CHAPTER 4
ANALYSIS AND DESIGN

4.1 Analysis

4.1.1 Collecting Data

Data collection is obtained from the web scraping process using a Google Chrome Extention tool called Web Scraper. Data collected using Web Scraper in the form of CSV file is then imported into the MYSQL database. The data collected were 3 book categories and the total of the training data is 60 data.

Table 4.1: Training Data Example

title	deskripsi	kategori
FUN CICAN: BAJAK LAUT DAN MONSTER AIR	Saatnya bermain telah tiba. Cican dan teman-teman berpetualang mencari harta karun. Mereka bertemu makhluk yang sangat besar dan menakutkan! Apa yang akan terjadi pada cican dan teman-temannya ya...	kids
Kisah Mulia Princess di Dunia Cerita Putri yang Bijaksana & Baik Hati	Membangun Karakter Anak lewat Cerita Princess di Dunia Berbagai kisah tentang Princess selalu meryadi hal yang menarik bagi anak-anak. Sosok princess yang selalu tergambar baik hati dan cantik, berbalut busana yang cantik pula, masih merjjadi potret menarik bagi anak-anak perempuan...	kids
Dongeng Pertamaku : Cinderella (Level 4)	Cinderella gadis yang amat malang. Dia selalu dipaksa memasak dan merawat kakak-kakak tirinya. Saat ada pesta dansa di Istana, Ibu tirinya yang jahat melarang Cinderella pergi ke sana. Lalu, Nenek Peri datang dan segalanya berubah...	kids
Jokowi Memimpin dengan HATI	okowi. Siapa sih yang tidak mengenal sosok nomor satu ini?	biography

	Seorang bapak yang ramah senyum dan dekat dengan rakyat kecil ini, telah menjadi panutan dan pemimpin bagi seluruh rakyat Indonesia. Namun, tak banyak yang tahu tentang kisah di balik kesuksesannya....	
Sir Alex Ferguson : Peracik Strategi Terbaik Sepanjang Masa	Bagi para pecinta bola terutama klub liga Inggris Manchester United, tentu figure seorang Sir Alex Ferguson adalah sesosok figur yang sangat dihormati dan dikagumi. Sosok figur yang sangat sentral dan kharismatik serta disegani baik oleh kawan maupun lawan. Melalui tangan dinginnya lah klub Setan Merah berhasil mencapai kejayaannya di era sepakbola modern ini....	biography
JEAN PAUL SARTRE : Filsuf Ekstensialisme Imajinatif	Biografi singkat Jean Paul Sartre adalah sebuah pengantar untuk mengenal sosok sang filsuf perancis terkenal ini, yang mudah dipahami namun cukup lengkap dan mendalam. Sartre merupakan anggota gerakan bawah tanah, penulis naskah drama serta orang yang berpengaruh dalam ekhidupan intelektual dan politik di Perancis....	biography
Sarapan Pagi Penuh Dusta	Jujur, aku selalu membandingkanmu dengan kekasihku. Aku tahu itu salah, kekasihku orang yang sangat baik. Tapi kebaikan sering tidak berbanding lurus dengan rasa suka...	literature
Perahu Kertas - Sapardi Djoko Damono	Di tangan anak-anak, kertas menjelma perahu Sinbad yang tak takluk kepada gelombang, menjelma burung yang jeritnya membukakan kelopak-kelopak bunga di hutan...	literature
Isyarat Cinta Yang Keras Kepala	Kumpulan Cerpen karya Puthut EA "Aku justru harus berani menghadapi seluruh peristiwa yang telah lewat,	literature

	dan bukan justru menghindarinya. Kenangan tidak bisa dihilangkan. Kenangan hanya bisa dihadapi atau diperam dengan risiko membusuk di dalam."...	

4.1.2 Text Preprocessing

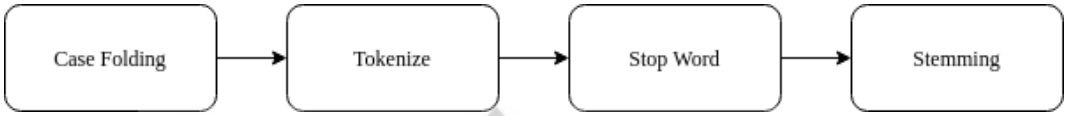

Illustration 4.1: Text Preprocessing Chart

After all data imported to database, the next step is to do text preprocessing. Definition of Text Preprocessing is a process of transforming unstructured data into structured data for further mining process. In short, text preprocessing is a method to convert text into term index. The goal is to generate a set of term indexes that can represent documents. Text preprocessing has few step to do :

4.1.2.1 Case Folding

Case folding is a process of changing all characters in a document into a lowercase. Case folding will only store alphabetic characters so numberic characters and other characters will be eliminated.

Example :

Table 4.2: Example of Case Folding Process

Before	Jujur, aku selalu membandingkanmu dengan kekasihku. Aku tahu itu salah, kekasihku orang yang sangat baik. Tapi kebaikan sering tidak berbanding lurus dengan rasa suka. Sarapan Pagi Penuh Dusta ï¿½ Kumpulan cerita pendek ini membawamu bertamasya ke keseharian yang tak biasa. Kisah-kisah yang dihadirkan menyelami dunia batin para tokohnya. Dinarasikan secara ringan

	dan tak rumit, tapi tetap memesona. Hanya saja, yang perlu diwaspadai, cerita-cerita di dalamnya menyimpan kejutan tak terduga.
After	jujur aku selalu membandingkanmu dengan kekasihku aku tahu itu salah kekasihku orang yang sangat baik tapi kebaikan sering tidak berbanding lurus dengan rasa suka sarapan pagi penuh dusta kumpulan cerita pendek ini membawamu bertamasya ke keseharian yang tak biasa kisah kisah yang dihadirkan menyelami dunia batin para tokohnya dinarasikan secara ringan dan tak rumit tapi tetap memesona hanya saja yang perlu diwaspadai cerita cerita di dalamnya menyimpan kejutan tak terduga

4.1.2.2 Tokenizing

Tokenizing is the process of dividing text that can be in the form of sentences, paragraphs or documents into certain tokens / parts.

Example :

Table 4.3: Example of Tokenizing Process

Tokenized Words from Desription		
kumpulan	cerita	pendek
ini	bertamasya	kebaikan
yang	orang	dunia
membawamu	rumit	narasumber

4.1.2.3 Stop Word

Stop word is the process of removing words that have low information from a text, so that it can focus on important words instead. Examples of stopwords in Indonesian are "yang", "dan", "di", "dari", etc.

Example :

Table 4.4: Example of Stop Word Process

Result of StopWord process	
kumpulan	bertamasya
cerita	orang
pendek	rumit
membawamu	kebaikan
narasumber	dunia

4.1.2.4 Stemming

Stemming is the process of removing word inflection into its basic form, but the basic form does not mean the same as root word.

Example :

Table 4.5: Example of Stemming Process

Result of Stemming process	
kumpul	tamasya
cerita	orang
pendek	rumit
bawa	baik
narasumber	dunia

4.1.3 Term Frequency-Inverse Document Frequency (TF-IDF)

Term Frequency-Inverse Document Frequency (TF-IDF) method is a method for calculating the weight of every word which most commonly used in information retrieval. It will calculate the value of Term Frequency (TF) and Inverse Document Frequency (IDF) on each keyword/token in each data.

TF-IDF Formula :

$$W_{dt} = tf_{dt} x\, idf_t$$

where :

- W : weight term-t to d-document
- tf : frequency of term on each document

- idf : Inversed Document Frequency

And idf value obtanied from :

$$idf_t = \log\left(\frac{N}{df}\right)(1)$$

where :

- N : amount of all document
- df : amount of all document which consist term-t

Example :

Term	tf			df	N	idf	W		
	D1	D2	D3				D1	D2	D3
anak	7	1	6	3	3	0	0	0	0
buku	2	1	1	3	3	0	0	0	0
baca	0	2	1	2	3	0.1761	0	0.3522	0.1761
main	2	0	1	2	3	0.1761	0.3522	0	0.1761
petualang	2	0	0	1	3	0.4771	0.9542	0	0
cerita	3	0	0	1	3	0.4771	1.4314	0	0
cerdas	1	0	0	1	3	0.4771	0.4771	0	0
orangtua	1	0	0	1	3	0.4771	0.4771	0	0
imajinasi	1	0	0	1	3	0.4771	0.4771	0	0
siswa	1	0	0	1	3	0.4771	0.4771	0	0
kreatif	1	0	0	1	3	0.4771	0.4771	0	0
sosok	0	1	0	1	3	0.4771	0	0.4771	0
bangga	0	1	0	1	3	0.4771	0	0.4771	0
hidup	0	1	0	1	3	0.4771	0	0.4771	0
tumbuh	0	1	0	1	3	0.4771	0	0.4771	0
sukses	0	1	0	1	3	0.4771	0	0.4771	0
bawa	0	1	0	1	3	0.4771	0	0.4771	0
gelar	0	1	0	1	3	0.4771	0	0.4771	0
kertas	0	0	2	1	3	0.4771	0	0	0.9542
jerit	0	0	1	1	3	0.4771	0	0	0.4771
bunga	0	0	1	1	3	0.4771	0	0	0.4771
kitab	0	0	1	1	3	0.4771	0	0	0.4771
tuan	0	0	2	1	3	0.4771	0	0	0.9542
suci	0	0	1	1	3	0.4771	0	0	0.4771
hati	0	0	1	1	3	0.4771	0	0	0.4771

Illustration 4.2: TF-IDF Result

The result of TF-IDF process will be use as vector parameters in classification using the LVQ algorithm.

4.1.4 Data Processing using LVQ Algorithm

At this phase is the implementation of the LVQ algorithm into the classification process. Classification proces in LVQ algorithm divided into two stages : Training stage and Testing stage.

In this program will classify data into 3 categories, which is kids, biography and literature.

4.1.4.1 LVQ Algorithm Analysis on Training Data

First step after all weight of term obtained from TF-IDF process, it will find representation data from each categories for the initiation weight for LVQ algorithm. The representation is a data of each categories that have the most common terms from all data of each categories. Data representation is obtained by summing the same terms in each category.

Table 4.6: Example of Training Data Representation

title	deskripsi	kategori
FUN CICAN: BAJAK LAUT DAN MONSTER AIR	main cican teman teman petualang cari harta karun makhluk akut cican teman teman pesan cerita tampil kuat hubung orang tua anak diskusi etc....	kids
Sir Alex Ferguson : Peracik Strategi Terbaik Sepanjang Masa	cinta bola klub liga inggris manchester united figure sir alex ferguson sosok figur hormat kagum sosok figur sentral kharismatik segan etc...	biography
Perahu Kertas - Sapardi Djoko Damono	tangan anak anak tangan anak anak kertas jelma takluk gelombang jelma burung jerit buka lopak lopak bunga hutan mulut anak anak jelma kitab etc...	literature

Then after the representation data is found, the data will be the initiation of weights for the rest of the training data

- Initiation weight from each class

data	x1	x2	x3	x4	x5	x6	x7	class
1	0.3521	1.7869	0.7658	0.6532	0	0	0	1
2	0	0.2552	0	0	0.3521	0.3521	0	2
3	0.176	1.5316	0.5105	0	0	0	1.3064	3

- Training data

data	x1	x2	x3	x4	x5	x6	x7	class
1	0.5282	0.5105	0.2552	0.6532	0.3521	0.3521	0	1
2	0.176	1.2763	0	0	0	0	0	1
3	0.176	0	0.7658	0	0.3521	0.3521	0	2
4	0	0	0	0	0.7043	0.3521	0	2
5	0.176	0	0	0	0	0	0	3
6	0	0	0.7658	0	0	0	0.6532	3

Second step is determine learning rate (α) and maximum epoch. But every epoch will reduce learning rate (α) by 0.1* α.

learning rate (α) = 0.05

MaxEpoch = 3

Third step is calculate Square of Euclidean Distance of every training data with initial weight.

DATA 1 → ‖x-wj‖

CLASS 1 : 1.47270816864714

CLASS 2 : 0.914314284040231

CLASS 3 : 1.90083354347507

The smallest value of the calculation will determine the data new category. Output (minimum value of results) 0.914314284040231 into class 2

Because the output has not the same value with the original class then the initial weight will be updated with the formula:

$$w_j(new) = w_j(old) - \alpha[x - w_j(old)]$$

But if the output has the same value with the original class then the initial weight will be updated with the formula:

$$w_j(new) = w_j(old) + \alpha[x - w_j(old)]$$

So updated the new weight will show below :

w2 new	-0.02641	0.242435	-0.01276	-0.03266	0.3521	0.3521	0

This process is done on each data. The results of the process are as follows:

DATA 2 → ‖x-wj‖

CLASS 1 : 1.14229744375097

CLASS 2 : 1.16577225885891

CLASS 3 : 1.42564697593759

- Output (minimum value of results) 1.14229744375097 into class 1
- Update Weight -> $w_j(new) = w_j(old) + \alpha[x - w_j(old)]$

w1 new	0.343295	1.76137	0.72751	0.62054	0	0	0

DATA 3→ ‖x-wj‖

CLASS 1 : 1.94033417988371

CLASS 2 : 0.840813229275681

CLASS 3 : 2.0894033669926

- Output (minimum value of results) 0.840813229275681 into class 2
- Update Weight -> $w_j(new) = w_j(old) + \alpha[x - w_j(old)]$

w2 new	-0.0162895	0.23031325	0.026168	-0.031027	0.3521	0.3521	0

DATA 4→ ‖x-wj‖

CLASS 1 : 2.18051125326722

CLASS 2: 0.423086066762795

CLASS 3 : 2.22802124541038

- Output (minimum value of results) 0.423086066762795 into class 2

- Update Weight -> $w_j(new) = w_j(old) + \alpha[x - w_j(old)]$

w2 new	-0.01547 5025	0.21879 75875	0.02485 96	-0.02947 565	0.36971	0.3521	0

DATA 5→ ‖x-wj‖

CLASS 1 : 2.01115702659564

CLASS 2 : 0.588796889639928

CLASS 3 : 2.07679796080408

- Output (minimum value of results) 0.588796889639928 into class 2

- Update Weight -> $w_j(new) = w_j(old) - \alpha[x - w_j(old)]$

w2 new	-0.02504 877625	0.22973 7466875	0.02610 258	-0.03094 94325	0.38819 55	0.36970 5	0

DATA 6→ ‖x-wj‖

CLASS 1 : 2.00835305402835

CLASS 2 : 1.146980670565

CLASS 3 : 1.69370094467707

- Output (minimum value of results) 1.146980670565 into class 2

- Update Weight -> $w_j(new) = w_j(old) - \alpha[x - w_j(old)]$

w2 new	-0.02630 1215062 5	0.24122 4340218 75	-0.01088 2291	-0.03249 6904125	0.40760 5275	0.38819 025	-0.03266

Every epoch will generate new initial weights that will be counted to count on the next epoch until it reaches MaxEpoch. This process is repeated by the same

calculation method up to MaxEpoch limit. When it has reached MaxEpoch, the final result is as follows :

- Final initiation weight

data	x1	x2	x3	x4	x5	x6	x7	class
1	0.3423752957375	1.7587033276 75	0.7235105137 75	0.6171285813 5	0	0	0	1
2	-0.0294829241 64541	0.2408087795 5712	-0.0158263615 64448	-0.0364312611 10519	0.4156083796 82068	0.3940350939 39955	-0.0395340790 15746	2
3	0.176	1.5316	0.5105	0	0	0	1.3064	3

- Final training result

data	x1	x2	x3	x4	x5	x6	x7	class	result
1	0.5282	0.5105	0.2552	0.6532	0.3521	0.3521	0	1	2
2	0.176	1.2763	0	0	0	0	0	1	1
3	0.176	0	0.7658	0	0.3521	0.3521	0	2	1
4	0	0	0	0	0.7043	0.3521	0	2	2
5	0.176	0	0	0	0	0	0	3	2
6	0	0	0.7658	0	0	0	0.6532	3	2

From the training calculation results using LVQ above, it shows that there are 3 data that does not comply to the original class then with that results above so the accuracy is 50%.

4.1.4.2 LVQ Algorithm Analysis on Testing Data

After the training process is complete, using the final initiation result to do the testing process.

- Final initiation weight

data	x1	x2	x3	x4	x5	x6	x7	class

1	0.3423752957375	1.758703327675	0.723510513775	0.617128581335	0	0	0	1
2	-0.029482924164541	0.24080877955712	-0.015826361564448	-0.036431261110519	0.415608379682068	0.394035093939955	-0.039534079015746	2
3	0.176	1.5316	0.5105	0	0	0	1.3064	3

After that calculate the testing data with Square of Euclidean Distance formula with the final initiation weight.

testing data	x1	x2	x3	x4	x5	x6	x7
	0.176	1.2763	0	0	0.7658	0	0

CLASS 1 : 1.32331345456988

CLASS 2 : 1.18131442886505

CLASS 3 : 1.61830743062003

The minimum value of results is 1.18131442886505 where the weight class 2. Then it is mean that the testing data is classified in biography category.

4.2 Design

4.2.1 Use Case Diagram

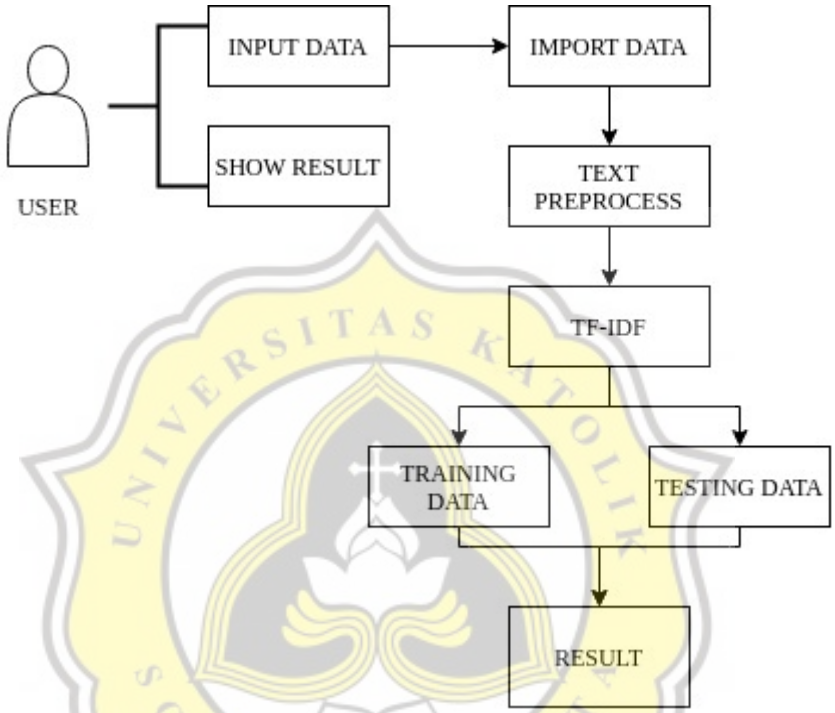

Illustration 4.3: Use Case Diagram

Based on use case diagram above, the first step for user input data and then import it to databse. After it done, the next process is text preprocessing and text mining to get the training data. After it declare the training data and testing data will be classify using LVQ algorithm. The result of this project will show the classification of book description.

4.2.2 Flow Chart

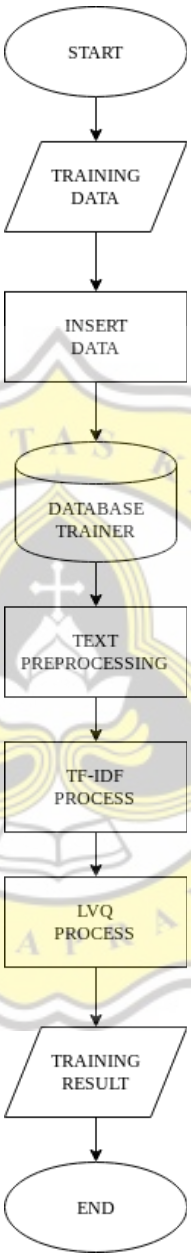

Illustration 4.4:
Training Data Flow Chart

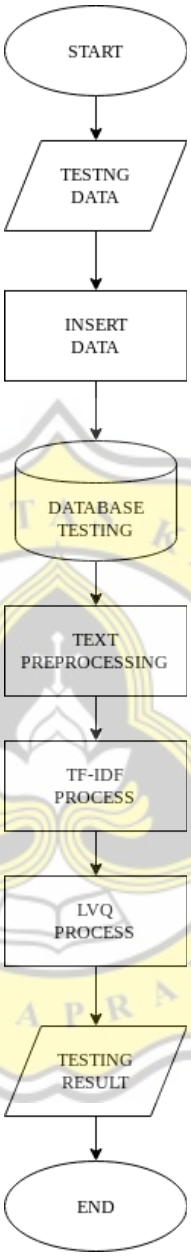

Illustration 4.5:
Testing Data Flow Chart

From Illustration 4.4 and 4.5, both training and testing data have exactly same process. After the data imported into database, the data will go through Text Preprocessing. And then TF-IDF process to weighting data terms and intiation weight on training data. After Text Preprocessing and TF-IDF process already

completed, LVQ process will be executed. Flow chart of LVQ process can be seen like illustration below.

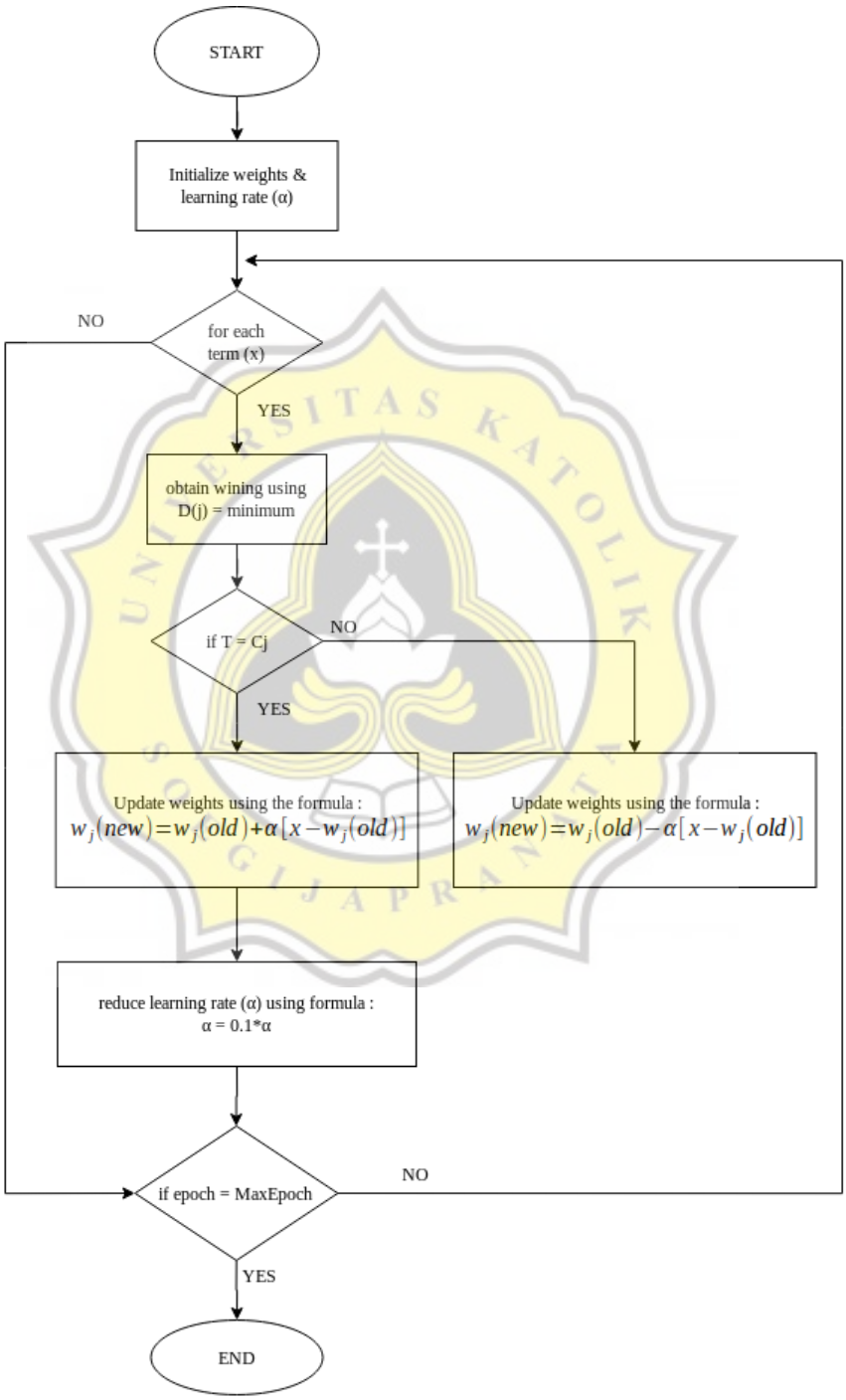

Illustration 4.6: LVQ Process Flow Chart

Like Illustration 4.6, each term will be calculated and finding the minimum value for the winning class. If the winning class equals with the former class, update the weights with $w_j(new)=w_j(old)+\alpha[x-w_j(old)]$ formula but if it isn't equals with the former class then update the weights with $w_j(new)=w_j(old)-\alpha[x-w_j(old)]$ formula. At every epoch the learning rate (α) will be reduced and after the process meet the MaxEpoch, the process is end.

4.2.3 UML Class Diagram

This classes are casefolding process for training and testing data.

Illustration 4.7: Case Folding Training Class Diagram

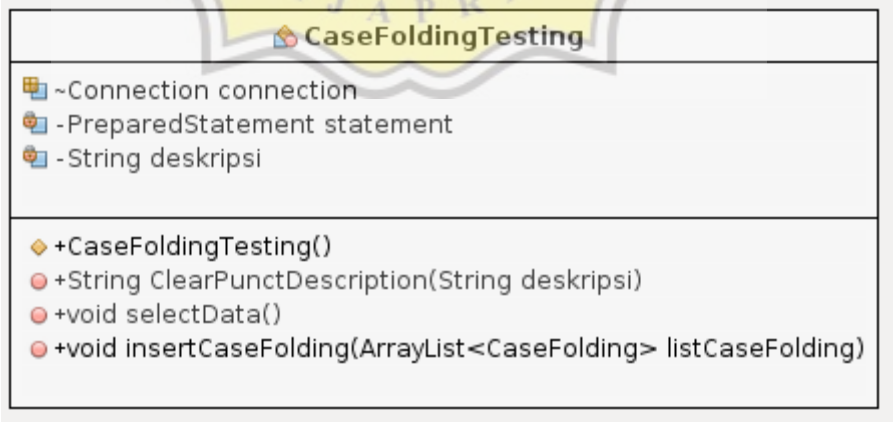

Illustration 4.8: Case Folding Testing Class Diagram

This classes are tokenizing process for training and testing data.

Illustration 4.9: Tokenize Training Flow Class Diagram

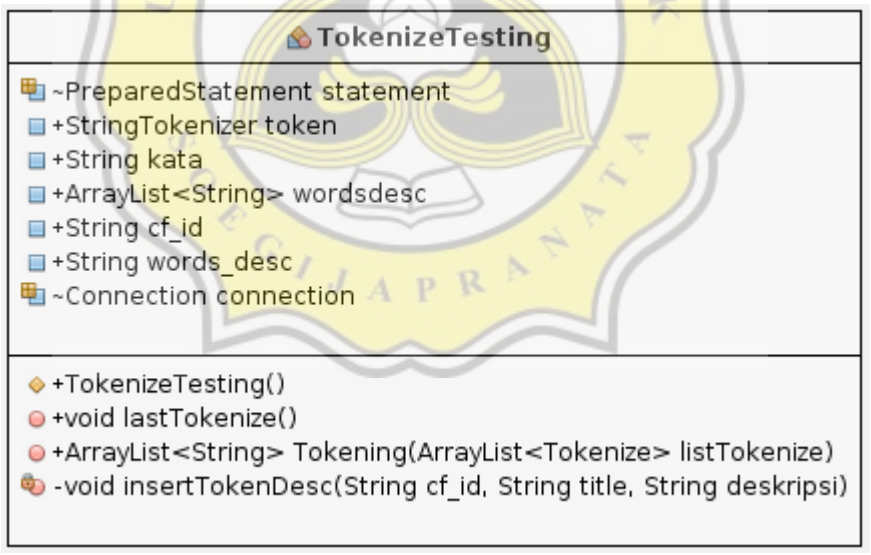

Illustration 4.10: Tokenize Testing Flow Class Diagram

This classes are stopword process for training and testing data.

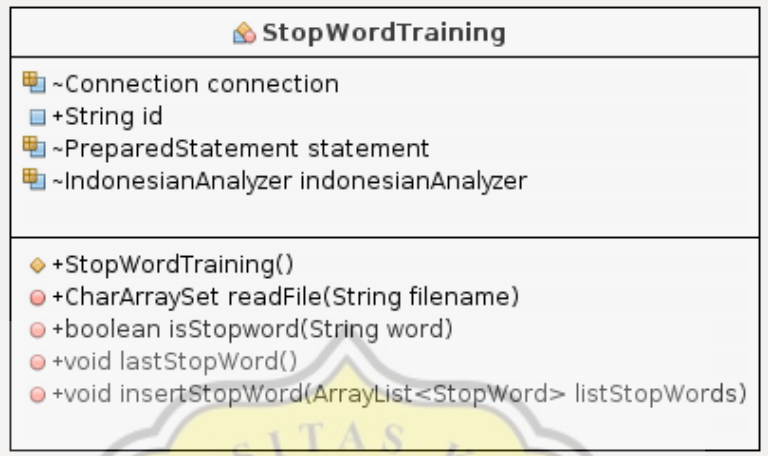

Illustration 4.11: Stop Word Training Class Diagram

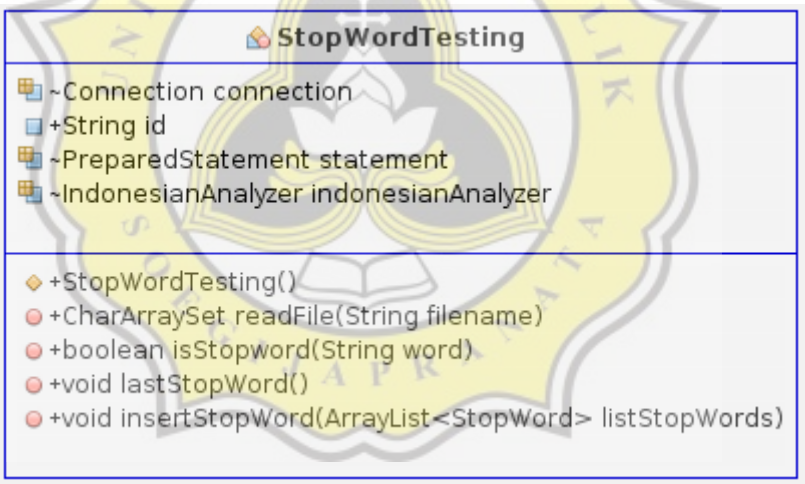

Illustration 4.12: Stop Word Testing Class Diagram

This classes are stemming process for training and testing data.

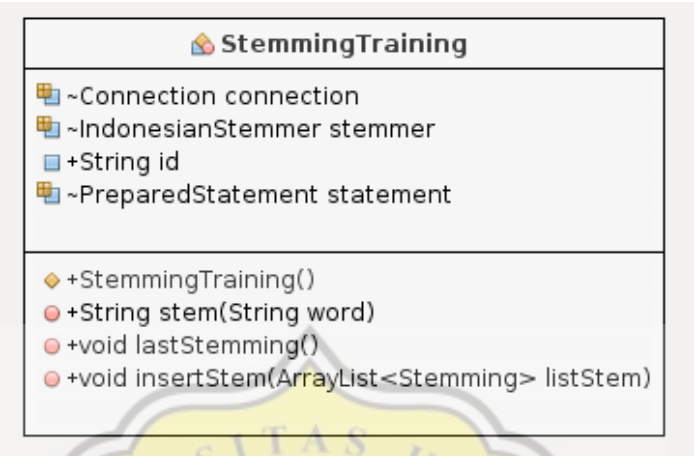

Illustration 4.13: Stemming Training Class Diagram

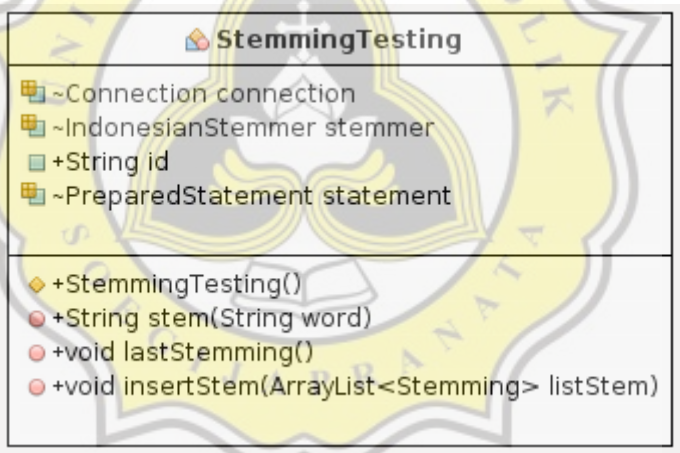

Illustration 4.14: Stemming Testing Class Diagram

This class function is for calculate TF-IDF values and reference initial weights for LVQ algorithm based on training data.

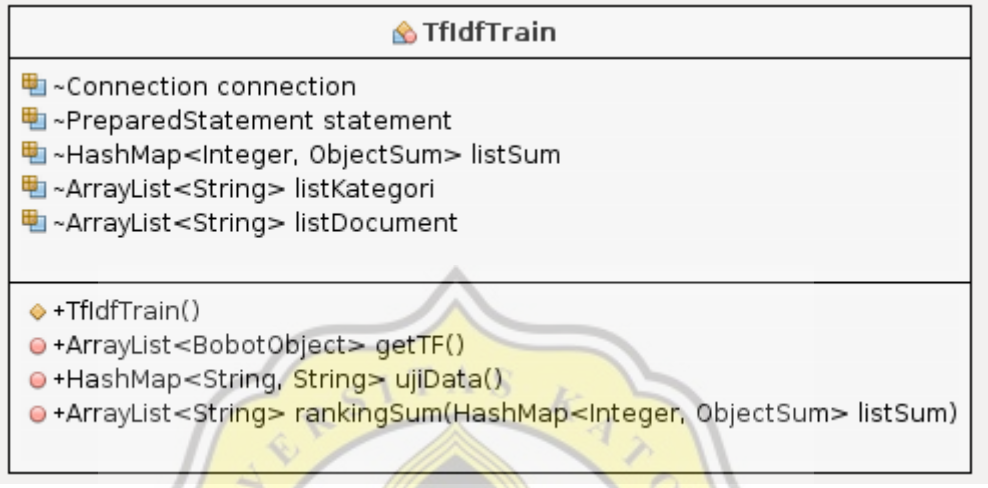

Illustration 4.15: TF-IDF Training Class Diagram

Classification class is to implement LVQ algorithm to all training data and how the accuracy of the algorithm based on training data.

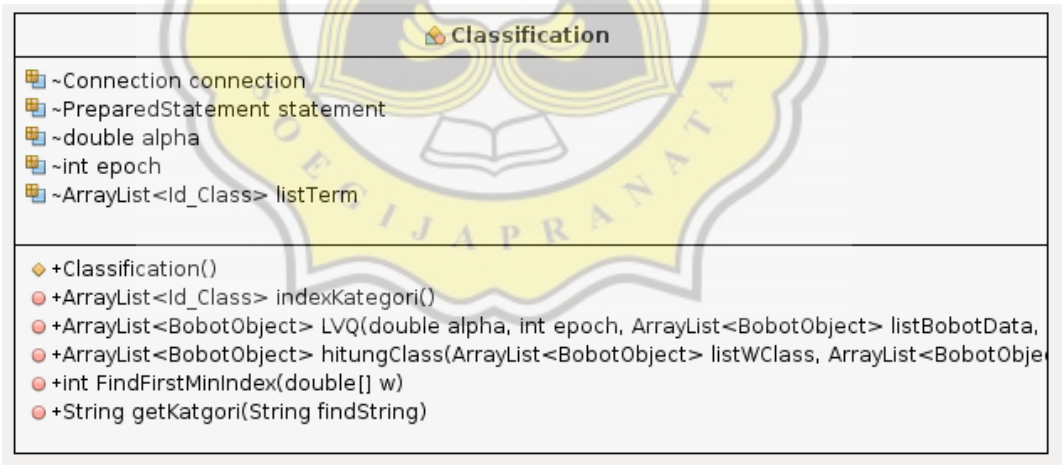

Illustration 4.16: Classification Class Diagram

TfIdfTesting class is to calculate TF-IDF values of testing data and classify the book categories.

Illustration 4.17: TfIdfTesting Class Diagram

4.2.4 Database Schema

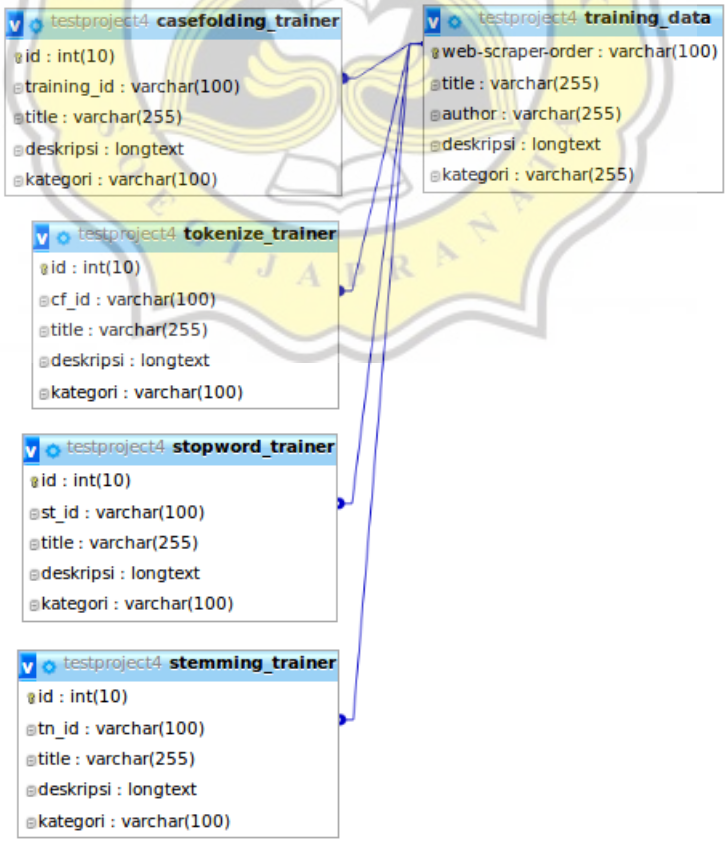

Illustration 4.18: Training Data Tables in the Database

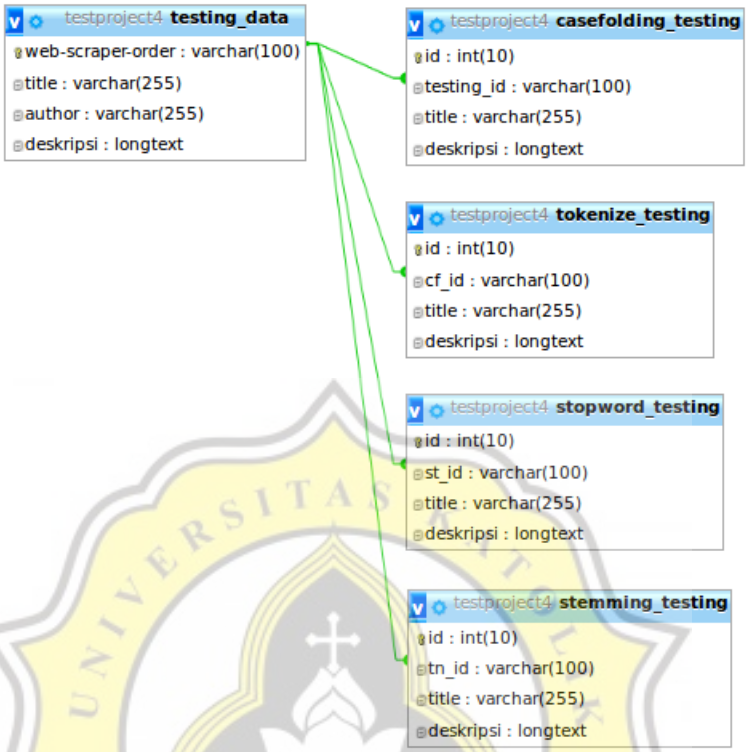

Illustration 4.19: Testing Data Tables in the Database