CHAPTER 5
IMPLEMENTATION AND RESULTS

5.1. Implementation

The first step in this program is to retrieve the data first. The data retrieval process must have an API Key. What you will get is the consumer key, consumer secret, access token, and access secret. After that we can use it for crawling data.

```
1. CONSUMER_KEY = 'MPc7Gg7lpJ70qwaCADwobAPx5'
2. CONSUMER_SECRET= 'mlwP0RjYVbKaWpBGBnjYlbpQSPQJ1SP7k7Ng8eTO2JQ61tDFwS'
3. ACCESS_TOKEN = '706895602404052992-6yFFdD4QSR8xbwY9DJccK36TJzNXx99'
4. ACCESS_TOKEN_SECRET = '3ZTsgtSwqF1AnulJfc2nXgtfOTmJudgzyKIXzEBWTV06e'
```

Now we can input keyword that become our concern in line 1. Spesification of the data is message, favorite count, retweet count, created user, username, followers count. The last procces is export it to csv

```
1. tweets = api.search(q="pindah ibu kota OR pinda ibu kota OR
   #pindahibukota")
2. message,favorite_count,retweet_count,created_at,user_name,followers_c
   ount = [],[],[],[],[],[]
```

In this section Before calculating using the algorithm, the data must be processed by going through. The following is case folding process that functions to convert all letters to lowercase. In addition to characters a to z will be omitted and will remove numbers and punctuation that have nothing to do with analysis. Also remove emoticon, URL, and mention.

```
1. clean = BeautifulSoup(tweet.Message[i], 'lxml')
2. clean = re.sub(r'@[A-Za-z0-9_]+','', tweet.cleanHTML[i])
3. clean = re.sub('https?://[A-Za-z0-9./]+','', tweet.cleanMention[i])
4. clean = re.sub(r"\\x(.){2}", "", tweet.cleanURL[i])
5. clean          =          re.sub(r"^b[\'\"]|#[A-Za-z0-9]+|RT|\\n|
   +|:\(|:\)|:v|:V|:'\)|:'\(", " ", tweet.cleanUnicode[i])
6. clean = re.sub(r"\d+", " ", tweet.cleanOther[i])
7. clean = tweet.cleanNum[i].lower()
8. clean = re.sub(r" +", " ", tweet.result[i])
```

after cleaning is done the data is ready to next process, it is Tokenization. Tokenization is the process of separating text into parts of words so that they can be analyzed based on Lexicon to determine sentiment. In this process, Tokenization,

Filtering stopword, and Stemming will combined its result in one table. Before run this code, declare the process first.

```
1. tweet['coba_token']                                          =
   tweet['hasil'].apply(nltk.word_tokenize)
2. tweet['coba_token']                                          =
   tweet['coba_token'].apply(filteringText)
3. tweet['coba_token']                                          =
   tweet['coba_token'].apply(stemmingText)
4. tweet.coba_token
5. tweet.head()
```

The result in coba_token now is ready to determine its sentiment using lexicon based method. In line 1 dan 7 we add positive and negative lexicon and decided that tha value of positive is >0 in line 23, value of negative is <0 in line 26, and if none of them are fulfilled its mean neutral.

```
1. lexicon_positive = dict()
2. import csv
3. with open('lexicon_positive.csv', 'r') as csvfile:
4. reader = csv.reader(csvfile, delimiter=',')
5. for row in reader:
6. lexicon_positive[row[0]] = int(row[1])


7. lexicon_negative = dict()
8. import csv
9. with open('lexicon_negative.csv', 'r') as csvfile:
10. reader = csv.reader(csvfile, delimiter=',')
11. for row in reader:
12. lexicon_negative[row[0]] = int(row[1])

13. def sentiment_analysis_lexicon_indonesia(text):
14. #for word in text:
15. score = 0
16. for word in text:
17. if (word in lexicon_positive):
18. score = score + lexicon_positive[word]
19. for word in text:
20. if (word in lexicon_negative):
21. score = score + lexicon_negative[word]
22. polarity=''
23. if (score > 0):
24. polarity = 'positive'
25. elif (score < 0):
26. polarity = 'negative'
27. else:
28. polarity = 'neutral'
29. return score, polarity
```

now we can use it to determine sentimen to our dataset, add the dataset first and count polarity score in coloumn coba_token in line 1. After that we show the result by adding 2 more columm to store polarity_score and polarity. This procces is in line 4

and 5. After we got all of them, export again new data in CSV. New data is contain hasil,coba_token, polarity_score, and polarity.

```
1. results =
   preprocessing['coba_token'].apply(sentiment_analysis_lexicon_
   indonesia)
2. results = list(zip(*results))
3. data["hasil"] = preprocessing["hasil"]
4. data['polarity_score'] = results[0]
5. data['polarity'] = results[1]
6. print(data['polarity'].value_counts())
7. # Export to csv file
8. data.to_csv(r'sentiment.csv', index = False, header =
   True,index_label=None)
9. data
```

After the labelling process is complete, change the sentiment in polarity column to number. Positive stand for 1, negative stand for 0, and then neutral stand for 2. This process running in line 2.

```
1. data = pd.read_csv('BigSentiment.csv',encoding='utf-
   8',error_bad_lines=False)
2. data.polarity.replace(['negative','positive',
   'neutral'],[0,1,2],inplace=True)
3. data
```

the next step is split the data. This is the process of dividing the data into 2 namely test data and train data. The data divide 30% for data test, and the rest of it is data training.

```
1. data_train, data_test = train_test_split(data,
   test_size=0.30)
2. data_train['hasil']=data_train['hasil'].values.astype('U')
3. data_test['hasil']=data_test['hasil'].values.astype('U')
```

If the data split process is complete, the next step is the TF-IDF process. This algorithm will helps transform text into a meaningful representation of numbers which is used to fit machine algorithm for prediction. Next process is started using SVM algorithm to test accuracy of this prediction based on vectorized and the sentiment from lexicon.

```
1. classifier_linear = LinearSVC(verbose=1)
2. t0 = time.time()
3. history = classifier_linear.fit(train_vectors,
   data_train["polarity"])
4. t1 = time.time()
5. prediction_linear = classifier_linear.predict(test_vectors)
6. t2 = time.time()
7. time_linear_train = t1-t0
```

```
8. time_linear_predict = t2-t1
9. # results
10.      print("Training time: %fs; Prediction time: %fs" %
   (time_linear_train, time_linear_predict))
11.      report = classification_report(data_test["polarity"],
   prediction_linear, output_dict=True)
12.      print('positive: ', report['1'])
13.      print('negative: ', report['0'])
14.      print('neutral: ', report['2'])
15.      y_train_hat=classifier_linear.predict(train_vectors)
16.      y_test_hat=classifier_linear.predict(test_vectors)
```

To test again this data, author using K-Folding Cross Validation to validate the distribution. When folds get cross contaminated like this, models get a misleading boost in performance. What we want is for the cross validation metrics to tell us how the model will generalize with unseen data.

```
1. train = kfoldproperty.iloc[split1[0]]
2. test = kfoldproperty.iloc[split1[1]]


3. print("Train -----\nAnalysis =", Id, "Counts:")
4. print(train['polarity'][train['polarity']                    ==
   Id].value_counts(sort=False))
5. display(train['polarity'][(train['polarity'] == Id)])


6. print("Test -----\nAnalysis =", Id, "Counts:")
7. print(test['polarity'][test['polarity']                      ==
   Id].value_counts(sort=False))
8. display(test['polarity'][(test['polarity'] == Id)])
```

After that is getting the accuracy of K-fold Validitaion and the result is about 88%.

```
1. from sklearn.model_selection import cross_validate
2. pipeline.set_params(lr__C=gs.best_params_['lr__C'])
3. print("Running stratified k-fold...", end='')
4. skf_results = cross_validate(
5. pipeline,
6. X=kfoldproperty['hasil'],
7. y=kfoldproperty['polarity'],
8. cv=skf,
9. return_train_score=False,
10.      verbose=False)
11.      print(" done.")
12.      print("Stratified    k-fold    average    accuracy:",
   np.mean(skf_results['test_score'])* 100)
```

5.2. Results

In this study, it showed that in 3000 data of Twitter, 1674 users gave positive response to the topic of the new capital city of Indonesia. But there are 1005 of the people who gave a negative response. Rest of them, 321 gave a neutral response. In this study, before classifying the data, the thing that must be done is to retrieve data using the crawling method. Here are some data results that have been successfully retrieved from Twitter.

	Unnamed: 0	Message	Favorite Count	Retweet Count	Created At	Username	Followers
0	0	Jokowi Utamakan Pemerataan dukung pemindahan I...	0	0	2022-06-12 02:19:15	clarke	848
1	1	ASN hingga TNI Pindah ke IKN Nggak Perlu Beli ...	0	0	2022-06-12 02:18:03	Ronnie Adi	5
2	2	Perencanaan Pembangunan IKN\nJokowi Utamakan P...	0	0	2022-06-12 02:17:38	Baim Saputra	1
3	3	Pembangunan IKN Menggerakkan Banyak Sektor Eko...	0	0	2022-06-12 02:17:36	Baim Saputra	1
4	4	IKN Nusantara Diyakini Bakal Gerakkan Banyak S...	0	0	2022-06-12 02:17:33	Baim Saputra	1
...	...	...	...	...	...	...	...
2995	2995	@MarukPiter Barisan sakit hati yg tidak puas d...	0	0	2022-06-10 03:04:51	Andra Gonsales	1
2996	2996	Pembangunan IKN Dipastikan Dorong Ekonomi dan ...	0	0	2022-06-10 03:04:37	Hendro Kartiko	233
2997	2997	PLN Pasok Listrik ke Titik Nol untuk Pembangun...	0	0	2022-06-10 03:04:28	Hendro Kartiko	233
2998	2998	Dukungan terus mengalir untuk pemindahan IKN N...	1	0	2022-06-10 03:04:23	Sophia Wahyu Ningrum	26
2999	2999	@kopitiam_ong Pembangunan IKN Nusantara sudah ...	0	0	2022-06-10 03:04:16	Rifta Ridwan	1

3000 rows × 7 columns

Figure 5.1 Crawling Result

Total data is about 3000 and then preprocessing can be done. Detail of it is remove Unicode, URL, emoticon or symbol, mention, and lowercase the result. After that the result stored in column hasil. It is the clean tweet, and ready to tokenize, filtering stopword, and stemming. 3 process that mention before stored the result in new column named coba_token.

	Message	cleanHTML	cleanMention	cleanURL	cleanUnicode	cleanOther	cleanNum	result	hasil	coba_token
0	Jokowi Utamakan Pemerataan dukung pemindahan I...	Jokowi Utamakan Pemerataan dukung pemindahan I...	Jokowi Utamakan Pemerataan dukung pemindahan I...	Jokowi Utamakan Pemerataan dukung pemindahan IKN	Jokowi Utamakan Pemerataan dukung pemindahan IKN	Jokowi Utamakan Pemerataan dukung pemindahan IKN	Jokowi Utamakan Pemerataan dukung pemindahan IKN	jokowi utamakan pemerataan dukung pemindahan ikn	jokowi utamakan pemerataan dukung pemindahan ikn	[jokowi, utama, perata, dukung, pindah, ikn]
1	ASN hingga TNI Pindah ke IKN Nggak Perlu Beli ...	ASN hingga TNI Pindah ke IKN Nggak Perlu Beli ...	ASN hingga TNI Pindah ke IKN Nggak Perlu Beli ...	ASN hingga TNI Pindah ke IKN Nggak Perlu Beli ...	ASN hingga TNI Pindah ke IKN Nggak Perlu Beli ...	ASN hingga TNI Pindah ke IKN Nggak Perlu Beli ...	ASN hingga TNI Pindah ke IKN Nggak Perlu Beli ...	asn hingga tni pindah ke ikn nggak perlu beli ...	asn hingga tni pindah ke ikn nggak perlu beli ...	[asn, tni, pindah, ikn, nggak, beli, rumah, se...
2	Perencanaan Pembangunan IKN\nJokowi Utamakan P...	Perencanaan Pembangunan IKN\nJokowi Utamakan P...	Perencanaan Pembangunan IKN\nJokowi Utamakan P...	Perencanaan Pembangunan IKN\nJokowi Utamakan P...	Perencanaan Pembangunan IKN\nJokowi Utamakan P...	Perencanaan Pembangunan IKN Jokowi Utamakan Pe...	Perencanaan Pembangunan IKN Jokowi Utamakan Pe...	perencanaan pembangunan ikn jokowi utamakan pe...	perencanaan pembangunan ikn jokowi utamakan pe...	[rencana, bangun, ikn, jokowi, utama, perata]
3	Pembangunan IKN Menggerakkan Banyak Sektor Eko...	Pembangunan IKN Menggerakkan Banyak Sektor Eko...	Pembangunan IKN Menggerakkan Banyak Sektor Eko...	Pembangunan IKN Menggerakkan Banyak Sektor Eko...	Pembangunan IKN Menggerakkan Banyak Sektor Eko...	Pembangunan IKN Menggerakkan Banyak Sektor Eko...	Pembangunan IKN Menggerakkan Banyak Sektor Eko...	pembangunan ikn menggerakkan banyak sektor eko...	pembangunan ikn menggerakkan banyak sektor eko...	[bangun, ikn, gerak, sektor, ekonomi, jokowi, ...
4	IKN Nusantara Diyakini Bakal Gerakkan Banyak S...	IKN Nusantara Diyakini Bakal Gerakkan Banyak S...	IKN Nusantara Diyakini Bakal Gerakkan Banyak S...	IKN Nusantara Diyakini Bakal Gerakkan Banyak S...	IKN Nusantara Diyakini Bakal Gerakkan Banyak S...	IKN Nusantara Diyakini Bakal Gerakkan Banyak S...	IKN Nusantara Diyakini Bakal Gerakkan Banyak S...	ikn nusantara diyakini bakal gerakkan banyak s...	ikn nusantara diyakini bakal gerakkan banyak s...	[ikn, nusantara, yakin, gerak, sektor, ekonomi...

Figure 5.2 Preprocessing Result

The data in coba_token is the result of all steps in preprocessing. Its mean this data is ready to determine sentiment steps using lexicon based method. This lexicon is in Indonesian. The detail is lexicon work with token and match them so in the end you will get polarity_score. It is the sum of all word in coba_token but in range point -5 until 5 based on lexicon. To get the sentiment. We declare if polarity score <0 its mean negative, and if >0 it is positive, and 0 mean neutral. In this section total data is 3000 divided by positive sentiment 1674, negative sentiment 1005, and neutral sentiment 321.

	hasil	coba_token	polarity_score	polarity
0	jokowi utamakan pemerataan dukung pemindahan ikn	[jokowi, utama, perata, dukung, pindah, ikn]	4	positive
1	asn hingga tni pindah ke ikn nggak perlu beli ...	[asn, tni, pindah, ikn, nggak, beli, rumah, se...	-1	negative
2	perencanaan pembangunan ikn jokowi utamakan pe...	[rencana, bangun, ikn, jokowi, utama, perata]	0	neutral
3	pembangunan ikn menggerakkan banyak sektor eko...	[bangun, ikn, gerak, sektor, ekonomi, jokowi, ...	2	positive
4	ikn nusantara diyakini bakal gerakkan banyak s...	[ikn, nusantara, yakin, gerak, sektor, ekonomi...	5	positive
...	...	...	...	...
2995	barisan sakit hati yg tidak puas dgn pembangu...	[baris, sakit, hati, yg, puas, dgn, bangun, ik...	0	neutral
2996	pembangunan ikn dipastikan dorong ekonomi dan ...	[bangun, ikn, dorong, ekonomi, libat, masyarakat]	0	neutral
2997	pln pasok listrik ke titik nol untuk pembangun...	[pln, pasok, listrik, titik, nol, bangun, ikn,...	-4	negative
2998	dukungan terus mengalir untuk pemindahan ikn n...	[dukung, alir, pindah, ikn, nusantara, infrast...	4	positive
2999	pembangunan ikn nusantara sudah final infrast...	[bangun, ikn, nusantara, final, infrastruktur,...	3	positive

3000 rows × 4 columns

Figure 5.3 Sentiment Determination Result

Performance evaluation of Accuracy, Precision and Recall from experiments that have been done, the final result of the test has an accuracy of 87%.

```
Accuracy score is 87%.

Classification Report:
              precision    recall  f1-score   support

           0       0.83      0.89      0.86       330
           1       0.90      0.91      0.91       486
           2       0.78      0.50      0.61        84

    accuracy                           0.87       900
   macro avg       0.84      0.77      0.79       900
weighted avg       0.86      0.87      0.86       900
```

Figure 5.4 Classification Report

Entering the classification process, this process is carried out to test the accuracy of the method Lexicon Based in determining the sentiment of an opinion tweet. In the data classification process, it is tested using the 5-fold cross validation method. So the dataset will be divided into two, namely 5 parts with 4/5 parts used for the training process and 1/5 of the part is used for the testing process. Iteration takes place 5 times with variations of training and testing data using a combination of 5 sections data. The Acuraccy result after that is about 88,23%

```
Train -----
Analysis = positive Counts:
positive    1339
Name: polarity, dtype: int64

0       positive
3       positive
4       positive
11      positive
12      positive
          ...
2988    positive
2990    positive
2991    positive
2993    positive
2999    positive
Name: polarity, Length: 1339, dtype: object
```

```
Test -----
Analysis = positive Counts:
positive    335
Name: polarity, dtype: int64

5       positive
6       positive
48      positive
65      positive
68      positive
          ...
2954    positive
2969    positive
2985    positive
2986    positive
2998    positive
Name: polarity, Length: 335, dtype: object
```

```
Running stratified k-fold... done.
Stratified k-fold average accuracy: 88.23333333333332
```

Figure 5.5 K-Folding Cross Validation Result