CHAPTER 4
RESEARCH METHODOLOGY

This research has several stages. First, get a suitable dataset for this research. Second, I did the pre-processing of the data. Third, classify the clean data on the Support Vector Machine (SVM) algorithm. In this study, the workflow is as follows:

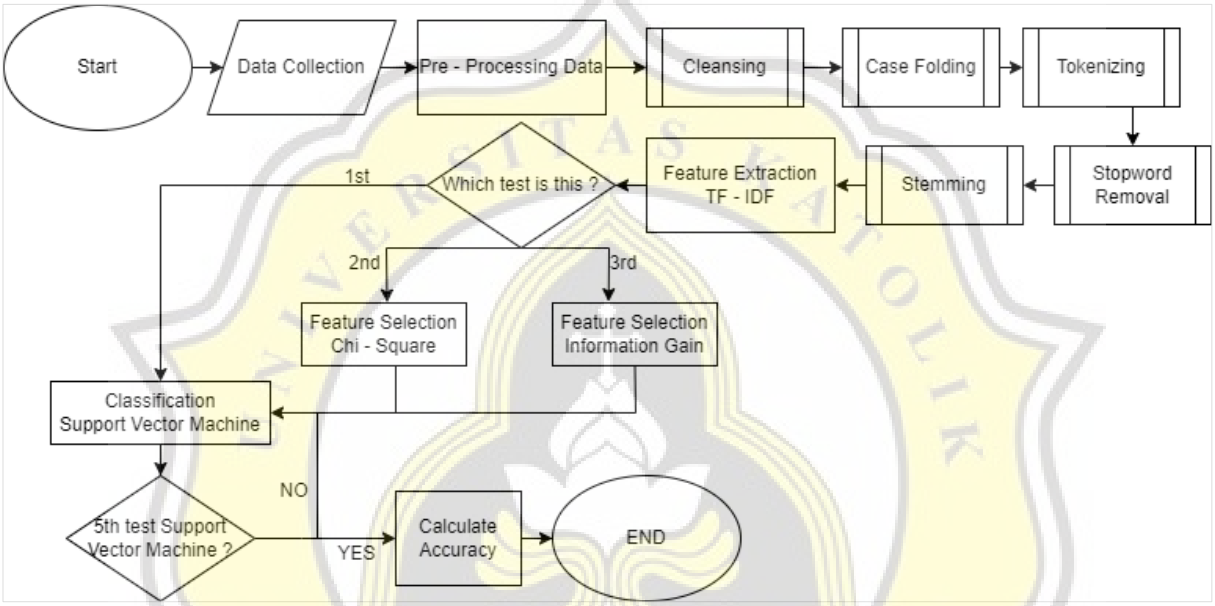

Figure 4.1 Workflow

4.1. Data Collection

From the workflow on Figure 4.1, I started with the data collection process. The dataset that I used was taken from the Kaggle website in 2016. The dataset can be downloaded at the link https://www.kaggle.com/andrewmvd/trip-advisor-hotel-reviews has a file size 14.6MB csv file with 20,492 review data along with the rating figures that have been given by hotel users. This dataset has 3 attributes, namely, id, review, and rating.

Table 4.1. Dataset

id	Review	Rating
1	Nice Beautifull Hotel Expensive parking got good deal stay hotel anniversary, arrived late evening took advice previous reviews did valet parking, check quick easy, little disappointed non-existent view room room clean nice size, bed comfortable woke stiff neck high pillows,….	4

2	ok nothing special charge diamond member hilton decided chain shot 20th anniversary seattle, start booked suite paid extra website description not, suite bedroom bathroom standard hotel room, took printed reservation…..	2
3	nice rooms not 4* experience hotel monaco seattle good hotel n't 4* level.positives large bathroom mediterranean suite comfortable bed pillowsattentive housekeeping staffnegatives ac unit malfunctioned stay desk disorganized, missed 3 separate wakeup calls, concierge busy hard touch...	3
4	unique, great stay, wonderful time hotel monaco, location excellent short stroll main downtown shopping area, pet friendly room showed no signs animal hair smells, monaco suite sleeping area big striped curtains pulled closed….	5
5	great stay great stay, went seahawk game awesome, downfall view building did n't complain, room huge staff helpful, booked hotels website seahawk package….	5

4.2. Data Preprocessing

Preprocessing data is the first step in doing sentiment analysis. In this data preprocessing stage, the raw data is cleaned in several steps before being entered into the feature extraction stage and others. Preprocessing this data is divided into several stages as follows:

4.2.1. Document Cleaning Process (Cleansing)

At this stage, the data that I have found is cleaned. This cleaning is done to remove characters such as html, hashtags, website addresses, usernames (@), and punctuation marks (.,'":;[]!?%&()<>) which aims to reduce noise on data. An example of data cleaning is as follows:

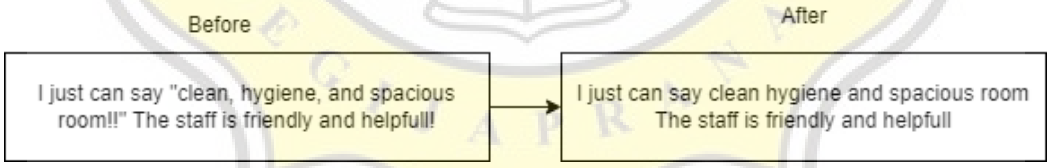

Figure 4.2 Cleansing

4.2.2. Case Folding

After the cleansing stage, the data is entered in the case folding stage where at this stage all the words contained in the review column will be converted into lowercase letters according to the letter and eliminate the characters in the review column because they can be considered as barriers. An example of the results of case folding can be seen below:

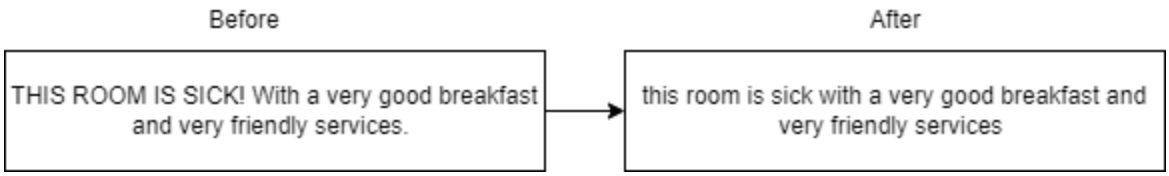

Figure 4.3 Case Folding

4.2.3. Tokenizing

At this stage, the data that has been carried out in the case folding stage will be continued with the tokenizing stage. In this tokenizing stage, each sentence in the review column will be broken down into words, then proceed with word collection analysis by separating the words and determining the syntactic structure of the data for each word.

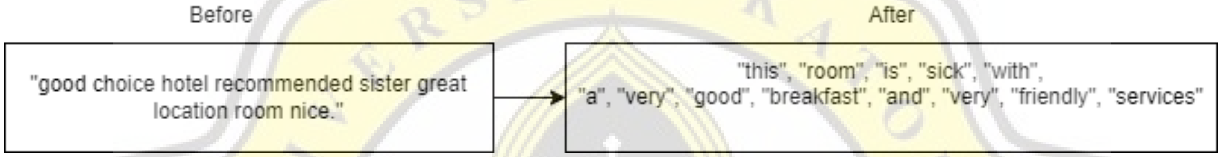

Figure 4.4 Tokenizing

4.2.4. Stopword Removal

After the tokenizing stage, the data is continued with the stopword removal process where at this stage the unimportant words are based on the stopword dictionary contained in the nltk.stopword library (English). These words are conjunctions such as (which, on, to, yes, no, etc.) or words that have no meaning will be deleted because they can affect the speed and performance of the classification later.

4.2.5. Stemming

At the stemming stage, the data that has been done with the stopword removal process will be continued at the stemming stage where at this stage the words in the review column are converted into basic words by removing affixes such as affixes, namely prefixes, insertions, suffixes, and combinations of prefixes and suffixes on derived words in the review sentence.

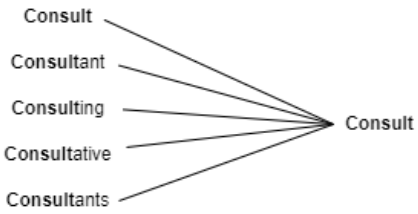

Figure 4.5 Stemming

4.3. Feature Extraction

The next process after preprocessing the data is term-weighting. Term-weighting is the process of assigning term weights to the data in my case study, namely the data in the reviews column. The method I use to perform feature extraction in this research is TF-IDF (Term Frequency-Inverse Document Frequency).

4.3.1. TF (Term Frequency)

Term Frequency (TF) is a process to calculate the frequency of the number of occurrences of words in a dataset. Because the length of each sentence can be different, usually the value of TF will be divided by the length of the data (the sum of all data in the dataset).

$$tf_{t,d} = \frac{n_{t,d}}{Total\ number\ of\ terms\ in\ document} \quad (3)$$

Description :

tf = frequency of occurrence of words in a data

n = number of occurrences of words in the data

4.3.2. IDF (Inverse Document Frequency)

After we get the value of TF (Term Frequency), we continue to calculate from the value of IDF (Inverse Document Frequency). IDF is a counting process to be able to determine how important a word is in the dataset. IDF assesses words that often appear as less important words based on how they appear throughout the document. The smaller the value of this IDF, the less important the word is. Meanwhile, the greater the value of the IDF, the more important the word will be.

$$idf_d = \log\left(\frac{Number\ of\ Document}{Number\ of\ selected\ word\ frequency}\right) \quad (4)$$

4.3.3. TF-IDF (Term Frequency - Inverse Document Frequency)

After we get the TF (Term Frequency) and IDF (Inverse Document Frequency) values, we can calculate the TF-IDF value which is the product of the TF value and IF value. The TF-IDF formula can be seen below:

$$tfidf_{t.d} = tf_{t.d} \times idf_d \quad (5)$$

We can see that the formula above is a multiplication of the TF (Term Frequency) formula and also the IDF (Inverse Document Frequency) formula.

Table 4.2. Tf – IDF Result

Words/ Sentences	good	hotel	nice	room	close	staff	food
1	0	0.21	0.5	0.27	0.085	0	0
2	0.17	0	0.097	0	0.14	0.09	0
3	0.05	0.11	0	0.15	0	0	0.01

4.4. Feature Selection

The next process after the data feature extraction is performed is the feature selection process. Feature selection is the process of reducing irrelevant features and redundant data to select the best features from a feature data set. There are 2 methods that I use to perform feature selection in this research, namely Chi – Square and Information Gain. However, in this study I conducted 3 experiments at the feature selection stage, namely the first one did not use feature selection as in previous studies [1], used Chi-square, and used information gain.

4.4.1. Chi - Square

Chi-square is a feature selection method to test the relationship or effect of two variables and measure the strength of the relationship between one variable and another. Chi-square has a formula that can be seen below:

$$x^2 = \sum \frac{(O - E)^2}{E} \quad (6)$$

Description :

X^2 = Chi Square value

Oi = F = Frequency of observed results (observed value)

Ei = Fe = Expected frequency (expected value)

Table 4.3. Chi - Square Example

Word, Label	O	E	O-E	Square of O-E	(Square of O-E) / E
Room,Negatif	38	44	-6	36	**0.818181818**
Room,Netral	178	172	6	36	**0.209302326**
Room,Positif	44	38	6	36	**0.947368421**
Chi Square Value					**1.98**

Description :

O = Observed Values

E = Expected Values

Table 4.3 is an example of the results of the calculation of the chi-square selection feature. Here I try some hyperparameters by taking 1000, 2000, 3000, 4000, and 5000. The results of this chi square value are directly transferred to the Support Vector Machine algorithm, however, I will explain the next feature selection first, namely Feature Selection using Information Gain.

4.4.2. Information Gain

Similar to chi-square, Information Gain is also a feature selection method that aims to determine attributes that will be used or discarded later. Information gain is carried out in several stages, namely calculating the information gain value for each attribute in the dataset, determining the desired threshold, and improving the dataset by reducing the attribute which is the purpose of this feature selection. Information gain has a formula that can be seen below:

$$Gain(A) = I(D) - I(A) \quad (7)$$

Description:

Gain (A) = Attribute Information A

I (D) = Total entropy

I (A) = Entropy A

Where to calculate the entropy (I (D) in the Information Gain formula above) the

formula is obtained as follows:

$$info(A) = -\sum_{i=1}^{m} \text{pi}\, log_2(\text{pi}) \quad (8)$$

Description :

D = Case set

m = Number of partitions D

pi = Proportion of Di to D

And to calculate the entropy A (I (A) in the Information Gain formula above) has a formula like this:

$$info_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \text{I}(D_j) \quad (9)$$

Description :

D = Case set

A = Attribute

v = Number of partitions A

|Dj| = Number of cases on j partition

|D| = Number of cases in D

I (Dj) = Total entropy in partition

Table 4.4. Entropy Label Data for Example

Label			Entropy
0 (Negative)	1 (Netral)	2 (Positive)	
139	461	873	**0.94**

Table 4.5. Entropy A for Example

		Label			
		0 (Negative)	1 (Netral)	**2 (Positive)**	
Word	room	**87**	**117**	**402**	**606**
	hotel	**102**	**157**	**338**	**597**
	staff	**50**	**87**	**133**	**270**

Total	**1473**
Entropy	**0.69**

From the example entropy value on Table 4.4 and 4.5 we can calculate the information gain value on calculation below :

IG(Label, Word) = E(Label) – E(Label, Word)

= 0.940 – 0.693

= 0.247

Therefore, the result from this example calculation of information gain produce value is 0.247. At this stage, I tried several hyperparameters, namely with information gain values above -0.3, -0.2, and -0.1 to be used for the next classification stage. After performing the Information Gain feature selection stage, features with a high gain value will be obtained and become new features to be included in the algorithm using the Support Vector Machine as in previous research [1].

4.5. Classification

In this study, I use the Support Vector Machine (SVM) algorithm which is one of the methods in supervised learning which at this time I am using it for classification although it can also be used for regression.

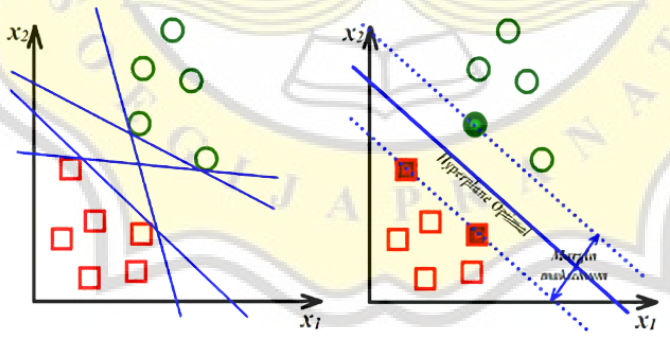

Figure 4.6 Support Vector Machine

It can be seen in Figure 4.5 that the Support Vector Machine (SVM) classification method tries to find the best hyperplane function among an unlimited number of functions. Hyperplane is a function that can be used to separate between classes. In 2 dimensions, the function used in this hyperplane for classification between classes is called line whereas. The

best hyperplane is the dividing line between 2 data classes in the input space which can be determined by measuring the margin of the hyperplane and finding its maximum point. Where margin means the distance between the hyperplane and the closest data from each class. The data closest to the hyperplane is called the support vector.

Table 4.6. Data Composition

Data Compare	
Data Training	Data Testing
80%	20%
70%	30%
60%	40%
50%	50%
40%	60%

In table 4.1, it is explained from the results of preprocessing that there are 20492 data divided into 3 sentiment classes, namely positive (2), neutral (1), and negative (0). The data that has been normalized before being entered into the Support Vector Machine (SVM) algorithm, the data is divided into 2, namely training data and test data performed during classification. In this test, the data is divided 5 times with different input training data and testing data.

The data that was trained and tested has been divided and each experiment is classified using the Support Vector Machine (SVM) algorithm with a kernel function that maps linear data so that it gets a new dataset of learning models in each experiment. The results of the learning model are classified by testing 5 times were in each test using a matrix with a size of 3 x 3 as a representative of the actual class and the predicted class.

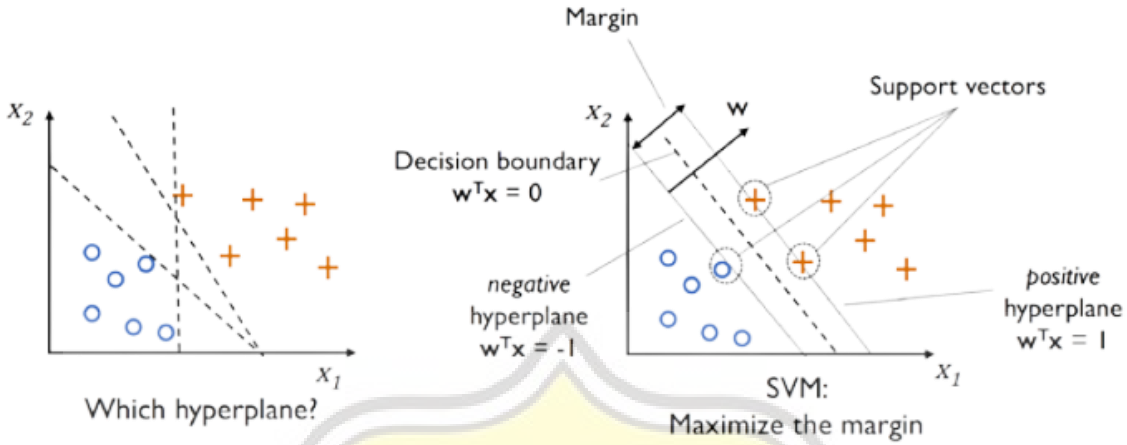

Figure 4.7 Hyperplane

The concept of the Support Vector Machine algorithm can be described as an attempt to find the best hyperplane that serves as a dividing line between the two classes. In Figure 4.14 the left and right sections show a pattern that is part of two classes, namely positive and negative. Patterns belonging to the positive class are represented by an orange plus sign, while the negative class is represented by a blue circle. The classification problem can be explained by trying to find the best hyperplane that separates the two classes. Alternative dividing lines are shown in Figure 4.14 on the left.

The best dividing hyperplane between the two classes is determined by measuring the hyperplane margin and finding its maximum point. Margin is the distance between the hyperplane and the nearest point or data in each class. The closest pattern is called the support vector. The dash-shaped line in Figure 4.14 on the right can be said to be the best hyperplane. It is called the best hyperplane because it is located right in the middle between the two classes, while the plus sign is orange and the blue circle inside the black circle is called the support vector. Efforts to find this hyperplane is the most important part of the classification of the Support Vector Machine algorithm. To get the perfect hyperplane location, it can be defined by the following formula:

$$f(x) = w^T\mathrm{x} + \mathrm{b} \tag{20}$$

So, based on the formula in Figure 4.15, the equation can be obtained:

$$[(w^T.x_i)] + \mathrm{b} \geq 1 \text{ untuk } y_i = +1 \qquad (31)$$
$$[(w^T.x_i)] + \mathrm{b} \leq -1 \text{ untuk } y_i = -1$$

With the description of xi as the training data set, i as 1,2,…., n , and yi as the class label of xi . The largest margin can be found by maximizing the value of the distance between the hyperplane and its closest point and can be formulated as follows:

$$\frac{1}{\left|\left|\underset{w}{\rightarrow}\right|\right|} \qquad (12)$$

With this, it can be formulated as a quadratic programming problem which means finding the minimum point that can be seen in Figure 4.16 with a note to pay attention to the constraints in Figure 4.17.

$$\min_{\vec{w}} \tau(w) = \frac{1}{2}\left|\left|\underset{w}{\rightarrow}\right|\right|^2 \qquad (13)$$

$$y_i\left(\underset{w}{\rightarrow}.\underset{x_l}{\rightarrow}\right) - 1 \geq 0, \forall i \qquad (14)$$

This problem can be solved by computational techniques, one of which is the Langrange Multiplier which can be seen in the formula below:

$$L(\underset{w}{\rightarrow}, b, a) = \frac{1}{2}\left|\left|\underset{w}{\rightarrow}\right|\right|^2 - \sum_{i=1}^{l}\alpha_i(y_i((\underset{w}{\rightarrow}.\underset{x_i}{\rightarrow} + b) - 1)), i = 1,2, \ldots, l \qquad (15)$$

$$\sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l}\alpha_i y_j \alpha_j y_i \underset{x_i}{\rightarrow}.\underset{x_j}{\rightarrow} \qquad (16)$$

αi is the Langrange Multiplier which can be 0 (zero) or positive αi ≥ 0 (zero). The optimal value of the formula below can be seen by looking at the value of L against $\vec{w}$ and b and can maximize the value L against αi. Based on basically the optimal point $L = 0$, the formula contained in Figure 4.18 can set the maximization of the problem which only contains αi.

From the results of the Langrange Multiplier on above calculations can be obtained the value of αi which is positive. Data that is related or correlated with αi positive is what can be

called a support vector which is on Support Vector Machine algorithm.

$$\alpha_i \geq 0(i = 1,2, \dots, l) \quad \sum_{i=1}^{l} \alpha_i y_i = 0 \tag{17}$$

To measure the classification performance on the original data and the result data from the classification model that has been done, you can use the confusion matrix which we will discuss in the next stage.

4.6. Confusion Matrix

Confusion matrix is a process of measuring the performance / performance of the classification model where the output can be in the form of 2 or more classes. There are four terms that can be said to be representative of the results of the classification process in the confusion matrix, namely True Positive, False Positive, True Negative and False Negative. From the results of these 4 categories, the values of accuracy, precision, recall, and F-1 Score can be calculated. Accuracy is how accurate the model is in classifying correctly. Precision is the accuracy between the requested data and the prediction results provided by the model. Then, recall is the success of the model in rediscovering information. While the F-1 Score is the average comparison between precision and recall which is weighted. These four things can be formulated as follows:

$$Accuracy = \frac{TP}{Number\ of\ data}$$
$$Precision = \frac{TP}{(TP + FP)}$$
$$Recall = \frac{TP}{(TP + FN)} \tag{18}$$
$$F - 1\ Score = \frac{(Precision + Recall)}{2}$$

At the feature extraction stage, it is carried out 3 times with a change of method. The first one does not use feature extraction, the second uses chi-square feature extraction, and the third uses information gain feature extraction. Each of these three methods was carried out 5 times with the distribution of different datasets as given above. After doing everything, I can determine which method is the best of all.