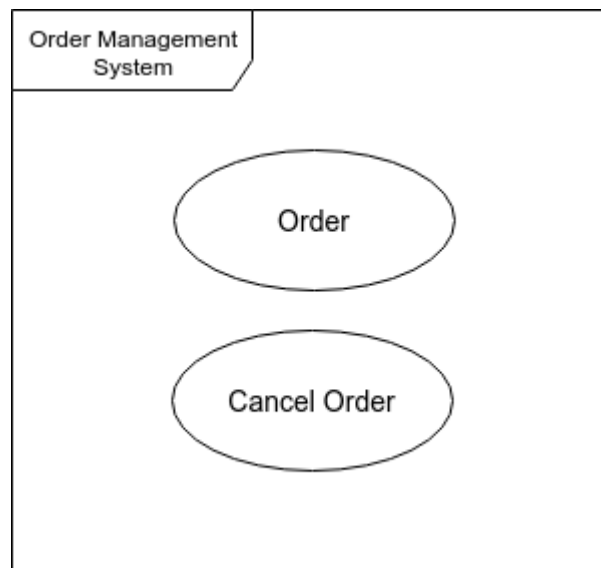
1. Menggambarkan batasan dan fungsi-fungsi sistem secara umum, dibuat dengan Use Case Diagram.
2. Menggambarkan hubungan antar objek serta strukturnya di dalam sistem, dibuat dengan Class Diagram.
3. Menggambarkan kegiatan atau proses bisnis yang dilakukan sistem secara umum, dibuat dengan Sequence Diagram.
4. Menggambarkan fungsi-fungsi ataupun kegiatan yang dilakukan sistem secara lebih terperinci, dibuat dengan Activity Diagram.

#### **2.6.1.1. Use Case Diagram**

Use Case Diagram merupakan sebuah model fungsional yang menggambarkan hubungan antara pengguna, sistem, dan sistem eksternal. Use Case Diagram menggambarkan fungsi-fungsi sistem dari sudut pandang pengguna. Use Case Diagram terdiri dari dua simbol utama, yaitu Use Case dan Actor. Use Case adalah fungsi-fungsi yang terdapat pada sistem. Sedangkan Actor adalah pengguna yang memanfaatkan sistem, atau sistem eksternal yang digunakan oleh sistem [13]. Berikut ini adalah simbol dan notasi dasar yang terdapat pada Use Case Diagram.

##### **1. Sistem**

Sistem digambarkan sebagai kumpulan Use Case yang saling terhubung satu dengan yang lainnya dan dikelilingi oleh garis batasan (boundaries) berbentuk persegi empat.



**Gambar 2.1 Contoh Sistem pada Use Case Diagram**

## 2. Use Case

Use Case digambarkan berbentuk lonjong/oval, ditengahnya terdapat kata kerja yang menggambarkan fungsi Use Case tersebut.

## 3. Actor

Actor dapat berupa pengguna (user) atau sistem eksternal. Pada umumnya, Actor digambarkan berbentuk tiruan manusia (stick figure/ stickman). Actor ditempatkan diluar sistem atau garis batas (boundarie).

## 4. Relationship

Relationship digunakan sebagai notasi yang menggambarkan hubungan antara Actor dengan Use Case atau Use Case dengan Use Case lainnya. Relationship digambarkan dengan menggunakan garis yang terhubung dengan Actor atau Use Case.

### **2.6.1.2. Class Diagram**

Class adalah kumpulan objek yang memiliki kesamaan dalam hal struktur umum. Class Diagram adalah diagram yang menggambarkan class-class yang ada pada sistem dan hubungannya secara logika. Elemen-elemen yang terdapat pada Class Diagram, yaitu class-class, struktur class, sifat class (class behavior), perkumpulan/gabungan (association), pengumpulan/kesatuan (agregation), ketergantungan (dependency), relasi-relasi turunannya, keberagaman dan indikator navigasi, dan role name (peranan/tugas nama) [13].

### **2.6.1.3. Sequence Diagram**

Sequence Diagram merupakan diagram yang menggambarkan interaksi antar objek di dalam sistem yang disusun secara beraturan atau sesuai rangkaian waktu. Sequence Diagram menjelaskan langkah-langkah untuk menghasilkan suatu output. Langkah-langkah tersebut berisi pesan yang dikirimkan dari satu objek kepada objek lain [13].

### **2.6.1.4. Activity Diagram**

Activity Diagram adalah diagram yang menggambarkan aliran suatu aktivitas ke aktivitas lainnya yang terjadi di dalam sistem. Sebuah aktivitas menjelaskan operasi yang melibatkan class-class sehingga terjadi perubahan keadaan (state) pada sistem. Activity diagram biasanya digunakan untuk memodelkan alur kerja (workflow) atau proses bisnis [13].

## **2.7. Bahasa Pemrograman**

### **2.7.1. Python**

Python merupakan bahasa interpretatif. Python relatif lebih mudah dipelajari bila dibandingkan dengan bahasa yang lain. Hal ini karena kode-kode pada python sangat jelas dan mudah dipahami. Python mendukung berbagai platform (multi

platform), seperti Windows, macOS, dan GNU/Linux. Berikut ini kelebihan yang dimiliki Python [14].

1. Bahasa Python berorientasi objek.
2. Mudah dipelajari dan digunakan karena struktur bahasanya jelas dan sederhana.
3. Memiliki library yang banyak.
4. Bersifat modular.
5. Memiliki sistem pengelolaan memori otomatis.

### **2.7.2. Javascript**

Javascript adalah bahasa pemrograman berupa script yang berorientasi pada objek. Javascript bertugas mengatur interaksi setiap objek, sehingga web menjadi lebih interaktif dan dinamis. Objek-objek tersebut merupakan elemen-elemen HTML, seperti form, button, frame, dan item lainnya. Setiap objek memiliki properti, dan setiap properti memiliki atribut [15].

Javascript merupakan bahasa pemrograman yang dijalankan di sisi client, sehingga perintah tidak akan dikirim terlebih dahulu ke server, tapi langsung dieksekusi oleh client, dalam hal ini adalah browser. Script Javascript akan dieksekusi oleh browser bersamaan dengan kode HTML. Browser memiliki interpreter, sehingga dapat menjalankan perintah-perintah Javascript [16].

### **2.7.3. HTML (Hypertext Markup Language)**

HTML (Hypertext Markup Language) adalah suatu bahasa yang berisi sekumpulan tag markup. Tag-tag markup ini digunakan untuk memberitahukan kepada browser seperti apa halaman web yang harus ditampilkan [5]. Pada dasarnya HTML adalah dokumen ASCII atau teks biasa yang digunakan sebagai

standar pemformatan dokumen teks yang dikembangkan dari SGML (Standard Generalized Markup Language) [15].

File HTML dapat dibuat dengan menggunakan editor teks apasaja, seperti Notepad, Atom, Sublime, ataupun yang lainnya. Selain menggunakan editor teks biasa, file HTML juga bisa dibuat dengan menggunakan software seperti Dreamweaver yang dapat mempermudah pembuatan file HTML. File HTML yang dibuat harus disimpan dengan menggunakan ekstensi htm atau html, sehingga browser dapat mengetahui bahwa itu adalah file HTML [15].

#### **2.7.4. CSS (Cascading Style Sheet)**

CSS (Cascading Style Sheet) merupakan suatu bahasa yang digunakan untuk mengatur atau memperindah suatu halaman web. CSS dapat mengatur setiap bagian web, seperti tata letak, pewarnaan, ukuran, dan hal lainnya yang berhubungan dengan tampilan web. CSS biasanya digunakan untuk memudahkan memformat tampilan HTML [15].

Terdapat 2 bagian utama pada penulisan CSS, yaitu selector dan deklarasi. Selector adalah elemen HTML yang ingin diubah atau diperindah. Sedangkan deklarasi adalah bagian yang menjelaskan tentang apa yang akan dilakukan pada elemen HTML yang dipilih. Deklarasi terdiri dari dua bagian, yaitu properti dan nilai. Properti adalah atribut style yang ingin diubah, dan setiap properti memiliki nilai [15].

### **2.8. MySQL**

MySQL adalah sebuah database yang memiliki struktur relasional. Pada database ini terdapat tabel-tabel yang menyimpan data. Setiap tabel terdiri dari kolom dan baris. Setiap kolom mendefinisikan jenis data yang disimpan. Sedangkan baris adalah masukan yang berisi nilai-nilai untuk disimpan pada setiap kolom [16].

MySQL menggunakan SQL (Structured Query Language) sebagai bahasa yang digunakan untuk berkomunikasi. SQL merupakan bahasa standar untuk sistem manajemen database relasional. Perintah-perintah yang digunakan untuk berkomunikasi dengan database disebut dengan query [16].

## **2.9. Software Pendukung**

Berikut ini merupakan perangkat lunak pendukung yang digunakan untuk pembuatan program dalam penelitian ini.

### **2.9.1. Web Browser**

Web browser atau biasa disebut dengan browser adalah suatu perangkat lunak yang berfungsi sebagai penampil halaman web. Web browser berjalan di sisi client, sehingga setiap kali pengguna ingin mengakses suatu halaman web, web browser akan menghubungi server untuk meminta halaman web yang akan diakses. Kemudian server akan mengirimkan halaman web dalam bentuk HTML. Halaman web tersebut akan diinterpretasi oleh web browser dan ditampilkan di layar komputer [16]. Berikut ini adalah contoh web browser yang biasa digunakan.

1. Mozilla Firefox
2. Google Chrome
3. Internet Explorer
4. Safari
5. Opera

