# Analisis dan Implementasi Algoritma *Graph-based K-Nearest Neighbour* untuk Klasifikasi *Spam* pada Pesan Singkat

**Gde Surya Pramartha**

Program Studi Sarjana Teknik Informatika Fakultas Informatika
Telkom University
Bandung, Indonesia
suryapramartha@gmail.com

**Abstrak :** Pesan singkat atau Short Message Service (SMS) adalah salah satu layanan komunikasi yang sangat populer pada *mobile phone* saat ini karena kemudahan penggunaan, sederhana, cepat, dan murah. Meningkatnya penggunaan *mobile phone* ini dimanfaatkan oleh banyak pihak untuk mendapatkan keuntungan, salah satunya adalah mengirimkan *spam* melalui SMS. *Spam* biasanya berisikan iklan dari suatu produk, promosi, atau *malware* yang sangat mengganggu pengguna *mobile phone.* Oleh sebab itu, dalam tugas akhir ini dibuatlah SMS *spam* filter untuk menyaring SMS yang menggunakan algoritma Graph-*based K- Nearest Neighbour* (GKNN). SMS yang didapatkan terlebih dahulu di *preprocessing* kemudian data akan direpresentasikan ke dalam model graf berbobot dan berarah. Pengujian algoritma dilakukan dengan menggunakan skenario pembagian data 5-fold dan 10-fold dan didapatkan hasil dengan rata-rata akurasi mencapai 99,06% untuk 5-fold dan 99,13% untuk 10-fold.

Kata Kunci : *spam, spam filtering, preprocessing,* klasifikasi, *k-nearest neighbour, graph-based k-nearest neighbour*

## I. Pendahuluan

Pesan singkat atau *Short Message Service* (SMS) adalah salah satu layanan komunikasi yang sangat populer pada *mobile phone* saat ini. SMS digunakan oleh jutaan pengguna mobile phone setiap harinya karena kemudahan penggunaan, sederhana, cepat, dan murah [1]. Generasi terbaru dari *mobile phone,* yang biasanya disebut *smart phone,* menggunakan SMS tidak hanya sebagai alat berkomunikasi, namun juga digunakan untuk berbagai kebutuhan seperti otentifikasi *mobile-banking,*dan sinkronisasi media sosial (Facebook, Twitter) [1]. Meningkatnya penggunaan *smart phone* saat ini dimanfaatkan oleh banyak pihak untuk mendapatkan keuntungan, salah satunya adalah mengirimkan *spam* melalui SMS.

*Spam*, didefinisikan oleh The Spam Track pada Text Retrieval Conference (TREC) adalah suatu konten yang tidak diminta dan tidak dikehendaki yang dikirim sembarang, baik langsung maupun tidak langsung oleh pengirim yang tidak memiliki hubungan dengan penerima [2]. *Spam* SMS adalah *Spam* yang dikirim melalui SMS. *Spam* biasanya berisikan iklan dari suatu produk, promosi, atau *malware*. Bagi kebanyakan orang, hal ini sangatlah mengganggu sehingga membuat penanganan terhadap *spam* pada SMS sangat penting untuk dilakukan. Dari kondisi tersebut, maka pada tugas akhir ini, dibuatlah suatu cara untuk melakukan penyaringan pada SMS dan melakukan klasifikasi SMS ke dalam kategori *spam* dan bukan *spam* (ham).

Klasifikasi adalah suatu teknik pada data *mining*, dimana terdapat set atau kumpulan data yang sudah memiliki label dan digunakan untuk membuat model untuk menentukan label dari data yang besar [3]. Beberapa algoritma yang dapat digunakan untuk melakukan klasifikasi adalah pohon keputusan ( Decision tree), Bayesian Classifier, SVM (Support Vector Machine), Neural Network, dan KNN (k-Nearest Neighbour) [4]. Kumpulan data yang sudah memiliki label disebut data *training*. Pada kasus ini, data *training* yang digunakan berupa kumpulan SMS yang cenderung bebas dan tidak teratur, sehingga sebelum dilakukan proses klasifikasi, data SMS yang didapatkan terlebih dahulu di *preprocessing* untuk membersihkan, menyederhanakan dan membuat data lebih teratur. *Preprocessing* yang dilakukan berupa penanganan singkatan, menghilangkan tanda baca, *stemming*, tokenisasi dan *feature selection.*

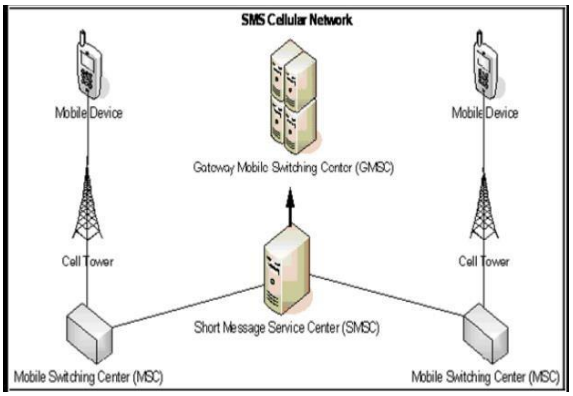
Dari penelitian yang pernah dilakukan tentang klasifikasi, didapatkan hasil bahwa algoritma *Graph-based K-Nearest Neighbour*(GKNN) dapat melakukan klasifikasi SMS dengan tingkat akurasi sebesar 98,9% [5]. Berdasarkan penelitian tersebut, maka dipilihlah GKNN sebagai algoritma yang digunakan untuk melakukan proses klasifikasi pada SMS *filter* di tugas akhir ini.

Pada Jurnal ini selanjutnya akan membahas : pada bagian II, akan membahas tentang tinjauan pustaka , bagian III membahas tentang gambaran umum sistem, bagian IV akan membahas tentang hasil pengujian dan bagian V berisi kesimpulan.

## II. Tinjauan Pustaka

### *Short Message Service* (SMS)

Pesan singkat atau Short Message Service (SMS) adalah salah satu layanan komunikasi yang sangat populer pada *mobile phone*. SMS digunakan oleh jutaan pengguna *mobile phone* setiap harinya karena kemudahan penggunaan, sederhana, cepat, dan murah [1]. Servis dari SMS diatur oleh pusat distribusi pesan yang disebut SMSC ( Short Message Service Center) yang bertanggungjawab dalam mengirimkan pesan ke perangkat pengguna. Gambaran dari sistem dasar SMS adalah sebagai berikut :




Gambar 1. Sistem dasar SMS

Komponen lainnya yaitu MSC (Mobile Switching Center), bertanggungjawab pada proses *routing call* dan pesan. Call Tower atau BTS (Base Station) bertugas untuk mengambil pengiriman pesan ke perangkat *mobile*. Sedangkan GMSC ( Gateway Mobile Switching Center) bertanggungjawab dalam mengambil pengiriman pesan antar jaringan atau antar operator *mobile* [9].

### *Spam Filtering*

*Spam* adalah suatu konten yang tidak diminta dan tidak dikehendaki yang dikirim sembarang, baik langsung maupun tidak langsung oleh pengirim yang tidak memiliki hubungan dengan penerima [2]. *Spam* biasanya berisikan iklan dari suatu produk, promosi, atau *malware* yang bersifat merugikan bagi penerimanya.

*Spam filter* adalah suatu teknik yang secara otomatis mendeteksi *spam* dengan tujuan melakukan penyaringan terhadap pesan yang masuk. Terdapat beberapa teknik yang dapat diterapkan

untuk mengurangi *spam* pada *mobile*. *Conten-based filtering* adalah teknik yang paling banyak digunakan untuk melakukan *filtering spam* [10]. Beberapa algoritma yang termasuk dalam teknik *Content-based filtering* adalah Bayesian classifier, SVM (Support Vector Machine), KNN (k-Nearest Neighbour), dan Neural Network. Teknik ini melakukan *filtering* berdasarkan isi dari suatu pesan kemudian akan di klasifikasikan ke dalam kelas *spam* atau bukan *spam.*

Model Representasi Data Berbasis Graf

Graf adalah kumpulan (set) titik-titik atau node yang memiliki bobot yang dihubungkan dengan garis atau edge. Pada model representasi data berbasis graf , dimisalkan G adalah suatu graf, yang memiliki 3 tupel: G= (V,E,FWM), dimana V adalah kumpulan node, E adalah kumpulan edge yang terhubung ke node. FWM (Feature Weight Matrix) adalah bobot dari edge yang menghubungkan node.

-Node: setiap node pada graf merepresentasikan sebuah token dari hasil feature selection yang dilakukan sebelumnya. Token ini bersifat unik di setiap kelompoknya.

-Edge: edge pada graf dibentuk berdasarkan kemunculan 2 buah token pada kelompok graf secara berurutan.

-Feature Weight Matrix(FWM) : misalkan terdapat 2 buah feature/token yaitu i dan j, maka bobot dari FWM(i,j) atau W(i,j) adalah frekuensi kemunculan urutan 2 buah feature tersebut dalam kelompok graf

Untuk meningkatkan pengukuran dengan representasi data berbasis graf, dihitung juga frekuensi kemunculan dari suatu feature pada kelompok graf yaitu W(i,i) sehingga pada graf menyimpan informasi berupa frekuensi dari suatu feature dan frekuensi kemunculan 2 buah feature secara berurutan [12].

**Klasifikasi**

Klasifikasi adalah *supervised learning method* dengan tujuan untuk dapat menentukan kelas dari suatu objek ke dalam kategori yang telah didefinisikan sebelumnya. Beberapa algoritma yang dapat digunakan untuk melakukan klasifikasi adalah pohon keputusan ( Decision tree), Bayesian Classifier, SVM (Support Vector Machine), Neural Network, dan KNN (k-Nearest Neighbour) [11].

Proses klasifikasi biasanya terdiri dari 2 fase yaitu *learning* dan *test*. Pada fase *learning* , sebagian data yang telah diketahui kelasnya digunakan untuk membentuk model prediksi. Pada fase *test*, dilakukan pengujian terhadap model yang dibentuk dengan data lainnya untuk mengetahui akurasi dari model tersebut.

***Graph-based K-Nearest Neighbour* (GKNN)**

*Graph-based K-Nearest Neighbour* (GKNN) merupakan pengembangan dari *K-Nearest Neighbour* dimana pada GKNN data training terlebih dahulu di representasikan ke dalam bentuk model graf. Tujuannya untuk mempercepat waktu dalam menghitung similarity antara dokumen yang diklasifikasikan dengan jumlah sampel dokumen yang besar. Dengan representasi ke model data graf , maka dapat mempercepat perhitungan tanpa mengurangi ukuran sampel data.

Suatu graf terdiri dari node, edge dan bobot dari edge, untuk mengukur similarity antara 2 graf, maka dilakukan klasifikasi untuk mengukur similarity-nya ($g_{i,}cg_{i)}$. Feature weight (FW) mendefinisikan similarity antara 2 buah graf berdasarkan bobot dari node dan edge yang terdapat pada kedua graf. Perhitungan FW :

```
Input:
    Testing graph g_i , training graph cg_i
Output:
    Fw
Procedure:
    1.  For each edge in g_i
    2.      If edge in cg_i
    3.       If(w_ij(g_i)>=w_ij(cg_i) //w_ij is the weight of edge
    4.            If(j>i)
    5.                Fw+=αw_ij(cg_i)
    6.            Else if(j=i)
    7.                Fw+=w_ij(cg_i)
    8.            End if
    9.       Else If(w_ij(g_i)<w_ij(cg_i)
    10.           If(j>i)
    11.             Fw+=αw_ij(g_i)
    12.           Else if(j=i)
    13.             Fw+=w_ij(g_i)
    14.           End if
    15.      End if
    16.     End if
    17.  End for
```

Gambar 2. Pseudo code menghitung Feature weight [12]

*Nft(Node fit percent)* menunjukkan berapa banyak node pada train graf dengan bobot>0 juga ada pada test graf. *Nft* dapat didefinisikan sebagai berikut :

$$Nfp = \frac{|\{node \mid node \in g_i \wedge node \in cg_i\}|}{|Number\ of\ Feature\ terms|} \quad (1)$$

Pertama-tama hitung nilai *Nft* dari test kata dan train kata dengan menghitung frekuensi dari setiap feature pada graf yaitu nilai W(i,i) Jika nilai *Nft* lebih besar daripada *threshold*, maka akan dihitung nilai FW dari 2 buah graf. Sebaliknya, jika nilai *Nft* lebih kecil daripada *threshold*, maka 2 graf yang dihitung bukan berada dalam satu kategori, sehingga tidak perlu menghitung nilai FW. Dengan cara ini, kompleksitas dari perhitungan similarity dapat dipercepat. Perhitungan GKNN dapat dilihat pada pseudo code berikut ini :

```
Input:
      Testing set graphs G={g_1,g_2,…g_i, …g_n} , value K
      Training set graphs CG={cg_1,cg_2,…cg_i, …cg_n}
Output :
      Result set   R={r_1,r_2,…r_i, …r_n }
      r_i is the categorization result of text d_i
Procedure:
1 For each g_i  in G
2.  Initial List RL to store Fw and text category (length is K)
3    For each cg_i in CG
4.       If Nfp(g_i,cg_i)>α
5.            Calculate Feature weight Fw(g_i,cg_i)
6             If RL is not full
7                 Add Fw(g_i,cg_i) and category of cg_i to RL
8             Else If RL is full
9                 If Fw(g_i,cg_i)>min(Fw_i in RL)
10                    Replace Fw_i in RL with Fw(g_i,cg_i)
11                End if
12            End if
13       End if
14  . End For
15   the category of g_i is the category appears most in RL
16   add the category of g_i  to the Result Set R.
16  End For
```

Gambar 3. Pseudo code klasifikasi dengan GKNN [12]

## Pengukuran Performansi

Pengukuran performansi dilakukan untuk mengetahui kemampuan dari sistem yang dibangun. Untuk dapat menentukan kebenaran dari hasil yang didapatkan, maka akan dibandingkan hasil dari sistem dengan data aslinya.

a. Accuracy merupakan ukuran yang digunakan untuk mengetahui seberapa besar kebenaran yang didapatkan, dari keseluruhan data.

b. Precision merupakan ukuran yang digunakan untuk mengetahui seberapa besar hasil yang telah terpilih itu benar

| Sistem | Data | |
|---|---|---|
| | Spam | Ham |
| Spam | True Positif | False Positif |
| Ham | False Negatif | True Negatif |

c. Recall merupakan ukuran yang digunakan untuk mengetahui

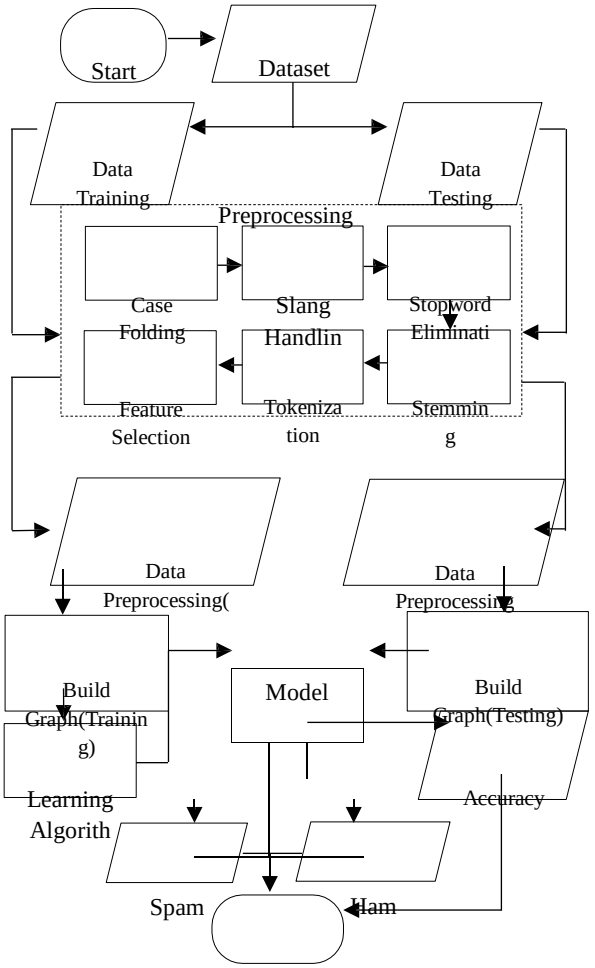
seberapa besar yang benar itu terpilih.

| Sistem | Data | |
|---|---|---|
| | Spam | Ham |
| Spam | True Positif | False Positif |
| Ham | False Negatif | True Negatif |

Dari ketiga pengukuran tersebut, sistem dianggap sangat baik ketika persentasi hasil dari ketiganya dapat mencapai nilai terbaik atau mendekati setidaknya 99%.

## III. Gambaran Umum Sistem

Secara umum sistem yang akan dibangun pada tugas akhir ini adalah sistem untuk melakukan *filtering* terhadap pesan yang masuk dan melakukan pengklasifikasian ke dalam 2 kategori, yaitu *spam* dan bukan *spam*. Algoritma yang digunakan untuk proses klasifikasi adalah *Graph-based k-Nearest Neighbour* (GKNN) dimana data training akan di *preprocessing* terlebih dahulu, kemudian dilakukan pembangunan model graf yang merepresentasikan kata-kata pada tahap *preprocessing*. Setelah itu, graf tersebut diproses dengan algoritma GKNN dan pesan pada data testing dikelompokkan kedalam kategori *spam* atau bukan *spam* berdasarkan data *training* yang sudah dikategorikan. Gambaran umum dari sistem dapat dilihat sebagai berikut:



Finish

Gambar 4.Gambaran umum sistem

Dari gambaran umum sistem di atas, terdapat beberapa proses yaitu :

1. Preprocessing

   Pada proses ini, Data training dan Data testing yang berupa SMS yang belum bersih (teratur,konsisten) akan di proses sehingga menghasilkan data yang lebih teratur dan konsisten. Proses yang dilakukan adalah :

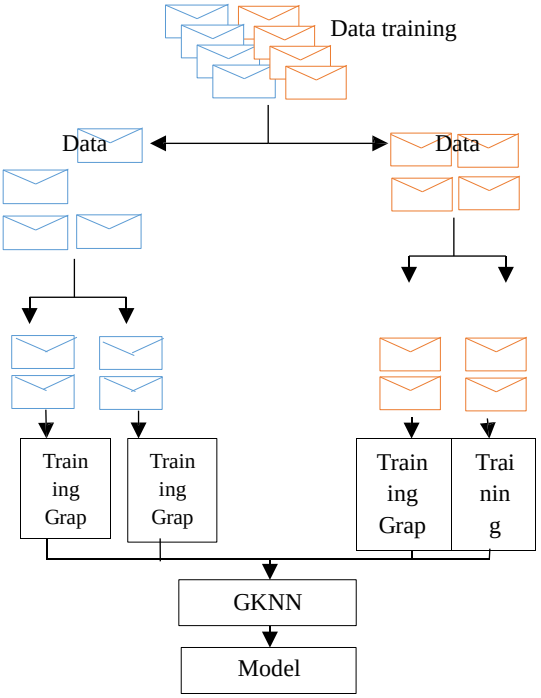
a. Case Folding
Merubah kata-kata pada data menjadi huruf kecil dan menghilangkan semua karakter selain huruf dan angka.
b. Slang Handling
Mengatasi isi SMS yang berisi kata singkatan ke dalam arti aslinya
c. Stopword Elimination

Menghilangkan kata-kata yang dianggap umum yang tidak terlalu berpengaruh terhadap kualitas data.
d. Stemming
Merubah kembali kata-kata yang mengalami penambahan imbuhan atau perubahan bentuk kata.
e. Tokenization
Memecah kalimat pada SMS menjadi kata berdasarkan spasi untuk memudahkan proses selanjutnya

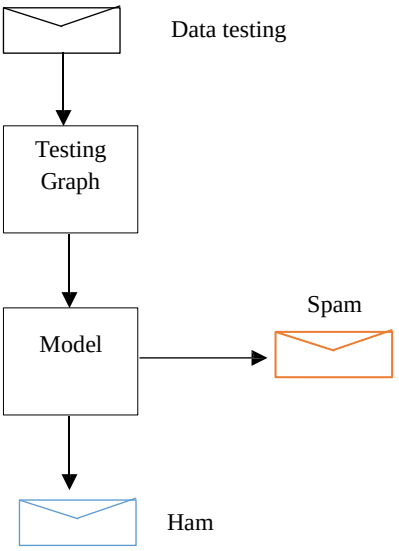
selanjutnya.
f. Feature Selection
Suatu metode yang digunakan untuk menghilangkan kata-kata yang tidak memiliki arti atau *noisy feature*, sehingga dapat menyederhanakan perhitungan dan meningkatkan akurasi.

2. Build Graph

Setelah dilakukan tahap *preprocessing*, maka dihasilkan data yang sudah siap digunakan yaitu Data preprocessing.Data ini kemudian digunakan sebagai input untuk representasi data berbasis graf. Setiap node pada graf menunjukkan sebuah token yang dipilih pada saat tahap *preprocessing*. Token ini berupa sebuah kata unik dalam suatu kelompok graf. Edge dibentuk berdasarkan urutan kemunculan antara 2 buah kata. Feature Weight Matrix digunakan untuk meunjukkan bobot dari edge .




Gambar 5. Skema pembangunan model dari training graph




Gambar 6. Skema pengklasifikasian data testing dari testing graph

3. Learning Algorithm

Pada proses ini dilakukan impelemtasi dari algoritma GKNN, hasil graf yang sudah dibangun sebelumnya akan dijadikan model untuk melakukan klasifikasi.

4. Model

Pada proses ini, data testing yang sudah dibentuk menjadi graf akan diproses dengan model klasifikasi. Hasil dari proses ini adalah menentukan kelas dari setiap SMS pada data test yaitu *spam* atau ham (bukan *spam*).Dari hasil penentuan kelas tersebut, akan didapatkan pula akurasi dari model klasifikasi.

## IV. Pengujian

Untuk mendapatkan data training dan data testing untuk pengujian pada tugas akhir ini, digunakan pembagian menggunakan teknik k-fold *cross validation* dengan membagi data menjadi k bagian, dan mengkombinasikan hasil pembagian sebagai data training dan data testing. Nilai k yang umum digunakan adalah 5 dan 10, pembagian ini dilakukan untuk melihat pengaruh jumlah data terhadap hasil pengujian.

Dari data set yang telah didapatkan yaitu *SMSSpamCollection* [8] dengan rincian total SMS sebanyak 5574 dengan SMS spam berjumlah 747 dan SMS ham berjumlah 4827, SMS spam yang dipakai untuk pengujian berjumlah 700 dan SMS ham berjumlah 4000.Kemudian dari data tersebut, dibagi menjadi data training dan data testing sesuai penggunaan 5-fold dan 10-fold dengan rincian sebagai berikut :

Tabel 1. Pembagian Data 5-fold

| Kelompok Sampel Data | SMS ke- | | Pengujian | |
|---|---|---|---|---|
| | Ham | Spam | Data Testing | Data Training |
| A | 1-800 | 1-140 | A | BCDE |
| B | 801-1600 | 141-280 | B | ACDE |
| C | 1601-2400 | 281-420 | C | ABDE |
| D | 2401-3200 | 421-560 | D | ABCE |
| E | 3201-4000 | 561-700 | E | ABCD |

Tabel 2. Pembagian Data 10-fold

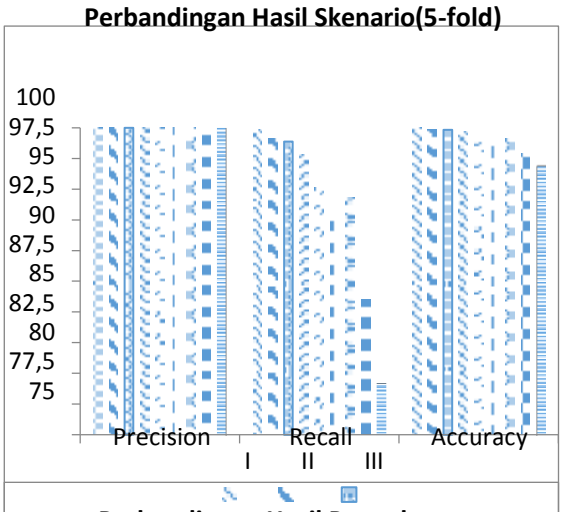
| Kelompok Sampel Data | SMS ke- | | Pengujian | |
|---|---|---|---|---|
| | Ham | Spam | Data Testing | Data Training |
| A | 1-400 | 1-70 | A | BCDEFGHIJ |
| B | 401-800 | 71-140 | B | ACDEFGHIJ |
| C | 801-1200 | 141-210 | C | ABDEFGHIJ |
| D | 1201-1600 | 211-280 | D | ABCEFGHIJ |

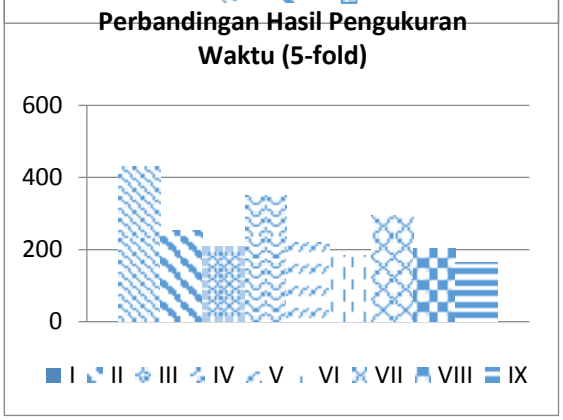
| | | | | |
|---|---|---|---|---|
| E | 1601-2000 | 281-350 | E | ABCDFGHIJ |
| F | 2001-2400 | 351-420 | F | ABCDEGHIJ |
| G | 2401-2800 | 421-490 | G | ABCDEFHIJ |
| H | 2801-3200 | 491-560 | H | ABCDEFGIJ |
| I | 3201-3600 | 561-630 | I | ABCDEFGHJ |
| J | 3601-4000 | 631-700 | J | ABCDEFGHI |

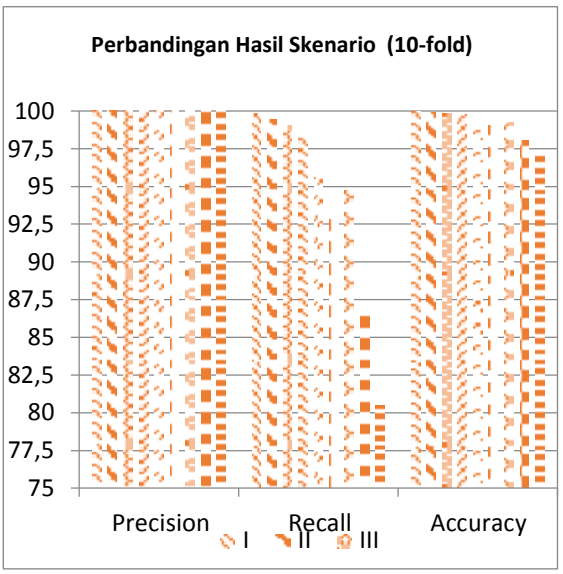
Skenario untuk pengujian pada tugas akhir ini adalah menggunakan pembagian data 5-fold dan 10-fold dan mengkombinasikannya dengan nilai *threshold* dan Jumlah SMS per Graf untuk melihat pengaruh keduanya. Nilai *threshold* yang digunakan adalah 0.01, 0.025, dan 0.04 . Sedangkan untuk Jumlah SMS per Graf digunakan nilai 5,8, dan 10 sehingga terdapat 9 kombinasi nilai yang digunakan dalam skenario. untuk pangjang list yang digunakan dalam diklasifikasi ditentukan oleh nilai K. Nilai K yang digunakan dalam pengujian adalah 5.

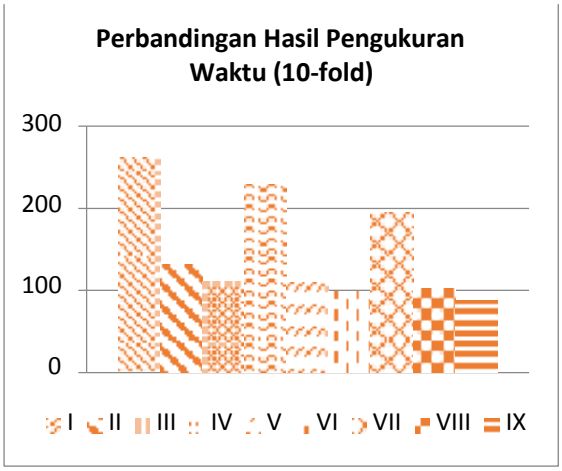
Tabel 3. Nilai Skenario Threshold dan Jumlah SMS per Training Graf

| Threshold | Jumlah SMS per Training Graf | | |
|---|---|---|---|
| | 5 | 8 | 10 |
| 0.01 | I | II | III |
| 0.025 | IV | V | VI |
| 0.04 | VII | VIII | IX |






Dari hasil pengujian yang dilakukan, maka dapat dikatakan bahwa

1. Untuk Precision, seluruh skenario mendapatkan hasil yang sangat baik yaitu 100%, yang berarti bahwa sistem mendeteksi spam dengan sangat baik.
2. Untuk Recall, hasil terbaik didapatkan dengan melakukan skenario I yaitu 99,86% baik pada Pembagian data 5-fold maupun pada pembagian data 10-fold. Hasil terburuk didapatkan dengan skenario IX yaitu 79,29% pada pembagian data 5-fold dan 80,43% pada pembagian data 10-fold.
3. Untuk Accuracy, rata-rata akurasi pada pembagian data 5-fold mencapai 99,06% dan pada pembagian data 10-fold mencapai 99,16%. Itu berarti sistem sangat baik dalam menentukan SMS spam dan ham.
4. Untuk waktu, skenario I baik pada pembagian data 5-fold maupun 10-fold membutuhkan waktu pemrosesan lebih lama dibandingkan skenario lainnya. Sedangkan skenario IX pada pembagian data 5-fold dan 10-fold membutuhkan waktu pemrosesan tercepat dibandingkan skenario lainnya.
5. Untuk Jumlah data training yang digunakan, pada pembagian data 5-fold hampir semua hasil akurasi di setiap skenario yang dilakukan memiliki nilai lebih kecil dibandingkan dengan hasil akurasi jika menggunakan pembagian data 10-fold.

Dari hasil yang didapatkan,diketahui bahwa algoritma GKNN memang efektif untuk digunakan dalam SMS spam filtering dimana untuk semua skenario, nilai akurasi yang didapatkan mencapai 99,06% untuk pembagian data 5-fold dan 99,16% untuk pembagian data 10-fold.

## V. Kesimpulan

Berdasarkan pengujian serta analisis yang telah dilakukan pada tugas akhir ini, dapat diambil beberapa kesimpulan, yaitu :

1. Penerapan algoritma GKNN dilakukan dengan membentuk model klasifikasi yang terdiri dari kumpulan Training Graf yang merupakan representasi dari data training yang sudah melalui tahap *preprocessing.*
2. Proses klasifikasi SMS dilakukan dengan menguji model klasfikasi yang telah dibuat dengan data testing yang direpresentasikan ke dalam graf yang membentuk sebuah Testing Graf dan dilakukan pencarian nilai *Feature Weight* (FW) untuk menentukan tingkat kesamaan antara Training Graf dan Testing Graf. Semakin tinggi nilai FW, maka semakin tinggi kemungkinan kedua graf tersebut berada pada satu kategori, yaitu *spam* atau ham.
3. Nilai *threshold* dan Jumlah SMS per Training Graf mempengaruhi hasil akurasi dan waktu

pemrosesan. Semakin kecil nilai *threshold* dan Jumlah SMS per training Graf yang digunakan, maka semakin tinggi nilai akurasi yang didapatkan, namun membutuhkan waktu pemrosesan yang semakin lama. Sebaliknya, semakin besar nilai *threshold* dan jumlah SMS per training graf yang digunakan, maka semakin rendah nilai akurasi yang didapatkan, namun membutuhkan waktu pemrosesan semakin sedikit.

4. Jumlah data training yang digunakan berpengaruh terhadap hasil yang didapatkan. Semakin banyak data training yang digunakan, nilai akurasi yang dihasilkan semakin tinggi.
5. Rata-rata nilai akurasi yang diperoleh dari pengujian terhadap algoritma GKNN mencapai 99,06% untuk pengujian dengan pembagian data 5-fold dan 99,16% untuk pengujian dengan pembagian data 10-fold, yang artinya SMS spam filter yang dibuat sangat baik dalam melakukan klasifikasi

Saran yang dapat diberikan untuk pengembangan atau penelitian selanjutnya adalah :

Penerapan dapat dilakukan untuk mendeteksi SMS bukan hanya SMS berbahasa inggris saja, tetapi dapat dilakukan pada bahasa selain bahasa inggris.

## Daftar Pustaka


[1] M. Z. Rafique and M. Abulaish, "Graph-Based Learning Model for Detection of SMS Spam on Smart Phones".

[2] G. V. Cormack and D. R. Cheriton, "Email Spam Filtering: A systematic Review," pp. 355-455, 2006.

[3] B. M. Ramageri, "DATA MINING TECHNIQUES AND APPLICATIONS," *Indian Journal of Computer Science and Engineering,* vol. 1, pp. 301-305.

[4] T. P. Ho, H.-S. Kang and S.-R. Kim, "Graph-Based KNN Algorithm for Spam SMS Detection," *Universal Computer Science,* 2013.

[5] "Slang Dictionary - Text Slang & Internet Slang Words," NoSlang.com, 2005. [Online]. Available: http://www.noslang.com/dictionary/. [Accessed 23 June 2015].

[6] "RANKS NL," [Online]. Available: http://www.ranks.nl/stopwords. [Accessed 23 June 2015].

[7] "Snowball," [Online]. Available: http://snowball.tartarus.org/. [Accessed 23 June 2015].

[8] UCI, "SMS Spam Collection Data Set," [Online]. Available: https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection. [Accessed 23 June 2015].

[9] D. Belem and F. Duarte-Figueiredo, "Content Filtering for SMS System Based on Bayesian Classifier and

Word Grouping," 2011.

[10] J. M. G. Hidalgo, G. C. Bringas and E. P. Sanz, "Content Based SMS Spam Filtering".

[11] S. R. Singh, H. A. Murthy and T. A. Gonsalves, "Feature Selection for Text Classification Based on Gini Coefficient of Inequality," *Workshop and Conference Proceedings 10,* pp. 76-85, 2010.

[12] Z. Wang and Z. Liu, "Graph-based KNN Text Classification," *Seventh International Conference,* 2010.

[13] T. M. Mahmoud and A. M. Mahfouz, "SMS Spam Filtering Technique Based on Artificial Immune System," *IJCSI,* 2012.

[14] P. Kantor, F. Robert, F.-Y. Wang, G. Muresan, D. D. Zeng, H. Chen and R. C. Merkle, Intellegence and Security Informatics, Atlanta: Springer, 2005.

[15] "Collecting SMS Message for a Public Research Group," National University of Singapore, 6 September 2014. [Online]. Available: http://wing.comp.nus.edu.sg:8080/SMSCorpus/history.jsp. [Accessed 2 November 2014].