

## CHAPTER 4

### ANALYSIS AND DESIGN

#### 4.1 Analysis

The analysis of this project is in the form of calculating the accuracy, precision, recall and f1-score of each algorithm which will show the performance of each algorithm. The purpose of the above calculations is to measure the performance of each algorithm in each calculation scheme that has been made. And to compare the performance of the two algorithms.

From the illustration above, TP is true positive, that is, data testing is positive and it is true positive. Whereas TN is true negative, the opposite of TP, namely negative testing data and true negative. FP is false positive where data testing is positive which should be negative. And FN is false negative, that is negative testing data but the data should be positive. The results of the TP, TN, FP, FN will be calculated which results in accuracy, precision, recall and f1-score. Accuracy is the ratio of the correct prediction to the overall data. Precision is the ratio of a positive true prediction to the overall positive predicted outcome. Recall is the ratio of true positive predictions to the overall true positive data. And finally, the f1-score is a weighted average comparison of precision and recall.

#### 4.2 Vector Space Model Analysis

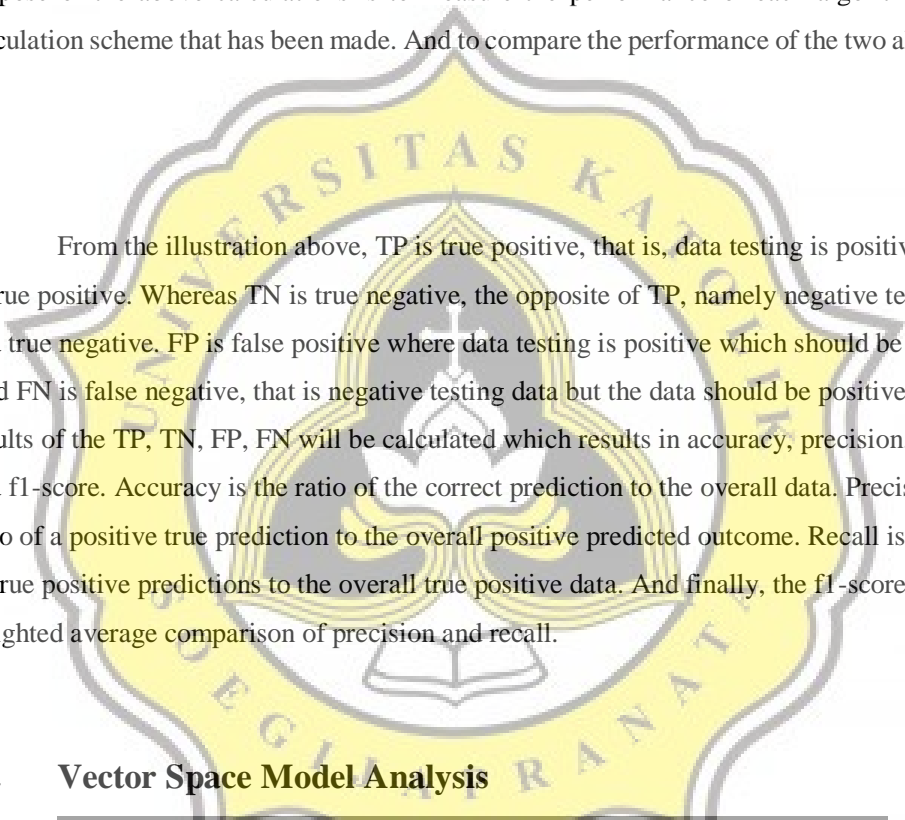

$$Sim(q, d_j) = \frac{q \cdot d_j}{|q| * |d_j|} = \frac{\sum_{i=1}^t w_{iq} \cdot w_{ij}}{\sqrt{\sum_{j=1}^t (w_{iq})^2 * \sum_{i=1}^t (w_{ij})^2}}$$

Illustration 4.2.1: Cosine Similarity

Q = bobot queri

Dj = bobot data training

|q| = akar kuadrat bobot queri

$|dj|$  = akar kuadrat bobot data training

The picture above is a formula for cosine similarity which is used to calculate the similarity between a document and the entered query. By calculating  $q$  dot product  $dj$  divided by  $|q|$  dot product  $|dj|$ .

This method uses the cosine equation to get maximum results. The following are some of the results obtained from the Vector Space Model method.

Table 4.2.1: Vector Space Model Data

Document	Value	Testing	Training	Result
1	0.14185072 710989607	Positive	Positive	TP
2	0.14884784 607990278	Negative	Negative	TN
3	0.32652987 196824484	Negative	Positive	FN
4	0.15406863 113045385	Positive	Negative	FP

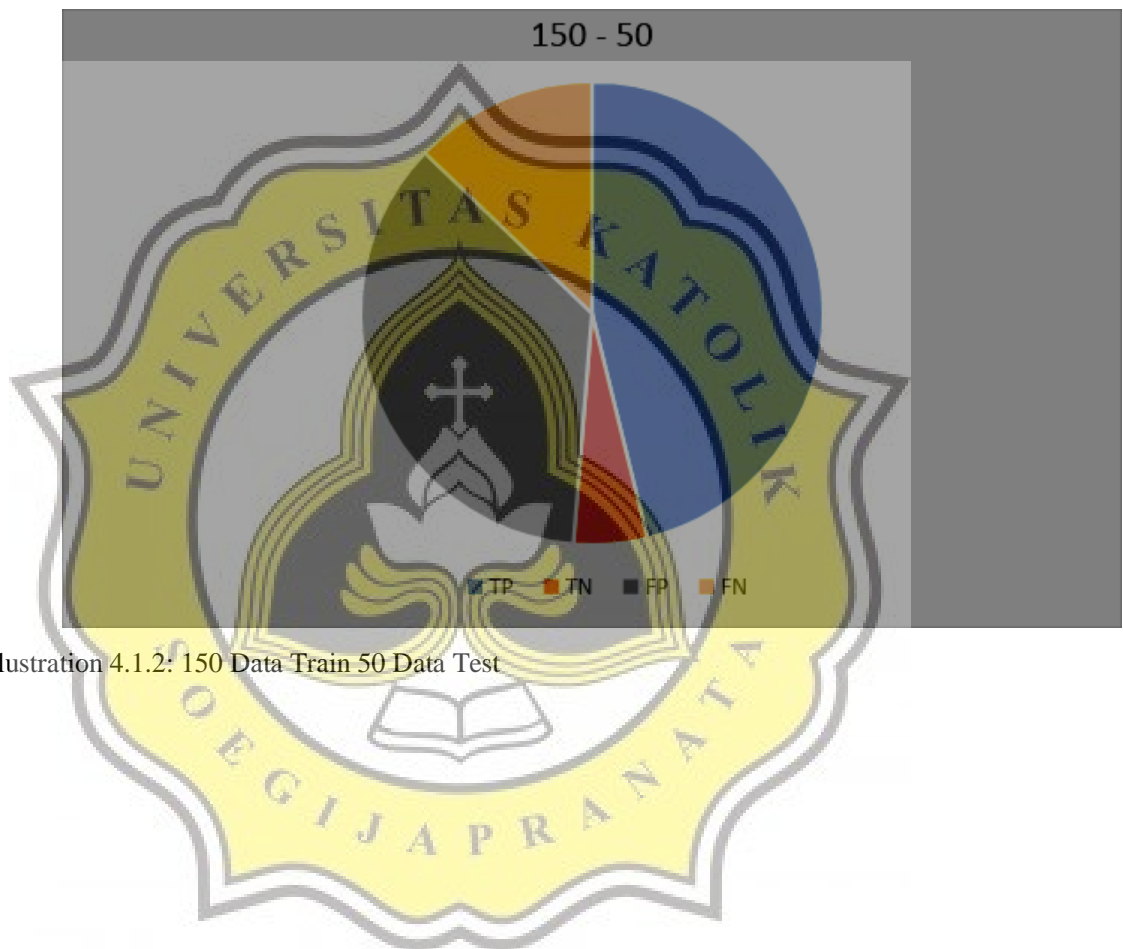
The image above is an example of the results of testing data using the vector space model. It is known that the data train has a value as shown in the picture above then the test results are either positive or negative. Then training is an evaluation manual which states that the document should be positive or negative. And the results are the results of testing data in the form of TP, TN, FP, FN.

There are 3 testing schemes with 150 training data and 50 test data, 100 training data and 98 test data, and 50 training data and 148 test data. From the three schemes, the following results were obtained:

Table 4.2.2: Vector Space Model Analysis 150 Training 50 Testing

TP	18
TN	12
FN	5
FP	14

By using the 80% train data scheme and 20% test data, the results are 36% for true positive results, 24% for true negative results, 10% for false negatives and 28% for false positives



. Illustration 4.1.2: 150 Data Train 50 Data Test

Table 4.2.3: Vector Space Model Analysis 100 Training 98 Testing

TP	43
TN	11
FN	26
FP	18

By using the second scheme, namely 50% train data and 50% test data, the results are 31% true positive, 17% true negative, 10% false negative and 40% false positive



Illustration 4.2.3: 100 Data Train 98 Data Test

Table 4.2.4: Vector Space Model Analysis 50 Training 148 Testing

TP	59
TN	22
FN	21
FP	46

Whereas for the last scheme, 20% train data and 80% test data, the results are 39% true positive, 14% true negative, 14% false negative and the rest 30% false positive.

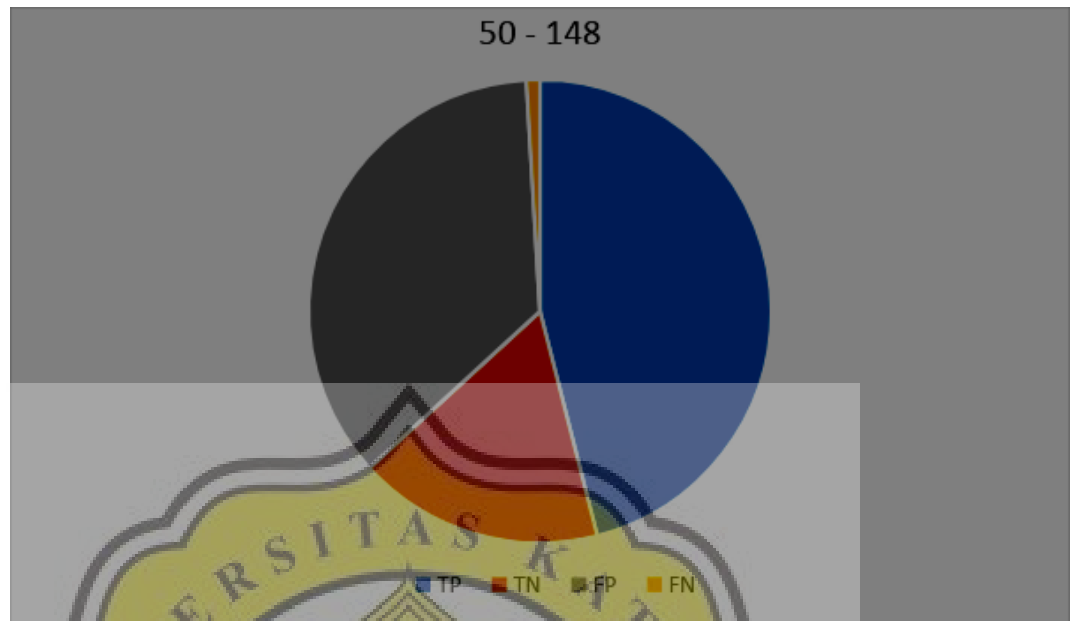


Illustration 4.2.4: 50 Data Train 148 Data Test

From the data that has been obtained according to the scheme above, the following results are obtained:

Table 4.2.5: Vector Space Model Calculation 150 Training 50 Testing

Accuracy	61%
Precision	78%
Recall	56%
F1-Score	65%

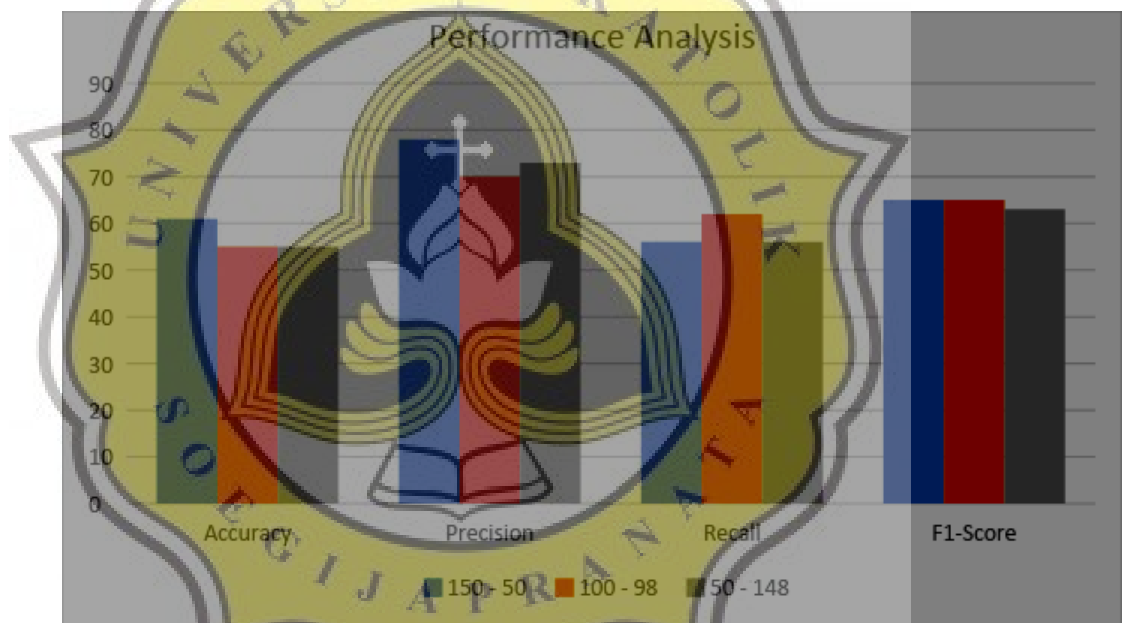
Table 4.2.6: Vector Space Model Calculation 100 Training 98 Testing

Accuracy	55%
Precision	70%
Recall	62%
F1-Score	65%

Table 4.2.7: Vector Space Model Calculation 50 Training 148 Testing

Accuracy	55%
Precision	73%
Recall	56%
F1-Score	63%

From the results of the accuracy, precision, recall and f1-score above, it was found that the 80% train data and 20% test data schemes have maximum results compared to other schemes. With 61% accuracy, 78% precision, 56% recall, and 65% f1-score. This shows that with more training data, it can produce a more varied index for testing data. So as to maximize algorithm performance to get maximum results.



. Illustration 4.2.5: Accuracy, Precision, Recall, F1-Score Comparison

### 4.3 Naive Bayes

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

$P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$

Illustration 4.3.1: Naive Bayes

X = data dengan class yang belum diketahui

C = hipotesis data merupakan suatu class spesifik

$P(c|x)$  = probabilitas hipotesis berdasar kondisi (posterior probability)

$P(x|c)$  = probabilitas berdasarkan kondisi pada hipotesis (likelihood)

$P(x)$  = probabilitas c (predictor prior probability)

This method uses statistical calculations from each data. This method is easier to use and understand than the Vector Space Model. The following is an example of data obtained from the Naive Bayes method.

Table 4.3.1: Naive Bayes Data

Document	Testing	Training	Result
1	Positive	Negative	FP
2	Positive	Positive	TP
3	Negative	Positive	FN
4	Positive	Negative	FP

Similar to the VSM test, this test also uses 3 schemes. By using 150 training data and 50 test data, 100 training data and 98 test data, and 50 training data and 148 test data. From the three schemes, the following results were obtained:

Table 4.3.2: Naive Bayes Analysis 150 Training 50 Testing

TP	18
TN	12
FN	4
FP	15

By using the 80% train data scheme and 20% test data, the results are 36% for true positive results, 24% for true negative results, 8% for false negatives and 30% for false positives.

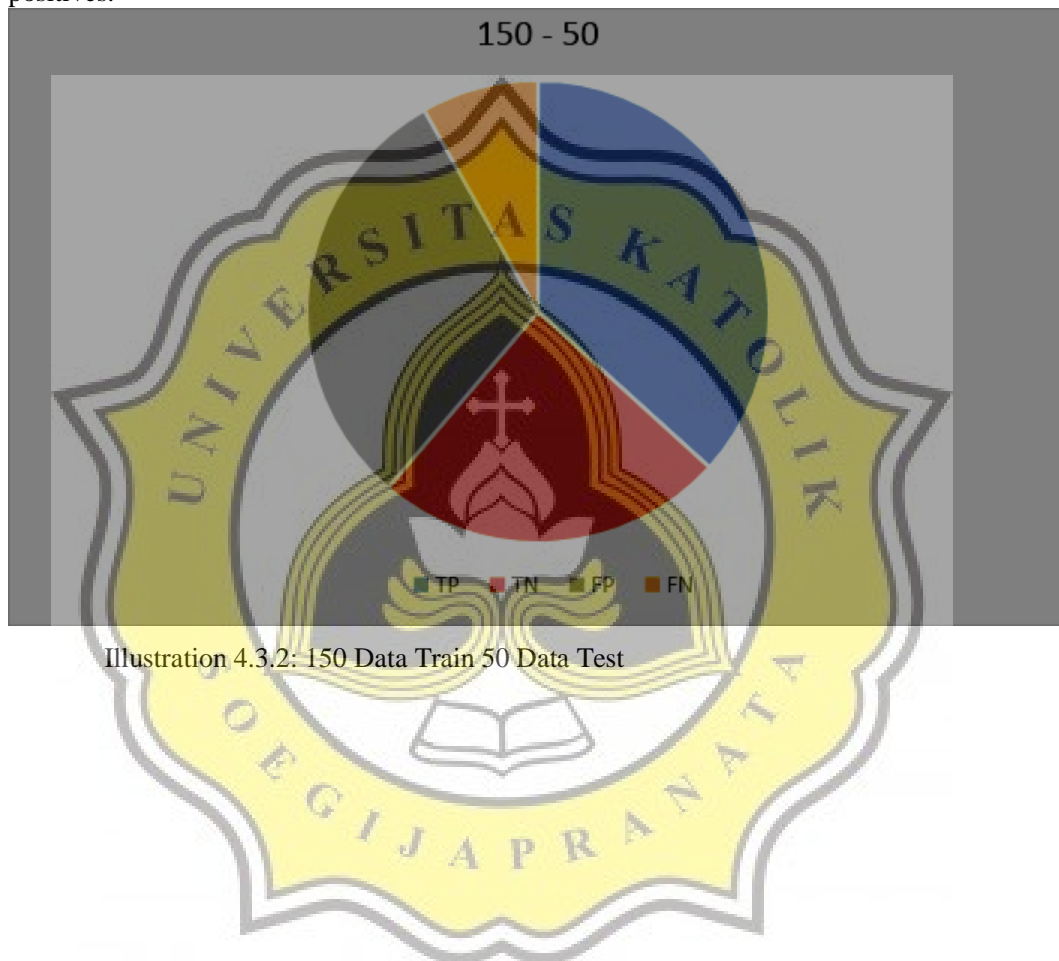


Illustration 4.3.2: 150 Data Train 50 Data Test



Table 4.3.3: Naive Bayes Analysis 100 Training 98 Testing

TP	39
TN	17
FN	11
FP	31

By using the second scheme, namely 50% train data and 50% test data, the results are 39% true positive, 17% true negative, 11% false negative and 31% false positive

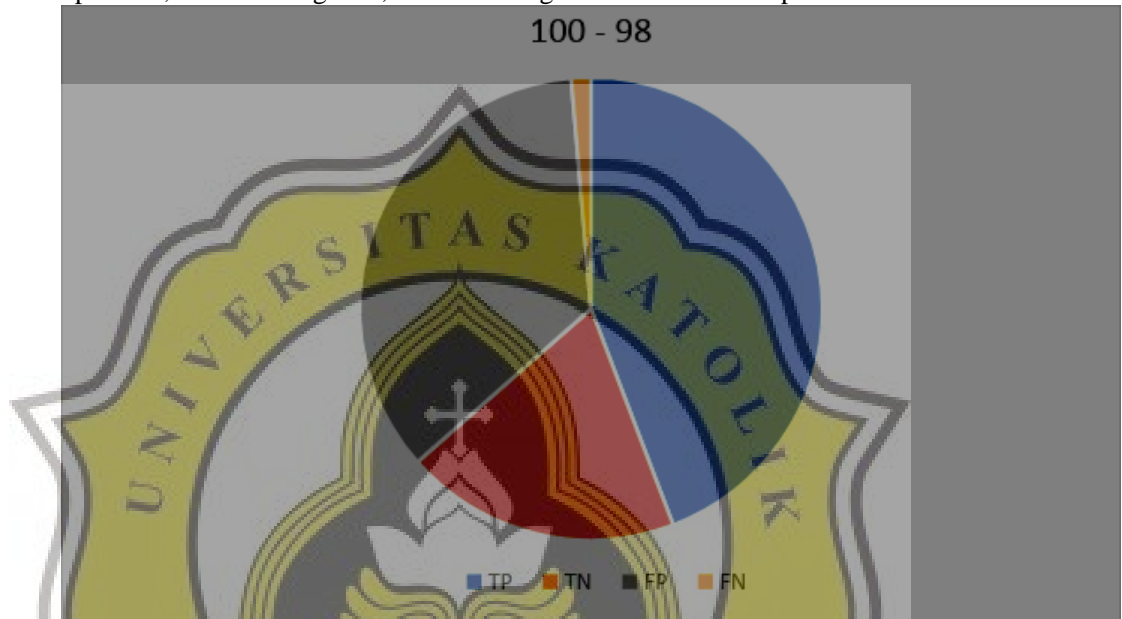


Illustration 4.3.3 : 100 Data Train 98 Data Test

Table 4.3.4: Naive Bayes Analysis 50 Training 148 Testing

TP	39
TN	36
FN	9
FP	64

Whereas for the last scheme, 20% train data and 80% test data, the results are 26% true positive, 24% true negative, 6% false negative and the rest 42% false positive.

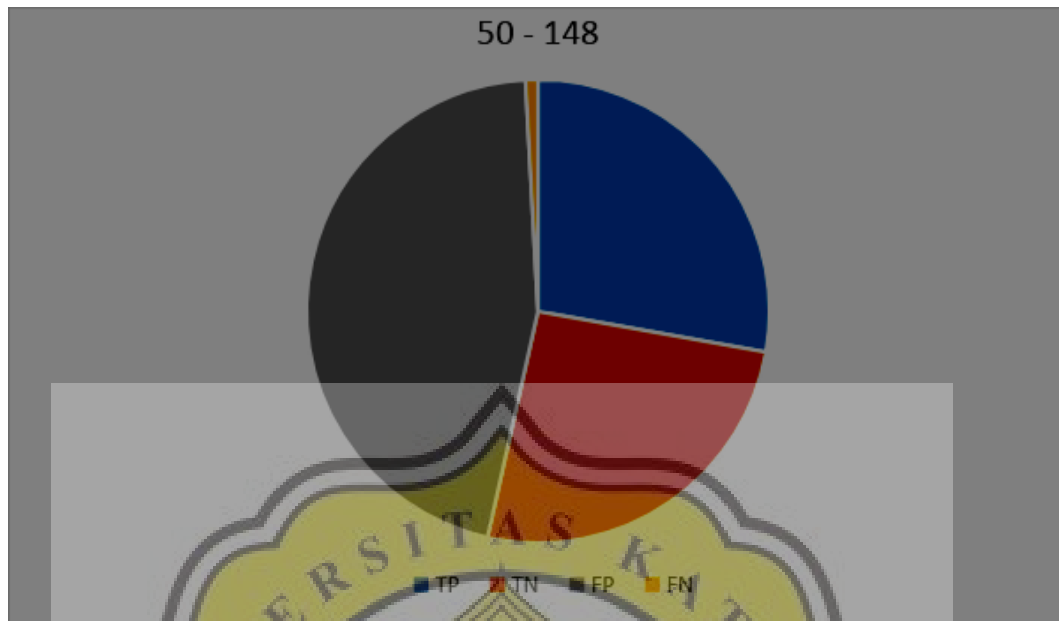


Illustration 4.3.4: 50 Data Train 148 Data Test

From the data that has been obtained according to the scheme above, the following results are obtained:

Table 4.3.5: Naive Bayes Calculation 150 Training 50 Testing

Accuracy	61%
Precision	81%
Recall	54%
F1-Score	65%

Table 4.3.6: Naive Bayes Calculation 100 Training 98 Testing

Accuracy	57%
Precision	65%
Recall	56%
F1-Score	60%

Table 4.3.7: Naive Bayes Calculation 50 Training 148 Testing

Accuracy	50%
Precision	81%
Recall	37%
F1-Score	51%

Testing using naive bayes is also the same, with the first scheme getting maximum performance. By using 80% train data and 20% test data, the results obtained 61% accuracy, 81% precision, 54% recall and an f1-score of 65%. From this testing, it was found that more training data could result in maximum performance.

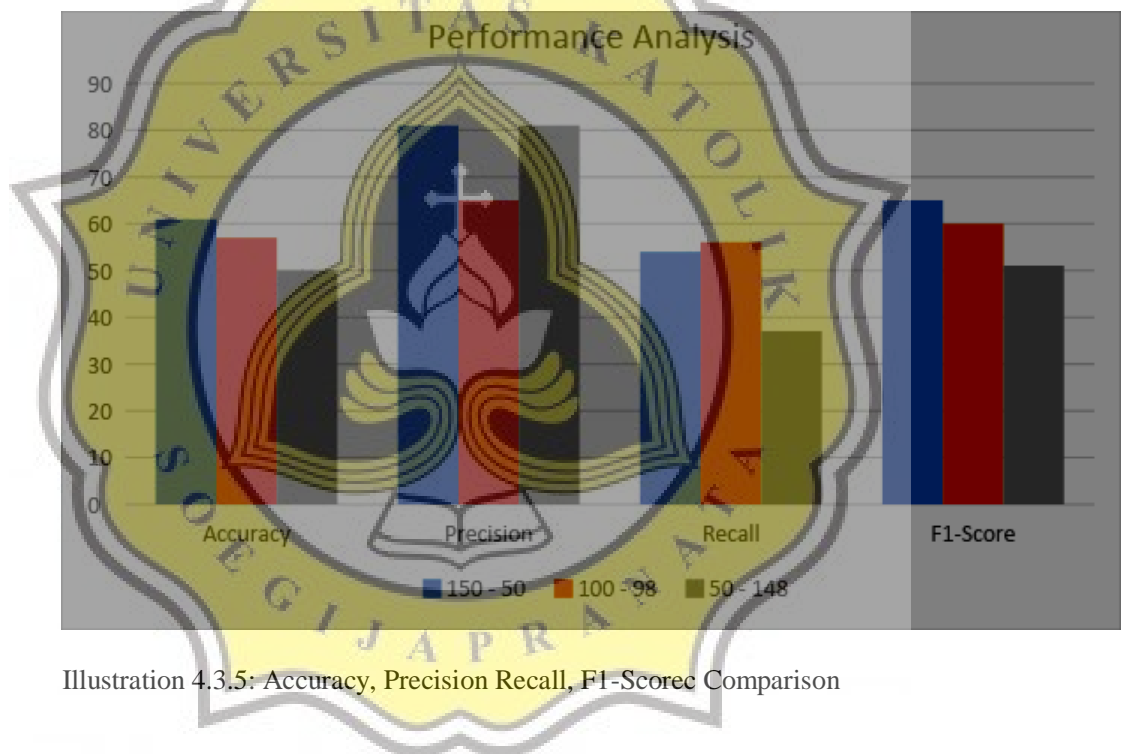


Illustration 4.3.5: Accuracy, Precision Recall, F1-Score Comparison

#### 4.4 Desain

The two methods used in this study are almost the same, the only difference is the calculation of the Vector Space Model and Naive Bayes algorithms. The following is the flowchart for this research.

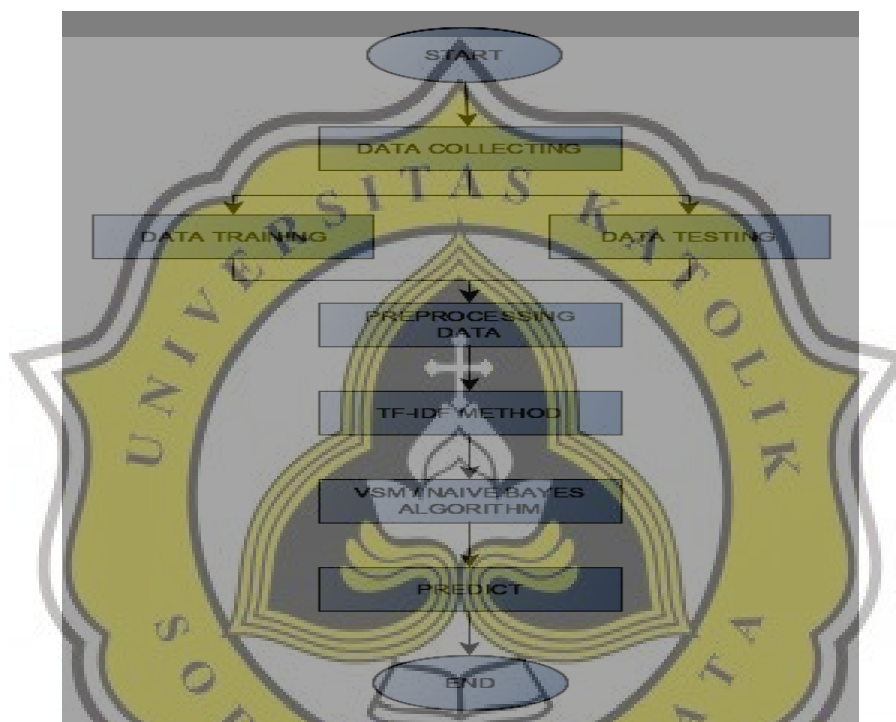


Illustration 5.1: Flowchart VSM & Naive Bayes

The two algorithms have almost the same process flow. Because these two algorithms are supervised classifiers that require weight calculations. Starting with data collecting using python scrape method. The data taken is as many as 300 data which has the attribute #socialdistancing. Then this data is divided equally, namely 150 which will later be divided into 3 test schemes. Starting with 80% train data, 20% test data, 50% train data 50% test data, and 20% train data 80% test data.

After the data is separated according to the test scheme, data preprocessing is carried out. Preprocessing data here consists of 4 steps, starting with case folding followed by tokenizing, removing stopword and ending with tokenization. This explanation has been

discussed in Chapter 3. Which is then followed by calculating the weight of each word using the TF-IDF method. An example of calculating tf-idf has also been explained in Chapter 3.

After finishing with the calculation of the weight of each word, it is continued by entering the algorithm used. Because the vector space model and naive bayes are supervised classifiers, the flow is almost the same. What distinguishes it is the calculation to determine the final result or predict a different label.

After the prediction is complete, the data is matched manually to get TP, TN, FP, FN. To later calculate the level of accuracy, precision, recall and f1-score. This section will show which scheme has the maximum performance in terms of accuracy, precision, recall and f1-score. After that, compare which algorithm has the maximum result.

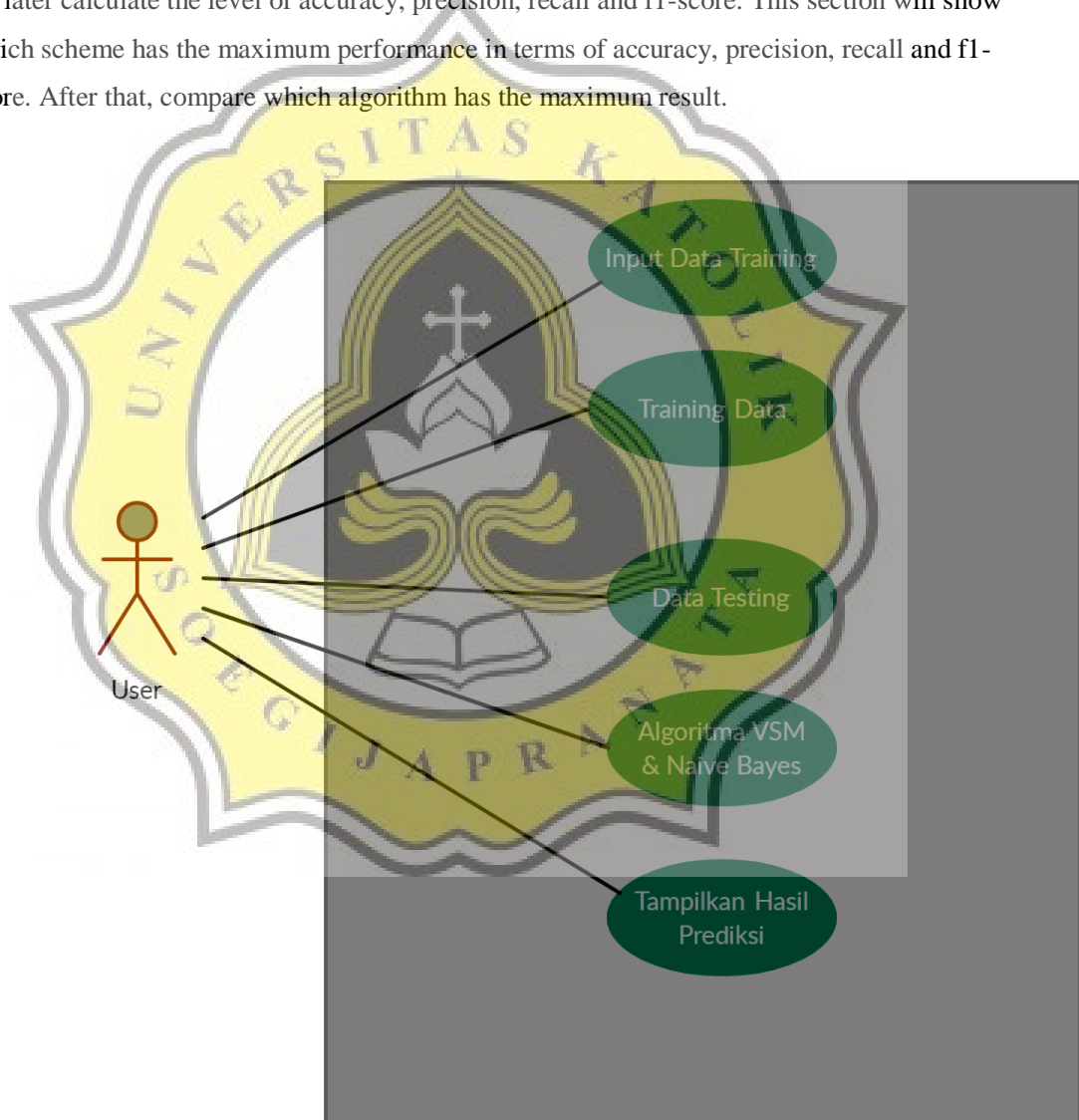


Illustration 5.2: Use Case Diagram VSM & Naive Bayes

The image above is a use case diagram of this project. The picture above shows the role of the user running the program. Users take data then input training data, followed by training data. After that the user enters the testing data which will later be tested using the Vector Space Model algorithm and Naive Bayes. Users will get classification results whether labeled positive or negative.

