TUGAS AKHIR – K141502

KLASIFIKASI SENTIMEN MENGGUNAKAN *SEMI-SUPERVISED SUBJECTIVE FEATURE WEIGHTING AND INTELLIGENT MODEL SELECTION* (SWIMS) PADA FORUM DISKUSI *ONLINE*

ADAM WIDI BAGASKARTA
NRP 05111440000089

Dosen Pembimbing
Dr. Agus Zainal Arifin, S.Kom., M.Kom.
Dini Adni Navastara, S.Kom., M.Sc.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]

TUGAS AKHIR – K141502

KLASIFIKASI SENTIMEN MENGGUNAKAN *SEMI-SUPERVISED SUBJECTIVE FEATURE WEIGHTING AND INTELLIGENT MODEL SELECTION* (SWIMS) PADA FORUM DISKUSI *ONLINE*

ADAM WIDI BAGASKARTA
NRP 05111440000089

Dosen Pembimbing
Dr. Agus Zainal Arifin, S.Kom., M.Kom.
Dini Adni Navastara, S.Kom., M.Sc.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]

FINAL PROJECT – K141502

SENTIMENTS CLASSIFICATION USING SEMI-SUPERVISED SUBJECTIVE FEATURE WEIGHTING AND INTELLIGENT MODEL SELECTION (SWIMS) ON ONLINE DISCUSSION FORUM

ADAM WIDI BAGASKARTA
NRP 05111440000089

Advisor
Dr. Agus Zainal Arifin, S.Kom., M.Kom.
Dini Adni Navastara, S.Kom., M.Sc.

INFORMATICS DEPARTMENT
Faculty of Information and Communication Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

KLASIFIKASI SENTIMEN MENGGUNAKAN *SEMI-SUPERVISED SUBJECTIVE FEATURE WEIGHTING AND INTELLIGENT MODEL SELECTION* (SWIMS) PADA FORUM DISKUSI *ONLINE*

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Komputasi Cerdas dan Visi
Program Studi S-1 Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:
ADAM WIDI BAGASKARTA
NRP: 05111440000089

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Dr. Agus Zainal Arifin, S.Kom., M.Kom. .....
NIP: 19720809 199512 1 001 (pembimbing 1)

Dini Adni Navastara, S.Kom., M.Sc .....
NIP: 19851017 201504 2 001 (pembimbing 2)

SURABAYA
JULI 2018

[Halaman ini sengaja dikosongkan]

KLASIFIKASI SENTIMEN MENGGUNAKAN *SEMI-SUPERVISED SUBJECTIVE FEATURE WEIGHTING AND INTELLIGENT MODEL SELECTION* (SWIMS) PADA FORUM DISKUSI *ONLINE*

Nama Mahasiswa : Adam Widi Bagaskarta
NRP : 05111440000089
Departemen : Informatika FTIK-ITS
Dosen Pembimbing 1 : Dr. Agus Zainal Arifin, S.Kom., M.Kom.
Dosen Pembimbing 2 : Dini Adni Navastara, S.Kom., M.Sc.

ABSTRAK

Jumlah pengguna internet di seluruh dunia bertambah dengan pesat dari 2 miliar di tahun 2010 sekarang menjadi 3 miliar pengguna seluruh dunia ditahun 2017. Bertambahnya penggunaan akses internet yang mudah, membuat sosial media menjadi sangat populer di kalangan masyarakat. Masyarakat akan cenderung bersosialisasi dengan cara mengekspresikan opini melalui sosial media yang ada seperti facebook, twitter, instagram, dan lain-lain. Dalam tugas akhir ini, akan dilakukan klasifikasi sentimen menggunakan data yang berasal dari twitter. Data tersebut meliputi tweet dari akun XL, Telkomsel, dan IM3. Dengan mengetahui hasil sentimen pada sebuah produk, pihak produsen akan dapat merancang sebuah rencana bisnis untuk meningkatkan pelayanan kepada konsumen. Data tersebut akan di klasfikasi menggunakan metode Semi-supervised Subjective Feature Weighting and Intelligent Model Selection (SWIMS). Untuk hasil klasifikasi data tersebut akan dilakukan perbadingan antara penggunaan jumlah fitur menggunakan POS Tagging, Term Presence (TP), dan gabungan dari kedua fitur tersebut. Dari hasil tersebut didapatkan bahwa penggunaan gabungan antara fitur POS Tagging dan Term Presence (TP) serta menggunakan kernel

linear pada SVM memberikan hasil terbaik yaitu accuracy sebesar 82,5%, *precision 82,7%, recall 83,5%,* dan *f-measure 82,4%.*

Kata kunci: twitter, SWIMS, SVM, cross validation, sentiwordnet.

SENTIMENTS CLASSIFICATION USING SEMI-SUPERVISED SUBJECTIVE FEATURE WEIGHTING AND INTELLIGENT MODEL SELECTION (SWIMS) ON ONLINE DISCUSSION FORUM

Name : Adam Widi Bagaskarta
NRP : 5114 100 089
Department : Informatics FTIK-ITS
Supervisor I : Dr. Agus Zainal Arifin, S.Kom., M.Kom.
Supervisor II : Dini Adni Navastara, S.Kom., M.Sc.

ABSTRACT

The number of internet users worldwide increased rapidly from 2 billion in 2010 now to 3 billion users worldwide in 2017. Increasing use of easy Internet access, making social media become very popular among the public. The community will tend to socialize by way of expressing opinions through existing social media such as facebook, twitter, instagram, and others. In this study, will be classified sentiment using data derived from twitter. The data includes tweets from XL, Telkomsel and IM3 accounts. By knowing the sentiment of a product, the manufacturer will be able to design a business plan to improve service to consumers. The data will be classified using the Semi-supervised Bubjective Feature Weighting and Intelligent Model Selection (SWIMS) method. For the results of the data classification will be made a comparison between the use of the number of features using POS Tagging, Term Presence (TP), and a combination of both features. From these result, it was found that the combined use of both features which is POS Tagging and Term Presence (TP) gave the best result of accuracy of 82.5%, 82.7% precision, 83.5% recall, and f-measure 82.4%.

Keywords: twitter, SWIMS, SVM, cross validation, sentiwordnet.

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT karenaa atas karunia dan rahmat-Nya penulis dapat menyelesaikan tugas akhir yang berjudul:

KLASIFIKASI SENTIMEN MENGGUNAKAN *SEMI-SUPERVISED SUBJECTIVE FEATURE WEIGHTING AND INTELLIGENT MODEL SELECTION* (SWIMS) PADA FORUM DISKUSI *ONLINE*

Melalui lembar ini, penulis ingin menyampaikan ucapan terima kasih dan penghormatan yang sebesar-besarnya kepada:

1. Ayah, Mama, Nuning, Rafi, dan keluarga besar yang selalu memberikan doa serta dukungan kepada penulis untuk menyelesaikan tugas akhir ini.
2. Bapak Dr. Agus Zainal Arifin, S.Kom., M.Sc selaku dosen pembimbing tugas akhir pertama yang telah membimbing dan memberi banyak masukkan dalam pengerjaan tugas akhir ini.
3. Ibu Dini Adni Navastara, S.Kom., M.Sc. selaku dosen pembimbing tugas akhir kedua yang telah memberikan masukkan serta koreksi dalam pengerjaan tugas akhir.
4. Bapak dan Ibu dosen serta seluruh civitas Departemen Informatika yang telah memberikan pelajaran dan pengalaman selama menjadi mahasiswa di Departemen Informatika.
5. Fafa yang selalu mendengar keluh kesah dan segala cerita penulis baik berkaitan dengan tugas akhir maupun tidak.
6. Teman-teman PH Ristek BEM ITS Wahana Juang yang menjadi *support system* penulis dalam menyelesaikan tugas akhir ini.
7. Teman-teman kabinet BEM ITS Wahana Juang yang memberikan dukungan dan inspirasi bagi penulis sebelum hingga pengerjaan tugas akhir ini.

8. Teman-teman lab. MIS Informatika ITS. Ovan, Irfan, Hanif, Ikhsan, Winda, Uly, Anisa, Ariya, Rafi, Purina, Fajar, Ayas, Muhajir, dan Fino yang telah memberikan semangat dan dukungan kepada penulis selama berada di lab.
9. Serta pihak-pihak lain yang namanya tidak dapat penulis sebutkan satu per satu.

Bagaimanapun juga penulis telah berusaha sebaik-baiknya dalam menyelesaikan tugas akhir ini. Namun, penulis mohon maaf apabila terdapat kekurangan ataupun kesalahkan yang penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan untuk ke depannya.

Surabaya, Juli 2018

Adam Widi Bagaskarta

DAFTAR ISI

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

[Halaman ini sengaja dikosongkan]

BAB I
PENDAHULUAN

1.1 Latar Belakang

Berdasarkan sebuah survei, jumlah pengguna internet di seluruh dunia bertambah dengan pesat dari 2 miliar di tahun 2010 sekarang menjadi 3 miliar pengguna seluruh dunia ditahun 2017. Bertambahnya penggunaan akses internet yang mudah, membuat sosial media menjadi sangat populer di kalangan masyarakat. Masyarakat akan cenderung bersosialisasi dengan cara mengekspresikan opini melalui sosial media yang ada seperti *facebook, twitter, instagram*, dan lain-lain. Analisis sentimen menjadi sesuatu yang penting pada kondisi tersebut, di mana akan mempermudah dalam mengetahui sebuah tren yang sedang berkembang di masyarakat melalui media sosial tersebut. Selain untuk mengetahui tren yang ada, analisis sentimen dapat digunakan untuk keperluan bisnis. Contohnya pada bisnis *e-commerce* dimana pendapat seorang konsumen sangat mempengaruhi produk yang ditawarkan. Dengan melakukan analisis sentimen maka setiap pendapat yang diberikan akan dianalisis. Analisis sentimen yang dilakukan secara akurat dan terus-menerus akan memberikan beberapa manfaat diantaranya, dapat meningkatkan pelayanan konsumen, dapat mengembangkan kualitas produk terkair, dapat meningkatkan pendapatan penjualan, dan dapat memberikan strategi bisnis untuk masa mendatang.

Analisis sentimen adalah kombinasi dari *natural language processing* (NLP) dan teknik *data mining* [1], hal tersebut dapat juga disebut sebagai *opinion mining* atau *polarity classification*. Beberapa riset telah melakukan uji coba mengenai *opinion mining*. Riset tersebut meliputi pencarian sebuah tren, prediksi suatu kejadian, maupun klasifikasi sentimen. Klasifikasi sentimen berfokus untuk mengidentifikasi sebuah opini atau pendapat pada sebuah tanggapan. Analisis sentimen dapat dilakukan dengan menggunakan *supervised, semi-supervised*, atau *unsupervised* algoritma pada machine learning dan pendekatan berbasis kamus

[2]. Pada penerapannya penggunaan algoritma *supervised learning* memberikan hasil performa yang bagus, namun pelabelan data secara manual akan memberikan waktu yang lama. Namun di sisi yang lain, penggunaan sentimen berbasis kamus yang tidak memerlukan data *training* sehingga tidak dapat memberikan hasil yang cukup memuaskan. Oleh karena itu untuk mendapatkan hasil yang lebih memuaskan, penggunaan sentimen berbasis *domain corpus* dikembangkan. Beberapa riset pada semi-supervised diajukan oleh Bhaskar dkk [3]. Prosedur yang digunakan menggunakan perhitungan frekuensi emosi dan *WordNetAffect* digunakan untuk merepresentasikan vektor teks. Namun frekuensi emosi yang didapat belum mencerminkan kekuatan emosi tersebut. Selain itu juga terdapat riset pada [4] dimana penggunaan *transfer learning* atau adaptasi domain memberikan hasil yang baik. Namun metode tersebut masih perlu diuji dengan menggunakan pendekatan yang lain. Oleh karena itu diperlukan suatu metode untuk melakukan proses klasifikasi berbasis *corpus* kata dan berbasis kamus pada umumnya yang disebut dengan SWIMS. SWIMS adalah metode yang berfokus untuk menangani masalah ketidaktersediaan data berlabel berbasis corpus dan meningkatkan performa klasifikasi sentimen berbasis kamus kata menggunakan *SentiWordNet* [5].

Oleh karena itu dalam tugas akhir ini akan mengimplementasikan penggunaan metode SWIMS dalam meningkat performa klasifikasi sentimen, dimana *accuracy, precision, recall* dan *f-measure* diharapkan dapat diperoleh dengan hasil yang terbaik. Setelah melakukan klasifikasi tersebut, akan didapatkan sebuah keputusan mengenai sentimen yang ada, dimana dapat memberikan sebuah rekomendasi terkait data yang dipakai. Sehingga strategi bisnis dapat segera disusun untuk meningkatkan mutu sebuah produk atau jasa terkait.

1.2 Rumusan Permasalahan

Rumusan masalah yang diangkat dalam tugas akhir ini adalah sebagai berikut:

1. Bagaimana cara membobotkan kata dari sebuah tweet berdasarkan bobot sentimen yang telah ditentukan?
2. Bagaimana memilih *feature* yang akan digunakan dalam melakukan klasifikasi tweet?
3. Apakah metode SWIMS dapat diimplementasikan untuk *tweet* Berbahasa Indonesia?

1.3 Batasan Permasalahan

Batasan masalah pada tugas akhir ini antara lain:

1. *Tools* yang digunakan adalah python 3
2. Bahasa yang digunakan pada dataset adalah Bahasa Indonesia.
3. Pembobotan kata diambil dari *SentiWordNet* Bahasa Indonesia.
4. Dataset yang digunakan berasal dari tweet akun provider XL, TELKOMSEL, dan IM3.

1.4 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah untuk melakukan klasifikasi sentimen menggunakan *Semi-supervied Subjective Feature Weighting and Intelligent Model Selection* (SWIMS) pada forum diskusi online untuk mendapatkan hasil klasifikasi dengan performa lebih baik.

1.5 Manfaat

Manfaat dari pembuatan tugas akhir ini antara lain:

1. Mempermudah dalam melakukan analisa produk atau jasa terhadap data yang digunakan.
2. Mengetahui cara melakukan klasifikasi sentimen menggunakan metode SWIMS.

1.6 Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan tugas akhir ini adalah sebagai berikut:

1. **Studi literatur**

   Pada studi literatur ini akan dipelajari sejumlah referensi yang relevan terhadap tugas akhir yang akan dikerjakan. Studi literatur ini didapatkan dari buku, internet serta materi-materi kuliah yang berhubungan dengan metode yang akan digunakan. Hal-hal yang akan dipelajari yaitu mengenai data processing pada python 3, *Support Vector Machine* (SVM) dan *Cross Validation*.

2. **Analisis dan desain metode**

   Metode yang akan dikembangkan merupakan penggabungan dari metode *Feature Selection* menggunakan *Min-max-SentiMI* dengan *Intelligent Model Selection (IMS)* menggunakan *classifier* SVM. Desain metode ini dibutuhkan untuk mengetahui bagaimana cara melakukan klasifikasi menggunakan *corpus SentiWordNet* dan meningkatkan hasil klasifikasi.

3. **Implementasi**

   Pengembangan dari metode yang akan dibuat pada tugas akhir ini akan menggunakan tools berupa python 3 menggunakan IDE ananconda 3, dimana IDE tersebut telah menyediakan semua *library* yang digunakan untuk data processing.

4. **Pengujian dan evaluasi**

   Pada tahap ini dilakukan pengujian dari model yang telah dibuat.

   1. Pengujian skenario uji coba jumlah fitur
      a. Fitur *POS Tagging* yang terdiri dari kata sifat, kata kerja, kata benda, dan kata keterangan pada dataset
      b. Fitur *Term Presence (TP)* pada dataset
      c. Fitur gabungan dari *Term Presence (TP)* dan *POS Tagging*

2. Pengujian kernel pada SVM, menggunakan *linear* atau *RBF*.
3. Pengujian pada jumlah *fold* pada metode *Intelligent Model Selection* (IMS).
4. Semua skenario tersebut dilakukan pengujian akurasi, presisi, *recall,* dan *F-measure.*

5. Penyusunan buku Tugas Akhir

Pada tahap ini dilakukan proses dokumentasi dan pembuatan laporan dari seluruh konsep, tinjauan pustaka, metode, implementasi, proses yang telah dilakukan, pengujian, evaluasi dan hasil-hasil yang telah didapatkan selama pengerjaan tugas akhir.

1.7 Sistematika Penulisan

Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri atas beberapa bagian seperti berikut ini:

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan tugas akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan tugas akhir.

Bab II Dasar Teori

Bab ini menjelaskan beberapa teori yang dijadikan penunjang dan berhubungan dengan pokok pembahasan yang mendasari pembuatan tugas akhir.

Bab III Analisis dan Perancangan Sistem

Bab ini membahas mengenai perancangan sistem yang akan dibangun. Perancangan sistem meliputi perancangan data dan alur proses dari sistem itu sendiri.

Bab IV Implementasi

Bab ini berisi implementasi dari perancangan sistem yang telah ditentukan sebelumnya.

Bab V Pengujian dan Evaluasi

Bab ini membahas pengujian dari metode yang ditawarkan dalam tugas akhir untuk mengetahui kesesuaian metode dengan data yang ada.

Bab VI Kesimpulan dan Saran

Bab ini berisi kesimpulan dari hasil pengujian yang telah dilakukan. Bab ini juga membahas saran-saran untuk pengembangan sistem lebih lanjut.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan tugas akhir.

Lampiran

Merupakan bab tambahan yang berisi data atau daftar istilah yang penting pada tugas akhir ini.

BAB II
DASAR TEORI

Bab ini membahas teori-teori yang menjadi dasar pembuatan tugas akhir.

2.1 *Text Processing*

Text processing disebut juga sebagai *Text Mining* adalah sebuah pemrosesan teks untuk menghasilkan informasi atau *insights* pada suata data. Untuk menghasilkan beberapa informasi *Text Mining* menggunakan beberapa metode salah satunya NLP (*Natural Laguage Processing)*. Terdapat beberapa tugas dalam NLP diantaranya:

2.1.1 *Stopwords*

merupakan *corpus* kata dalam bahasa tertentu yang sering muncul dan tidak memiliki arti yang signifikan. Sehingga dalam setiap percobaan selalu diabaikan atau dihapus.

2.2.2 *Tokenization*

Tokenization adalah memisahkan kata, simbol, frase, dan entitas penting lainnya (yang disebut sebagai token) dari sebuah teks untuk kemudian di analisa lebih lanjut. Token dalam NLP sering dimaknai dengan "sebuah kata", walau tokenisasi juga bisa dilakukan ke kalimat, paragraf, atau entitas penting lainnya.

2.2.3 *Stemming*

Menghasilkan sebuah bentuk kata yang disepakati oleh suatu sistem tanpa mengindahkan konteks kalimat. Syaratnya beberapa kata dengan makna serupa hanya perlu dipetakan secara konsisten ke sebuah kata baku. Banyak digunakan di IR & komputasinya relatif sedikit. Biasanya dilakukan dengan menghilangkan imbuhan (*suffix/prefix*).

2.2 *Semi-supervised Subjective Feature Weighting and Intelligent Model Selection*

2.2.1 *Min-max Normalization*

Merupakan salah satu tahapan dalam klasifikasi sebuah *text* dimana *min-max normalization* digunakan untuk mempermudah perhitungan bobot *term* yang akan dipakai sebagai *feature*. Rumus penggunaan *min-max normalization* dapat didefinisikan pada persamaan 2.1.

$$V'1 = \frac{Vi - minA}{maxA - minA}\ (maxA\_baru - minA\_baru) + minA\_baru \quad \textbf{(2.1)}$$

Di mana minA dan maxA adalah nilai *min* dan *max* dari atribut A. Sedangkan maxA_baru dan minA_baru adalah nilai *min-max normalization minimum* dan *maximum* dari atribut A

2.2.2 *Feature Selection - Point-wise Mutual Information (PMI)*

Feature Selection atau disebut juga sebagai *vaiable selection* atau *attribute selection. Feature selection* berbeda dengan *feature reduction.* Pada *feature selection* akan dipilih mana fitur yang akan digunakan tanpa melakukan perubahan fitur lain [6]. *Point-wise Mutual Information* (PMI) merupakan salah satu *feature-selection* yang berguna untuk mendapatkan jumlah fitur yang sesuai dengan kriteria. Persamaan PMI dapat ditulis sebagai berikut:

$$PMI(t,l) \cong \log 2\ \frac{A\ x\ N}{(A+B)\ x\ (A+C)} \quad \textbf{(2.2)}$$

Di mana pada persamaan tersebut *t* didefinisikan sebagai *feature (term# part-of-speech), l* adalah label *class* dan *N* adalah jumlah total fitur yang dilabeli. *A, B,* dan *C* masing-masing adalah jumlah kemunculan *t* dengan label *l*, jumlah *t* tanpa label *l*, dan jumlah kemunculan label tanpa *t*. Diagam alur penggunaan PMI ditunjukkan oleh Gambar 3.1

2.2.3 *Intelligent Model Selection (IMS)*

Merupakan sebuah *framework* baru yang diacu oleh penulis untuk meningkatkan hasil performa sistem menggunakan *classifier SVM* dan *cross validation* yang diusulkan [5]. *Framework* ini menggunakan *cross validation* berganda untuk mendapatkan nilai akurasi terbaik pada setiap iterasi klasifikasi yang dilakukan. Diagram alir penggunaan IMS ditunjukkan oleh Gambar 3.3.

2.2.4 *SentiWordNet* Bahasa Indonesia

SentiWordNet adalah sumber *lexical (*kamus) untuk *opinion mining*. *SentiWordNet* terdiri dari beberapa *synset (Synonim Set)* yang memiliki dua nilai untuk masing-masing yaitu, positif dan negatif. Setiap nilai *term* dalam *synset* juga diurutkan berdasarkan seberapa sering digunakan dalam pengertian itu [5]. Dari *SentiWordNet* akan didapatkan bobot polaritas kata yang akan digunakan untuk melabeli kata pada tahapan berikutnya yang didapat dari [7]. Pada penggunaan *SentiWordNet* terdapat empat *POSTag* yang digunakan. Oleh karena itu perlu adanya perubahan format *POSTag* dari Bahasa Indonesia ke dalam *SentiWordNet.* Perubahan format dapat dilihat pada table 2.1.

Tabel 2. 1 Perubahan Format pada *SentiWordNet* untuk *POSTag* Bahasa Indonesia

No	***POS Tagging* Indonesia**	***POS Tagging* pada SentiWordNet**
1	n	n
2	k	r
3	v	v
4	adj	a

Keterangan:
n = *noun,* k = kata keterangan, r = adverb, v = *verb,*
adj = adjektif*,* a = *adjective*

2.3 *Part of Speech Tagging (POS Tagging)*

POS *Tagging* merupakan suatu cara untuk mengkategorikan kelas kata, seperti kata benda, kata kerja, kata sifat, dan lain-lain. Dalam Bahasa Indonesia ada beberapa kelas kata yang dipakai antara lain sebagai berikut [8]:

1. Kata benda (*nomina*)
   Merupakan kata-kata yang merujuk pada bentuk suatu benda. Benda dapat bersifat konkret atau abstrak.
2. Kata jerja (*verba)*
   Merupakan jenis kata yang menyakan suatu perbuatan oleh subjek.
3. Kata sifat (*adjektiva)*
   Merupakan kelompok kata yang mampu menjelaskan atau mengubah kata benda atau kata ganti menjadi lebih spesifik. Selain itu, kata sifat mampu menerangkan kuantitas dan kualitas dari kelompok kelas kata benda atau kata ganti.
4. Kata ganti (*pronomia)*
   Merupakan kata yang digunakan untuk menggantikan suatu benda atau sesuatu yang dibendakan.
5. Kata keterangan (*adverbia)*
   Merupakan kata yang memberikan keterangan kepada kata kerja, kata sifat, dan kata bilangan, bahkan mampu memberikan keterangan pada seluruh kalimat.
6. Kata bilangan (*numeralia)*
   Merupakan jenis kata yang menyatakan jumlah, ukuran, dan urutan sesuatu yang dibendakan.
7. Kata tugas
   Merupakan kata yang memiliki arti gramatikal dan tidak memiliki arti leksikal. Dari segi bentuk umumnya, kata-kata tugas sukar mengalamai perubahan bentuk, seperti kata dengan, telah, dan tetapi.

2.4 *Support Vector Machine* (SVM)

Support Vector Machine (SVM) adalah salah satu algoritma *supervised machine learning* yang dapat digunakan baik itu dalam masalah klasifikasi dan regresi. Dalam SVM terdapat istilah *support vector* yang merupakan titik terdekat dengan *hyperplane. Hyperplane* dapat dimisalkan sebuah garis yang memisahkan dan mengklasifikasikan secara linier satu set data [9]. Dalam algoritma ini, dimisalkan menaruh titik dalam ruang n-dimensi (n adalah fitur yang digunakan) dengan nilai setiap fitur pada koordinat tertentu. Kemudian tujuan yang dilakukan adalah mencari *Hyperplane* [10]. Ilustrasi *Hyperplane* ditunjukkan oleh Gambar 2.1. Persamaan 2.3 digunakan sebagai fungsi diskriminasi untuk model diskriminatif SVM [11].

$$g(x) = W^T \emptyset(x_i) + b \quad \textbf{(2.3)}$$

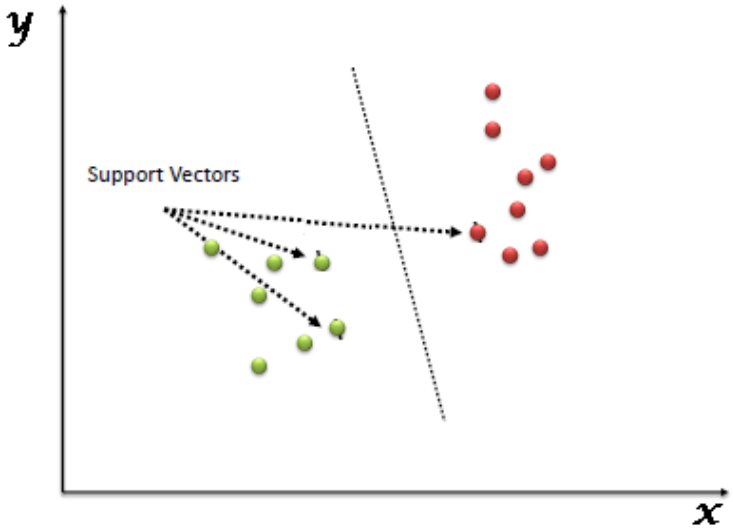
dimana 'w' adalah berat vektor, 'b' adalah bias dan '$\emptyset(x_i)$' adalah ruang input pada dimensi fitur tingi dengan pemetaan *non-linear*. Parameter ini dipelajari otomatis dengan *maximized margin principle* pada *training dataset.*

$$\min \frac{1}{2} W^T W + C \sum_{i=1}^{N} \varepsilon_i \quad \textbf{(2.4)}$$

$$y_i(W^T \emptyset(x_i) + b) \geq 1 - \varepsilon_i \quad \textbf{(2.5)}$$
$$\varepsilon_i \geq 0 \; N = 1,2, \ldots, N$$

ε_i dan C adalah *slack variable* dan koefisien penalti. Dengan diperkenalkannya fungsi *kernel*, fungsi diskriminan didefinisikan sebagai persamaan 2.6.

$$g(x) = \sum_{i=1}^{N} \emptyset(x_i)\, y_i\, K\,(X_i, X) \quad \textbf{(2.6)}$$



Gambar 2. 1 Ilustrasi *hyperplane* pada SVM

2.5 Fungsi *Kernel*

Fungsi *kernel* merupakan fungsi yang dapat mengatasi masalah non-linear yang umumnya terjadi pada dunia nyata. Fungsi ini diaplikasikan pada setiap data untuk memetakan data asli non-linear ke dalam ruang dimensi yang lebih tinggi (*higher-dimensional space*). Terdapat beberapa fungsi *kernel* yang umum digunakan yaitu linear, *polynomial* dan *Radial Basis Function* (RBF) yang ditunjukkan oleh Tabel 2.1.

Tabel 2. 2. Kernel pada *SVM*

Kernel	**Fungsi**
Linear	$k(\boldsymbol{x}_i \,.\, \boldsymbol{x}) = \boldsymbol{x}_i \,.\, \boldsymbol{x}$
Radial Basis Function (RBF)	$k(\boldsymbol{x}_i \,.\, \boldsymbol{x}) = \exp\left(-\gamma \lVert \boldsymbol{x}_i - \boldsymbol{x} \rVert^2\right)$

2.6 *K-Fold Cross Validation*

Dalam *K fold cross validation*, data dibagi menjadi *K* himpunan bagian. Sehingga setiap kali proses ini, salah satu *K* himpunan digunakan sebagai tes atau set validasi dan himpunan *K*-1 lainnya disatukan untuk membentuk satu set *training*. Estimasi kesalahan adalah rata-rata atas semua percobaan *K* untuk mendapatkan efektivitas total dari model. Ini secara signifikan mengurangi bias karena itu menggunakan sebagian besar data untuk pemasangan, dan juga secara signifikan mengurangi varians karena sebagian besar data juga digunakan dalam set validasi. Saling menukar posisi pada pelatihan dan set tes juga menambah keefektifan metode ini. Sebagai aturan umum dan bukti empiris, K = 5 atau 10 umumnya lebih banyak digunakan, tetapi tidak ada yang tetap dan dapat mengambil nilai berapapun [12].

2.7 *Confusion Matrix*

Confusion matrix adalah ringkasan hasil prediksi pada masalah klasifikasi. Jumlah prediksi yang benar dan salah dirangkum dengan nilai dan hitungan setiap masing-masing kelas. Terdapat istilah *true positive, false negative, true negative,* dan *false positive* pada *confusion matrix* yang digunakan untuk menghitung beberapa nilai.

1. *Accuracy*

   Merupakan nilai rasio data yang diklasifikasikan benar dari jumlah total data. Dapat ditulis dengan rumus

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad \textbf{(2.7)}$$

2. *Recall*

   Merupakan nilai rasio dari jumlah total data positif yang diklasifikasikan dengan benar dibagi dengan jumlah total data positif. Dapat ditulis dengan rumus

$$recall = \frac{TP}{TP+FN} \quad \textbf{(2.8)}$$

3. Precision

Merupakan nilai dari total data positif yang diklasifikasikan dengan benar dibagi dengan hasil prediksi data positif. Dapat ditulis dengan rumus

$$precision = \frac{TP}{TP+FP} \quad \textbf{(2.9)}$$

4. F-measure

Merupakan nilai yang diperoleh dari *recall* dan *precision* yang menggunakan *harmonic mean.* Dapat ditulis dengan rumus

$$f - measure = \frac{2*recall*precision}{recall+precision} \quad \textbf{(2.10)}$$

BAB III
ANALISIS DAN PERANCANGAN SISTEM

Bab ini akan menjelaskan tentang analisis dan perancangan sistem untuk mencapai tujuan dari tugas akhir. Perancangan ini meliputi perancangan data dan perancangan proses. Bab ini juga akan menjelaskan tentang analisis implementasi metode secara umum pada sistem.

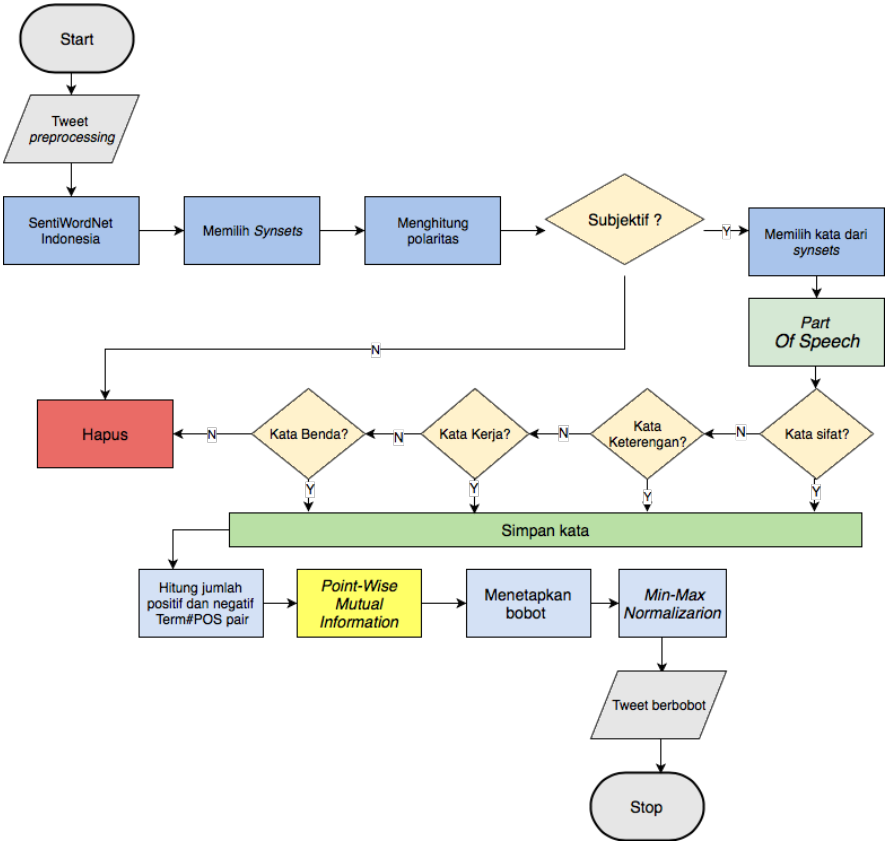
3.1 Analisis Metode Secara Umum

Pada tugas akhir ini akan dibangun suatu sistem untuk melakukan klasifikasi terhadap data tweet pengguna provider XL, Telkomsel, dan IM3 pada bulan April 2016 menggunakan pustaka Python 3. Proses-proses yang dilakukan dalam pengimplementasian sistem ini meliputi tahap *pre-processing* data, tahap pembobotan kata, tahap pemilihan fitur serta normalisasi bobot, dan tahap klasifikasi. Diagram alir dari keseluruhan *framework* SWIMS ditunjukkan oleh Gambar 3.2. Pada diagram alir 3.2, input *dataset* untuk tugas akhir ini berasal dari data *tweet.* Tahap pertama setelah mendapatkan *tweet* tersebut adalah *preprocessing tweet*. Setelah itu melakukan proses pembobotan kata dan normalisasi bobot menggunakan *Point-wise Mutual Information.* Setelah itu pada tugas akhir ini akan diterapkan beberapa fitur. Setelah itu data akan dirubah formatnya ke dalam matriks yang dapat dibaca oleh *classifier*. Setelah itu tahap terakhir merupakan tahapan klasifikasi menggunakan SVM.

Terdapat tiga skenario fitur yang akan dibandingkan pada tugas akhir ini. Perbandingan tiga skenario fitur tersebut berguna untuk mengetahui skenario fitur terbaik yang dapat diimplementasikan untuk melakukan klasifikasi sentimen pada *tweet* provider.

Tahap *pre-processing* merupakan tahap di mana data tweet dari hasil *crawling* akan dilakukan proses pembersihan data. Hasil dari tahap ini adalah *tweet* yang berupa kata-kata dasar yang siap digunakan untuk tahap pembobotan.

Tahap *POS Tagging* dan pembobotan kata dilakukan untuk semua kata tweet yang didapat dari tahap *pre*-processing. Tahapan ini menggunakan kamus SentiWordNet Bahasa. Skenario fitur akan dilakukan pada tahap ini. Diagram alir dari tahap pembobotan kata ditujukkan oleh Gambar 3.1.

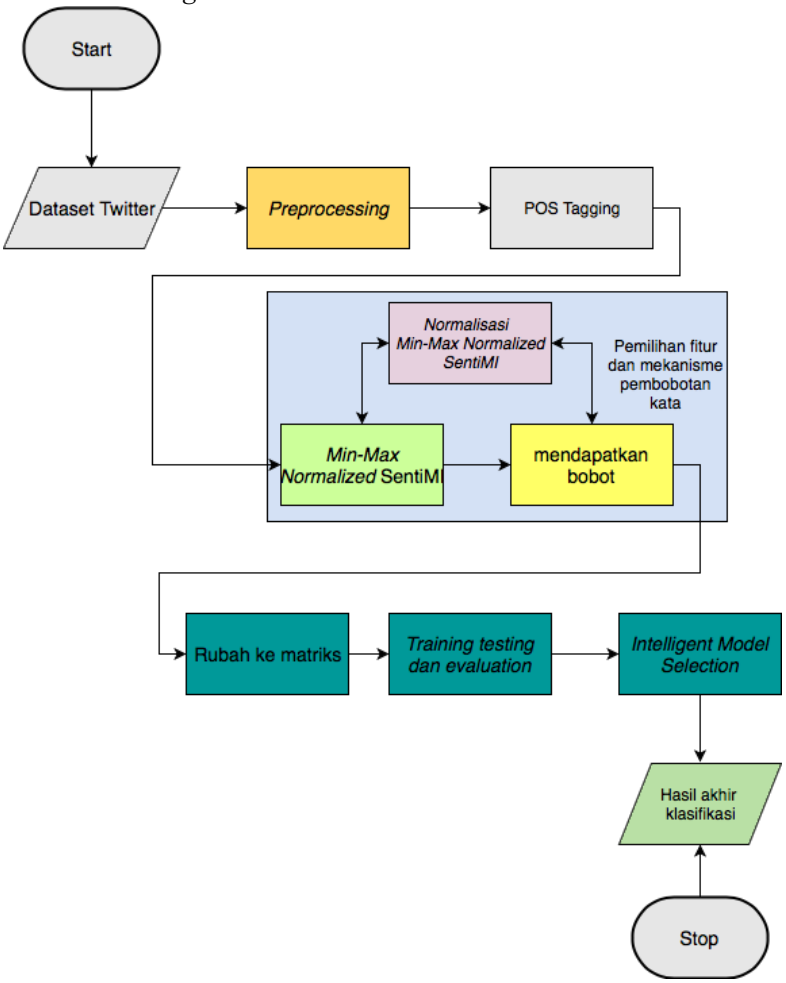


Gambar 3. 1 Diagram alir implementasi pembobotan kata dan *POS Tagging*

Pada diagram alir implementasi *POS Tagging* dan pembobotan kata input untuk proses awal tersebut adalah mendapatkan hasil *tweet* hasil *preprocessing*. Setiap *tweet* akan dipisah melalui setiap katanya. Setiap kata tersebut akan dicari kelas *POS Tagging* dalam Bahasa Indonesia. Setelah mendapatkan kelas setiap katanya, maka proses berikutnya adalah membuka *library SentiWordNet* Indonesia. Setiap kata yang telah diberi kelas kata dicari pada *library SentiWordNet.* Setelah menemukan kata tersebut, maka akan dihitung nilai polaritas kata tersebut. Nilai polaritas didapat dari nilai bobot positif – bobot negatif. Jika nilai akhir yang didapat tidak sama dengan nol, maka kata tersebut akan diambil. Namun jika nilai polaritas sama dengan nol, maka kata tersebut akan tidak akan dihitung sebagai fitur. Kata yang diambil kemudian akan dicari kelas katanya yang sesuai dengan kelas kata *POS Tagging* Bahasa Indonesia. Kelas kata yang diambil diantaranya kata sifat, kata kerja, kata benda, dan kata keterangan. Setelah mendapatkan kata yang sesuai dengan kelas kata pada *SentiWordNet*, maka kata-kata tersebut akan disimpan. Kata yang disimpan akan dihitung jumlah frekuensi bobot positif dan frekuensi bobot negatif sesuai dengan bobot polaritas yang dihitung setiap katanya. Setelah mendapatkan frekuensi masing-masing kata, tahap selanjutnya adalah perhitungan bobot menggunakan *Point-wise Mutual Information* menggunakan persamaan pada 2.2. Setelah mendapatkan nilai bobot setiap kata, maka proses selanjutnya adalah proses normalisasi bobot. Menggunakan persamaan 2.1, setiap bobot akan dihitung nilai normalisasi dengan batas yang telah ditentukan. Setelah mendapatkan bobot normalisasi tersebut, semua kata tersebut disimpan dan digunakan untuk tahap berikutnya.

Tahap pemilihan fitur dan normalisasi bobot bertujuan untuk memberikan nilai bobot yang berada pada batas yang telah ditentukan. Pembobotan pada tahap ini menggunakan *poin-wise mutual information.* Setelah mendapatkan bobot pada setiap kata, tahap berikutnya adalah normalisasi pada setiap bobot tersebut.

Tahap klasifikasi menggunakan *Intelligent Model Selection* merupakan tahap terakhir pada sistem ini. Tahapan ini menggunakan *Support Vector Machine* (SVM) sebagai *classifier* dan *Cross-validation* sebagai metode untuk pembagian data *testing* dan *training*.



Gambar 3. 2 Diagram alir metode SWIMS

3.2 Perancangan Data

Subbab ini akan membahas perancangan data yang merupakan bagian penting karena akan menjadi acuan untuk membuat skenario uji coba dari tugas akhir. Data yang digunakan adalah data *tweet* dari tiga provider di Indonesia, yaitu XL, Telkomsel, dan IM3. Data *tweet* sebelumnya sudah diberi label yaitu positif atau negatif. Terdapat total data sebanyak 86 tweet dengan 43 label positif dan 43 label negatif. Data tersebut didapatkan dari gabungan 46 *tweet* XL, 18 *tweet* Telkomsel, dan 22 *tweet* IM3 yang diambil pada bulan April 2016 melalui sosial media Twitter.com. Berikut contoh dataset dari *tweet* ditunjukkan oleh Tabel 3.1.

Tabel 3. 1 Contoh dataset *tweet*

Tweet	**label**
@myxl masih kurang bagus sinyalnya didaerah saya	0
kecepatan 4g @myxl kembali stabil, balik pake xl atau tetap dengan @telkomsel?	1

3.3 Perancangan Proses

Subbab ini membahas mengenai perancangan proses yang dilakukan untuk setiap tahap pada metode SWIMS berdasarkan Gambar 3.2.

3.3.1. Pre-processing

Pada tahap ini, *tweet* pada masing-masing provider akan dilakukan proses *pre-processing*. Proses itu meliputi, *stopword removal, replace slang, tokenization,* dan *stemming*. Penggunaan *library* seperti NLTK dan sastrawi pada python sangat membantu proses tersebut. Sehingga pada tahap ini didapatkan *tweet* yang sudah menjadi kata dasar dan dapat dilakukan tahap pembobotan. Output dari proses ini ditunjukkan pada Gambar 3.3

```
kurang bagus sinyal daerah
hai daerah semper jakarta utara sinyal tidak stabil
sinyal jelek sombong
sayang sinyal kurang rata
```

Gambar 3. 3 Output proses *preprocessing*

3.3.2. POS Tagging

Pada proses ini setiap kata dari output *preprocessing* akan diberi label setiap *(POSTag)*. Hal ini bertujuan untuk dapat memberikan kelas kata sehingga dapat digunakan kepada *SentiWordNet* Bahasa. Menggunakan *corpus* kelas kata, setiap kata pada *tweet* akan diberi kelas kata. Terdapat beberapa kelas kata yang digunakan antara lain, kata kerja, kata sifat, kata benda, dan kata keterangan. Output dari proses ini ditunjukkan oleh Gambar 3.4.

```
[[0, 'asyik', 'adj', '1'],
 [0, 'streamonterus', 'n', '1'],
 [0, 'spotify', 'n', '1'],
 [0, 'langgan', 'v', '1'],
 [0, 'freedomcombo', 'n', '1'],
 [0, 'juara', 'n', '1'],
 [1, 'min', 'adj', '0'],
 [1, 'komplain', 'n', '0'],
 [1, 'tinggal', 'v', '0'],
```

Gambar 3. 4 Output proses *POS Tagging*

3.3.3. Pembobotan Kata dan Pemilihan Fitur

Pada proses ini, hasil *tweet* yang telah diperoleh dari *pre-processing* akan dilakukan proses pembobotan kata (*polarity score)*. Sebelum dilakukan proses pembobotan kata, hasil setiap kata dari tahap *POS Tagging* akan digunakan pada tahap ini. Hasilnya berupa pasangan kata dan kelas kata pada masing-masing kata pada *tweet*. Kemudian pasangan kata dan POS tersebut akan

dicari pada kamus *SentiWordNet* Bahasa. Sebelum dicari pada kamus *SentiWordNet,* kelas kata pada Bahasa Indonesia harus dirubah terlebih sesuai dengan format kelas kata pada *SentiWordNet.* Perubahan format pada kelas kata dapat ditunjukkan pada Tabel 2.1. *SentiWordNet* adalah sumber *lexical* (kamus) untuk *opinion mining.* SentiWordNet terdiri dari beberapa *synset (Synonim Set)* yang memiliki dua nilai untuk masing-masing yaitu, positif dan negatif. Setiap nilai *term* dalam *synset* juga diurutkan berdasarkan seberapa sering digunakan dalam pengertian itu. Dari *SentiWordNet* akan didapatkan bobot polaritas kata. Polaritas pada kata tersebut dapat bernilai positif atau negatif. Setiap pasangan kata dan POS dapat muncul lebih dari satu kali karena tingkat penggunaan pada sebuah kalimat. Setiap kata memiliki nilai polaritas positif dan negatif. Nilai total polaritas didapat dari hasil polaritas positif – polaritas negatif. Nilai total polaritas yang tidak sama dengan 0 yang akan digunakan sebagai fitur kata. Proses rinci pembobotan ditunjukkan pada Gambar 3.2. Adapun contoh hasil dari proses perhitungan frekuensi *POS Tagging* pada dataset ditunjukkan oleh Tabel 3.2. Sedangkan contoh *corpus* pada SentiWordNet Bahasa Indonesia ditunjukkan oleh tabel 3.3. Output dari proses ini dapat ditunjukkan pada Gambar 3.5.

Tabel 3. 2 Contoh perhitungan frekuensi *POS Tagging*

Kata	POS	Jumlah positif	Jumlah negatif
Bagus	v	12	0
Jelek	a	3	18

Tabel 3. 3 Contoh *corpus* pada SentiWordNet

Synset	Language	Goodness	Lemma	P	N
00001740-a	B	L	akauntan	0,125	0
00001740-a	B	L	Berdaya upaya	0,125	0

```
,ID,tweet_id,term,pos,pos_fre,neg_fre,labe
0,0,0,asyik,adj,5,2,1,-2.3918860593308198,
1,1,0,juara,n,2,1,1,-2.4914217328817343,0.
2,2,1,tinggal,v,3,2,0,-3.092780838470603,0
```

Gambar 3. 5 Output proses pembobotan kata

3.3.4. Min-max Normalized SentiMI

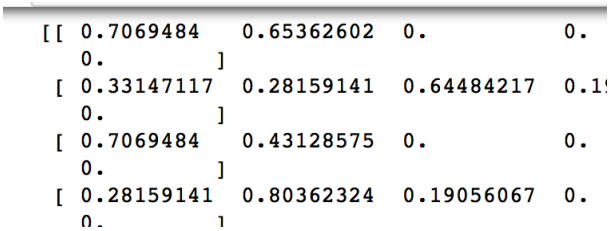
Proses yang dilakukan pada tahap ini adalah normalisasi menggunakan hasil pembobotan kata pada proses sebelumnya. Setiap kata yang telah memiliki bobot akan dilakukan proses normalisasi menggunakan persamaan 2.1. Menggunakan batas normalisasi (-1,1). Setelah mendapatkan nilai normalisasi dari setiap kata, maka kata tersebut akan disimpan ke dalam sebuah *file* untuk dilakukan proses berikutnya. Output dari proses ini ditunjukkan oleh Gambar 3.6.

```
pos_fre,neg_fre,label,fea_weight,normalize_weight
-2.3918860593308198,0.7069483984668539
-2.4914217328817343,0.6536260162498211
-3.092780838470603,0.3314711652907798
```

Gambar 3. 6 Output proses normalisasi

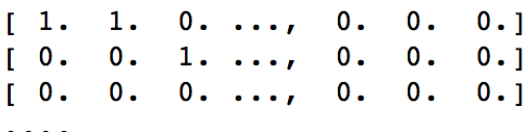
3.3.5. Merubah Format ke Bentuk Matriks

Sebelum masuk pada proses klasifikasi, setiap bobot dan nilai setiap kata pada *tweet* perlu dirubah ke dalam format yang dapat dibaca oleh *classifier*. Oleh karena itu perlu dirubah kedalam matriks. Matriks tersebut menunjukkan jumlah fitur yang digunakan. Terdapat tiga jenis fitur yang akan digunakan yaitu, fitur *POS Tagging*, fitur *Term Presence*, dan fitur gabungan kedua fitur tersebut. Output fitur POS Tagging ditunjukkan Gambar 3.7. Output fitur Term Presence ditunjukkan Gambar 3.8. Sedangkan fitur gabungan ditunjukkan Gambar 3.9.



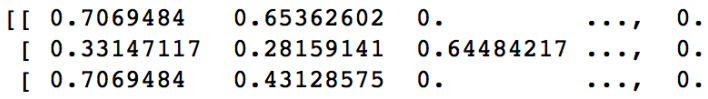
```
[[ 0.7069484   0.65362602  0.          0.
   0.        ]
 [ 0.33147117  0.28159141  0.64484217  0.1
   0.        ]
 [ 0.7069484   0.43128575  0.          0.
   0.        ]
 [ 0.28159141  0.80362324  0.19056067  0.
   0.        ]
```

Gambar 3. 8 Matriks fitur *POS Tagging*



```
[[ 1.  1.  0. ...,  0.  0.  0.]
 [ 0.  0.  1. ...,  0.  0.  0.]
 [ 0.  0.  0. ...,  0.  0.  0.]
 ...
```

Gambar 3. 7 Matriks fitur *Term Presence*



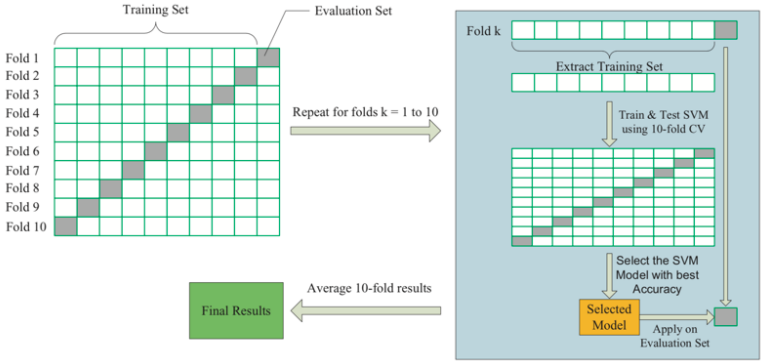
```
[[ 0.7069484   0.65362602  0.          ...,  0.
 [ 0.33147117  0.28159141  0.64484217  ...,  0.
 [ 0.7069484   0.43128575  0.          ...,  0.
```

Gambar 3. 9 Matriks fitur gabungan

3.3.6. Klasifikasi Menggunakan *Intelligent Model Selection* (IMS)

Proses terakhir dari metode SWIMS adalah proses klasifikasi. Klasifikasi pada SWIMS menggunakan tiga pembagian data yaitu data testing, data traning, dan data evaluasi. Proses tersebut disebut dengan *Intelligent Model Selection* (IMS). *Framework* SWIMS merupakan salah satu metode yang berfokus dalam meningkatkan kinerja dari kamus sentimen umum yang mempertimbangkan domain independen sebagai perhatian yang utama. *Framework* ini memberikan pengembangan pada tahap validasi hasil klasifikasi data dengan mengubah proses 10-*Fold cross-validation.* Pada umumnya 10-*Fold cross-validation*, 10 model dibentuk dari 10 *training set.* Kemudian data akan diuji pada masing-masing *test sets.*

Namun pada *framework* SWIMS *dataset* akan dibagi menjadi 10 model *fold* yang sama, dimana 9*fold* digunakan untuk *training testing* dan *fold* ke-10 digunakan untuk *evaluation set*. Data pada 9*fold* dibagi lagi menjadi rasio 90 – 10 dan 10-*Fold cross-validation* diterapkan pada rasio tersebut. Hasil akurasi untuk setiap *fold* pada sub bagian akan dicatat dan model yang memiliki nilai akurasi tinggi akan dipilih. Model yang dipilih akan diterapkan pada bagian *evaluasi set*. Hal tersebut diulang sebanyak 10 kali dengan rangkaian *training*, *test*, dan *evaluation* sets yang berbeda. Kemudian hasil rata-rata akan dihitung untuk pemilihan model. Hasil dari proses klasifikasi ini berupa nilai dari evaluasi *accuracy.* Adapun diagaram alir dari proses IMS ditunjukkan oleh Gambar 3.10.



Gambar 3. 10 Diagram ali*r Intelligent Model Selection (IMS)*

BAB IV
IMPLEMENTASI

Bab ini membahas implementasi dari perancangan sistem sesuai dengan perancangan yang telah dibuat. Bahasa pemrograman yang digunakan untuk implementasi sistem adalah Bahasa pemrograman python.

4.1 Lingkungan Implementasi

Lingkungan implementasi sistem yang digunakan untuk mengembangkan tugas akhir ini memiliki spesifikasi perangkat keras dan perangkat lunak yang ditunjukkan oleh Tabel 4.1.

Tabel 4. 1 Spesifikasi Perangkat

Perangkat	Spesifikasi
Perangkat Keras	• Prosesor: Intel® Core™ i5 CPU @ 2.3GHz ~ 3.6GHz • Memori: 8 GB
Perangkat Lunak	• Sistem Operasi macOS High Sierra 64-bit • Perangkat Pengembang Ananconda 3.4 • Perangkat Pembantu Sublime Text 3, Microsoft Excel 2016, Microsoft Word 2016, Microsoft Power Point 2016

4.2 Implementasi Proses

Implementasi proses dilakukan berdasarkan perancangan proses yang dijelaskan pada bab analisis dan perancangan. Untuk implementasi setiap tahapan, *file* yang dibaca di dalam program tergantung dari skenario data yang sedang dijalankan. Berikut adalah daftar dari nama *file dataset* yang digunakan:

Tabel 4. 2 Nama *File* pada Setiap Proses

Skenario Data	Data
XL	data_XL.txt
Telkomsel	data_TELKOMSEL.txt
IM3	data_IM3.txt
POS Tagging Bahasa Indonesia	kata_dasar.txt
Pre-processing data	slang.dic
Pre-processing data	stopwords.txt
Pembobotan kata dari SentiWordNet	data_TOTAL.txt
Normalisasi bobot	data_DIC_TOTAL.csv
Klasifikasi	data_OUT_TOTAL.csv

4.2.1. Implementasi Tahap *Pre-Processing*

Bagian ini membahas implementasi tahapan *Pre-processing* yang merupakan tahapan awal pada tugas akhir. Proses *stopword removal, symbol removal, replacing slang,* dan *tokenization* diterapkan pada tahap ini. Bagian pertama mengenai proses ini adalah menentukan *regex* dan *stemmer* yang ditunjukkan pada Kode Sumber 4.1. Bagian kedua adalah proses implementasi semua *regex* dan *stemmer* ditunjukkan pada Kode Sumber 4.2.

```
1  #regresion untuk menghapus simbol
2  regexPattern1 = re.compile(r'(@[A-Za-z0-9]+)|([^0-9A-Za-z
   \t])|(\w+:\/\/\S+)')
3  regexPattern2 = re.compile(r'\w+:\/{2}[\d\w-]+(\.[\d\w-
   ]+)*(?:(?:\/[^\s/]*))*')
4  regexPattern3 = re.compile("^\d+\s|\s\d+\s|\s\d+$")
5  regexTokenizer = RegexpTokenizer(r'\w+')
6  #stemmer dari sastrawi untuk Bahasa Indonesia
7  factory = StemmerFactory()
8  stemmer = factory.create_stemmer()
```

```
9  #membaca kamus slang untuk diganti
10 slangDic = dict((t.split(":")[0],t.split(":")[1]) for t in slangDic)
11 #membaca stopword untuk dihapus
12 stop_ind = [t.strip() for t in stop]
```

Kode Sumber 4. 1 Implementasi Proses *Preprocessing* Bag.1

```
1  tweet = re.sub(regexPattern1,' ',tweet)
2  tweet = re.sub(regexPattern2, ' ', tweet)
3  tweet = re.sub(regexPattern3, ' ', tweet)
4  #megganti semua slang pada semua dataset
5  for slang, formal in slangDic.items():
6      tweet = tweet.replace(slang,formal)
7  tweet = regexTokenizer.tokenize(tweet)
8  tweet_stop = [word for word in tweet if word not in stop_ind]
```

Kode Sumber 4. 2 Implementasi Proses *Preprocessing* Bag.2

4.2.2. Implementasi Tahap *POS Tagging* dan Pembobotan Kata

Dalam mengimplementasikan tahap pembobotan kata, digunakan pustaka *pandas* untuk menyimpan hasil pembobotan kata di dalam sebuah file yang memiliki format .csv.

Sebelum dilakukan pembobotan kata, perlu adanya proses *POS Tagging* untuk mendapatkan kelas kata suatu *tweet*. Fungsi *POS Tagging* ditunjukkan pada Kode Sumber 4.3. Implementasi fungsi *POS Tagging* pada dataset ditunjukkan pada Kode Sumber 4.4.

```
1  def loadPos_id(file = 'data/kata_dasar.txt'):
2      kata_pos = {}
3      df=open(file,"r",encoding="utf-8", errors='replace')
4      data=df.readlines();df.close()
5      for line in data:
6          d = line.split()
7          kata = d[0].strip()
```

```
8                 pos = d[-1].strip().replace("(","").replace(')',")
9                 kata_pos[kata] = pos
10        return kata_pos
```

Kode Sumber 4. 3 Implementasi Membaca Berkas *POS Tagging*

```
11  try:
12      pos_term.append([idx, term , kata_pos[term], label[0]])
13  except:
14      pos_term.append([idx, term , 'n', label[0]])
```

Kode Sumber 4. 4 Implementasi Fungsi *POS Tagging*

Setelah diperoleh hasil *POS Tagging,* proses selanjutnya adalah mencari *POS Tagging* beserta kata yang sudah didapat didalam *corpus* SentiWordNet Bahasa Indonesia untuk memperoleh bobot. Implementasi tersebut ditunjukkan pada Kode Sumber 4.5.

```
1   #mencari kelas kata yang sesuai dengan Sentiwordnet
2   if pos_term[itter][2] == "adj":
3       pos_synset = "a"
4   elif pos_term[itter][2] == "n":
5      pos_synset = "n"
6   elif pos_term[itter][2] == "v":
7      pos_synset = "v"
8   elif pos_term[itter][2] == "k":
9      pos_synset = "r"
10  #mencari kata berbahasa Indonesia
11  item = item[item['language'].str.contains("I|B")]
12  item = item[item['synset'].str.contains(pos_synset)]
13  item = item[item['goodness'].str.contains("Y|O|M")]
14  #mencari nilai jumlah polaritas kata yang netral atau tidak sama dengan 0
15  item = item[item['PosScore'] - item['NegScore'] != 0]
```

Kode Sumber 4. 5 Implementasi Pembobotan Kata *SentiWordNet*

4.2.3. Implementasi Tahap Pemilihan Fitur dan Normalisasi Bobot

Input pada tahap ini berupa *file* CSV yang berisi kata dengan jumlah polaritas positif dan negatif kata tersebut yang didapat melalui *SentiWordNet* Bahasa pada proses sebelumnya. Setiap kata pada *file* ini yang akan menjadi fitur untuk proses klasifikasi. Setiap kata dihitung bobot jumlah frekuensi kelas positif dan negatif menggunakan rumus *point-wise mutual information.* Setelah mendapatkan nilai bobot menggunakan PMI, maka tahap selanjutnya adalah proses normalisasi. Menggunakan rumus *normalize-sentiMI* untuk mendapatkan bobot dengan batas yang telah ditentukan (-1,1). Implementasi proses perhitungan frekuensi kelas dan PMI ditunjukkan pada Kode Sumber 4.6. Implementasi normalisasi bobot dari hasil PMI ditunjukkan pada Kode Sumber 4.7.

```
1  selected_term = file.loc[itter]
2  others_term = file[file['ID'] != itter]
3  if selected_term['label'] == 0:
4      A = selected_term['neg_fre']
5      B = selected_term['pos_fre']
6      C = others_term['neg_fre'].sum()
7      tmp = (A*N) / ((A+B)*(A+C))
8      result = math.log2(tmp)
9  elif selected_term['label'] == 1:
10     A = selected_term['pos_fre']
11     B = selected_term['neg_fre']
12     B = selected  term['neg  fre']
13     tmp = (A*N) / ((A+B)*(A+C))
14     result = math.log2(tmp)
```

Kode Sumber 4. 6 Implementasi Pemilihan Bobot Kata

```
1  selected = file.loc[i]
2  v = selected['fea_weight']
```

```
3  nor_minmax = (((v - minA) / (maxA - minA)) * (newMax - newMin)) + newMin
4  normalize.append(nor_minmax)
```

Kode Sumber 4. 7 Implementasi Normalisasi

Setelah mendapatkan nilai bobot setiap kata pada *file* tersebut, hasil dari normalisasi akan di simpan dalam *file* CSV baru data_OUT_TOTAL.csv.

4.2.4. Implementasi Tahap Klasifikasi Menggunakan *Intelligent Model Selection*

Untuk mengimplementasikan metode *Intelligent Model Selection*, terdapat tiga bagian terpisah. Bagian pertama adalah pembuatan matrik fitur berdasarkan nilai bobot normalisasi pada setiap *tweet*. Hanya kata yang memiliki nilai bobot yang akan memiliki nilai pada matrik selain itu fitur lain bernilai 0. Kode sumber 4.8 menujukkan implementasi dari bagian pertama untuk menentukan matrik data sebagai *input* pada *classifier*.

```
1  for a, tweet in selected.iterrows():
2      feature = float(tweet['normalize_weight'])
3      if len(selected.index) < pd.value_counts(file['tweet_id']).max():
4          b = len(selected.index)
5          while b < pd.value_counts(file['tweet_id']).max():
6              feature = float(0)
7      elif len(selected.index) == 0:
8      i = 0
9      while i < pd.value_counts(file['tweet_id']).max():
10         feature = float(0)
```

Kode Sumber 4. 8 Implementasi Fitur Matrik pada *Tweet*

Matrik pada bagian pertama merupakan fitur kata yang di dapat dari *SentiWordNet* Bahasa, yang terdiri dari kata kerja, kata

sifat, kata keterangan, dan kata benda. Namun untuk implementasi metode IMS diperlukan pula fitur *Term Presence (TP)*. Fitur ini merupakan kemunculan kata pada setiap *tweet* yang sudah didapat dari *SentiWordNet* Bahasa. Jumlah fitur TP merupakan jumlah dari semua kata yang di dapat dari *SentiWordNet* Bahasa. Kode sumber 4.9 menunjukkan implementasi dari bagian kedua untuk implementasi matrik *Term Presence (TP)*.

```
1  for k in range(len(df['term'])):
2          dd = df.iloc[k]['term']
3          if any(dd in s for s in tmp):
4                  tt.append(float(1))
5          else:
6                  tt.append(float(0))
7  #penggabungan kedua matrik
8  data_total = np.column_stack([data, term_presence])
   print(data_total.shape)
```

Kode Sumber 4. 9 Implementasi Fitur *Term Presence (TP)*

Setelah mendapatkan total fitur untuk semua matrik, maka bagian terakhir adalah untuk melakukan proses klasifikasi menggunakan metode *Intelligent Model Selection.* Menggunakan beberapa parameter untuk proses *tuning* klasifikasi. Menggunakan *fold* dengan nilai K = 2 sampai K = 10. *Kernel* SVM menggunakan skenario *linear* dan RBF. Kode sumber 4.10 menunjukkan dari bagian klasifikasi menggunakan metode IMS.

```
1  for train_test_index, validation_index in skf.split(data, label_target):
2          for i in train_test_index:
3                  ltt.append(label_target[i])
4                  dtt.append(data[i])
5          #menemukan best model menggunakan 90% data
```

```
6   score2 = cross_val_score(SVCclassification_1, data_testing_test,
    label_testing_test, cv=skf)
7   #menghitung nilai model dengan akurasi paling tinggi
8   best_score_index = np.argmax(score2)
9   for f_index,g_index in
    skf.split(data_testing_test,label_testing_test):
10      if counter == best_score_index:
11          for c in f_index:
12          best_model_tt.append(data_testing_test[c])
13          label_best_model_tt.append(label_testing_test[c])
14  #menyimpan index evaluasi set
15  for j in validation_index:
16      evltt.append(label_target[j])
17      evdtt.append(data[j])
18  #fit model dengan menggunakan evaluasi data
19  SVCclassification_2.fit(best_model, best_model_label)
20  #menampilkan hasil classification report
21  print(classification_report(evaluation_label_datatest, y_predic,
    target_names=target_names))
```

Kode Sumber 4. 10 Implementasi *IMS*

BAB V
PENGUJIAN DAN EVALUASI

Bab ini membahas uji coba dan evaluasi terhadap sistem yang telah dikembangkan untuk mengklasifikasi sentimen pada *tweet* provider Indonesia XL, Telkomsel dan IM3 pada bulan April 2016.

5.1 Lingkungan Pengujian

Lingkungan penguijan sistem pada pengerjaan tugas ini dilakukan pada lingkungan dan alat kakas sebagai berikut:

Prosesor	: Prosesor: Intel® Core™ i5 CPU @ 2.3GHz ~ 3.6GHz
RAM	: 8 GB
Jenis *Device*	: Laptop
Sistem Operasi	: macOS High Sierra 64-bit

5.2 Data Uji Coba

Data yang digunakan untuk uji coba metode SWIMS adalah data tweet provider Indonesia yaitu XL, Telkomsel, dan IM3. Pembagian data *training* dan *testing* dilakukan dengan menggunakan metode *cross validation.*

5.3 Skenario Uji Coba

Subbab ini akan menjelaskan skenario uji coba yang telah dilakukan. Terdapat beberapa skenario uji coba yang telah dilakukan. Pada masing-masing skenario dilakukan uji coba untuk jumlah nilai K pada metode IMS untuk *cross validation* dan jenis *kernel* pada metode *SVM.*

1. Skenario Pengujian 1: dalam skenario ini akan dilakukan perhitungan nilai *accuracy, precision, recall,* dan *f-measure* dengan menggunakan jumlah fitur yang sesuai dengan jumlah pasangan kata *POS Tagging* pada *dataset* yang telah didapat dari

SentiWordNet. Terdapat 223 kata yang telah diberi bobot. Pada matriks fitur terdapat 8 fitur untuk setiap *tweet.* Pengujian ini bertujuan untuk mendapatkan hasil iterasi nilai K dengan nilai *accuracy* terbaik dengan membandingkan penggunaan *kernel linear* dan *kernel RBF*.

2. Skenario Pengujian 2: dalam skenario ini akan dilakukan perhitungan nilai *accuracy, precision, recall,* dan *f-measure* dengan menggunakan jumlah fitur *term presence* pada *dataset tweet.* Terdapat 100 kata unik pada *tweet* yang didapat dari *SentiWordNet* yang muncul pada dataset. Pengujian ini bertujuan untuk mendapatkan hasil iterasi nilai K dengan nilai *accuracy* terbaik dengan membandingkan penggunaan *kernel linear* dan *kernel RBF*.

3. Skenario Pengujian 3: dalam skenario ini akan dilakukan perhitungan nilai *accuracy, precision, recall,* dan *f-measure* dengan menggunakan jumlah fitur *POS Tagging* Indonesia dan digabung dengan *Term Presence* pada dataset *tweet.* Terdapat total 108 fitur dalam fitur gabungan antara *POS Tagging* dan *Term Presence* untuk setiap *tweet*. Pengujian ini bertujuan untuk mendapatkan hasil iterasi nilai K dengan nilai *accuracy* terbaik dengan membandingkan penggunaan *kernel linear* dan *kernel RBF*.

5.3.1. Skenario Pengujian 1

Pada skenario ini dilakukan uji coba klasifikasi data menggunakan jumlah fitur *POS Tagging* dengan menggunakan nilai $K = \{2,3,4,5,6,7,8,9,10\}$ pada *cross validaiton.* Selain menggunakan nilai *K* yang berbeda, pada pengujian ini dilakukan

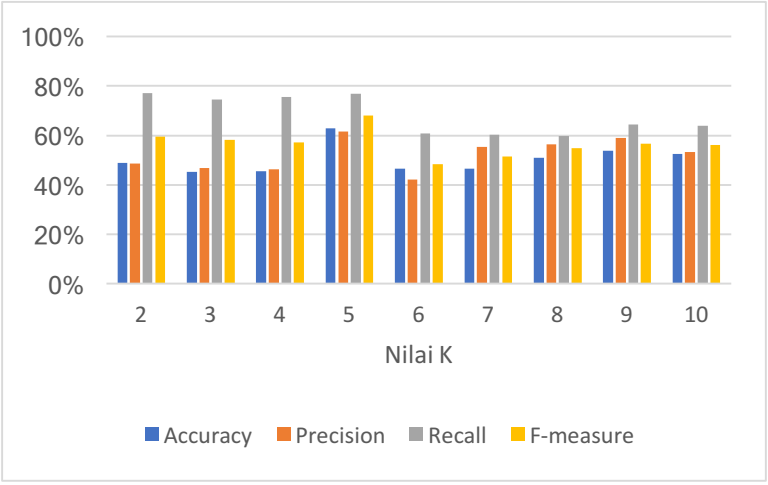
perbandingan dua kernel yaitu *linear* dan *RBF*. Terdapat 233 kata serta 8 fitur pada masing-masing *tweet* yang telah diberi bobot menggunakan *normalize-sentiMI* pada dataset. Hasil uji coba terbaik pada *kernel linear* ditunjukkan saat menggunakan nilai *K = 5* dengan nilai rata-rata *accuracy* sebesar 63,0%, *precision* 61,7%, *recall* 76,9%, dan *f-measure* 68,0%. Sedangkan pengujian menggunakan *kernel RBF,* nilai terbaik ditunjukkan saat menggunakan nilai K = 2 dengan nilai rata-rata *accuracy sebesar 60,5%, precision 57,4%, recall 81,2%, dan f-measure 67,0%.* Berikut hasil lengkap pada pengujian 1 menggunakan *kernel linear* ditunjukkan pada Tabel 5.2 dan plot data ditunjukkan pada Gambar 5.1. Hasil lengkap pada pengujian 1 menggunakan *kernel RBF* ditunjukkan pada Table 5.2 dan plot data ditunjukkan pada Gambar 5.2.

Tabel 5. 1 Hasil performa pengujian menggunakan fitur *POS Tagging* dengan *kernel Linear*

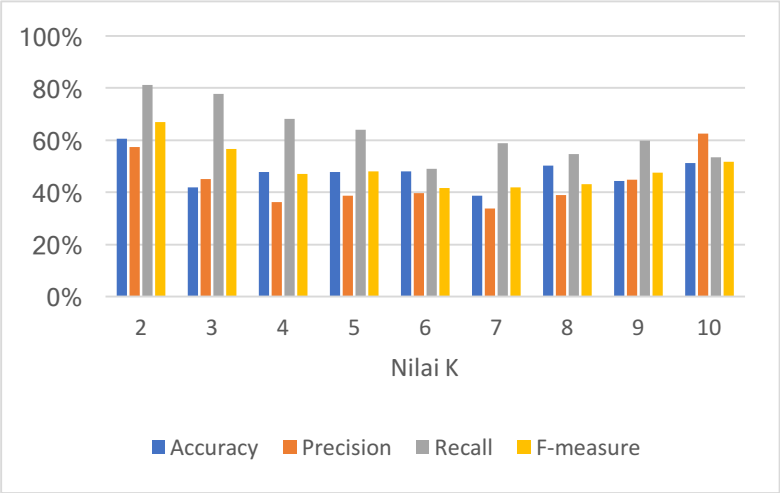
Nilai K	Accuracy	Precision	Recall	F-measure
2	48,8%	48,7%	77,2%	59,4%
3	45,3%	46,9%	74,6%	58,1%
4	45,6%	46,4%	75,5%	57,1%
5	**63,0%**	**61,7%**	**76,9%**	**68,0%**
6	46,6%	42,1%	60,9%	48,5%
7	46,6%	55,4%	60,4%	51,5%
8	51,0%	56,5%	59,8%	54,9%
9	53,8%	58,9%	64,4%	56,7%
10	52,5%	53,3%	64,0%	56,1%

Tabel 5. 2 Hasil performa pengujian menggunakan fitur *POS Tagging* dengan *kernel RBF*

Nilai K	Accuracy	Precision	Recall	F-measure
2	**60,5%**	**57,4%**	**81,2%**	**67,0%**
3	41,9%	45,1%	77,8%	56,7%
4	47,7%	36,2%	68,2%	47,1%
5	47,7%	38,6%	63,9%	47,9%
6	47,9%	39,6%	48,9%	41,7%
7	38,6%	33,7%	58,7%	41,8%
8	50,1%	38,8%	54,6%	43,1%
9	44,2%	44,9%	59,8%	47,4%
10	51,2%	62,5%	53,5%	51,6%



Gambar 5. 1 Plot data menggunakan fitur *POS Tagging* dengan *kernel Linear*



Gambar 5. 2 Plot data menggunakan fitur *POS Tagging* dengan *kernel RBF*

5.3.2. Skenario Pengujian 2

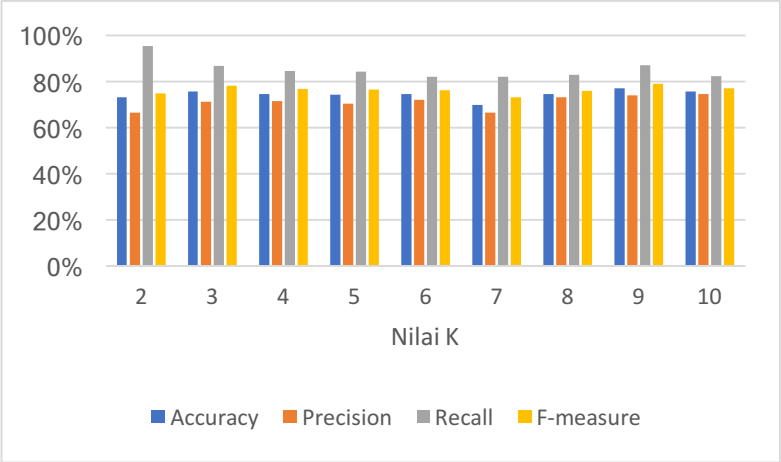
Pada skenario ini dilakukan uji coba klasifikasi data menggunakan jumlah fitur *term presence (TP)* kata yang telah didapatkan melalui *SentiWordNet* dengan menggunakan *K = {2,3,4,5,6,7,8,9,10}* pada *cross validation*. Selain menggunakan nilai *K* yang berbeda, pada pengujian ini dilakukan perbandingan dua *kernel* yaitu *linear* dan *RBF*. Hasil uji coba terbaik menggunakan *kernel linear* ditunjukkan saat menggunakan nilai *K = 9* dengan nilai rata-rata *accuracy* sebesar 77,0%, *precision* 74,0%, *recall* 87,0%, dan *f-measure* 79,0%. Sedangkan pengujian menggunakan *kernel RBF,* nilai terbaik ditunjukkan saat menggunakan nilai K = 10 dengan nilai *accuracy* 74,7%, *precision* 70,8%, *recall* 93,0%, *dan f-measure* 79,2%. Berikut hasil lengkap pada pengujian 2 menggunakan *kernel linear* ditunjukkan pada Tabel 5. 3 dan plot data pada Gambar 5.3. Hasil lengkap pada pengujian 2 menggunakan *kernel RBF* ditunjukkan pada Table 5.4 dan plot data ditunjukkan pada Gambar 5.4.

Tabel 5. 3 Hasil performa pengujian menggunakan fitur *Term Presence* (TP) dengan *kernel Linear*

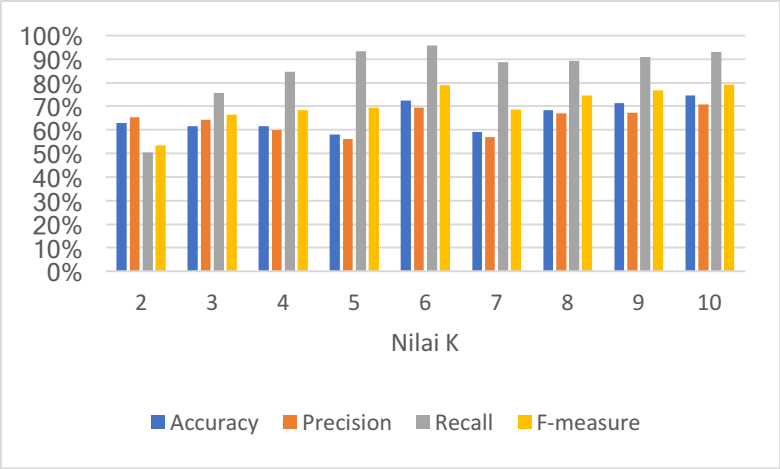
Nilai K	Accuracy	Precision	Recall	F-measure
2	73,3%	66,5%	95,5%	74,8%
3	75,7%	71,2%	86,7%	78,1%
4	74,5%	71,5%	84,6%	76,7%
5	74,3%	70,5%	84,2%	76,6%
6	74,5%	72,2%	82,1%	76,2%
7	69,8%	66,6%	82,0%	73,3%
8	74,7%	73,1%	82,9%	76,0%
9	**77,0%**	**74,0%**	**87,0%**	**79,0%**
10	75,8%	74,7%	82,5%	77,0%

Tabel 5. 4 Hasil performa pengujian menggunakan fitur *Term Presence* (TP) dengan *kernel RBF*

Nilai K	Accuracy	Precision	Recall	F-measure
2	62,8%	65,3%	50,5%	53,4%
3	61,5%	64,3%	75,6%	66,3%
4	61,6%	59,8%	84,5%	68,2%
5	57,9%	56,1%	93,3%	69,3%
6	72,5%	69,3%	95,8%	78,9%
7	59,2%	56,9%	88,8%	68,7%
8	68,4%	67,1%	89,2%	74,5%
9	71,4%	67,3%	90,9%	76,8%
10	**74,7%**	**70,8%**	**93,0%**	**79,2%**



Gambar 5. 3 Plot data menggunakan fitur *Term Presence (TP)* dengan *kernel Linear*



Gambar 5. 4 Plot data menggunakan fitur *Term Presence (TP)* dengan *kernel RBF*

5.3.3. Skenario Pengujian 3

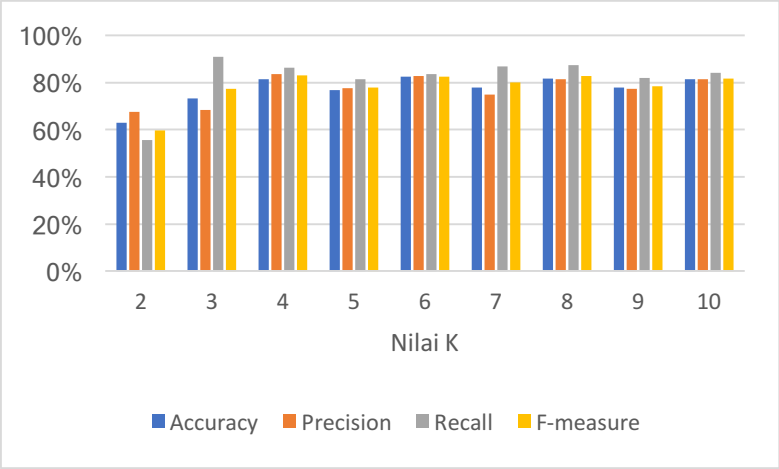
Pada skenario pengujian ini fitur kata yang digunakan merupakan gabungan dari fitur *term presence (TP)* dan fitur *POS Tagging* yang telah didapatkan dari *corpus SentiWordNet*. Pada pengujian ini menggunakan *cross validation* dengan nilai *K = {2,3,4,5,6,7,8,9,10}*. Selain menggunakan nilai *K* yang berbeda, pada pengujian ini membandingkan dua *kernel,* yaitu *linear* dan *RBF*. Hasil uji coba terbaik menggunakan *kernel linear* ditunjukkan saat menggunakan nilai *K = 6* dengan nilai rata-rata *accuracy* sebesar 82,5%, *precision* 82,7%*, recall* 83,5%, dan *f-measure* 82,4%. Sedangkan pengujian menggunakan *kernel RBF,* nilai terbaik ditunjukkan saat menggunakan nilai K = 9 dengan nilai rata-rata *accuracy* sebesar 71,4%, *precision* 66,3%, *recall* 93,7%*,* dan *f-measure* 77,1%. Berikut hasil lengkap pada pengujian 3 menggunakan *kernel linear* ditunjukkan pada Tabel 5.5 dan plot data ditunjukkan pada Gambar 5.5. Hasil lengkap pada pengujian 2 menggunakan *kernel RBF* ditunjukkan pada Table 5.6 dan plot data ditunjukkan pada Gambar 5.6.

Tabel 5. 5 Hasil perhitungan menggunakan fitur gabungan *POS Tagging* dan *Term Presence* dengan *kernel Linear*

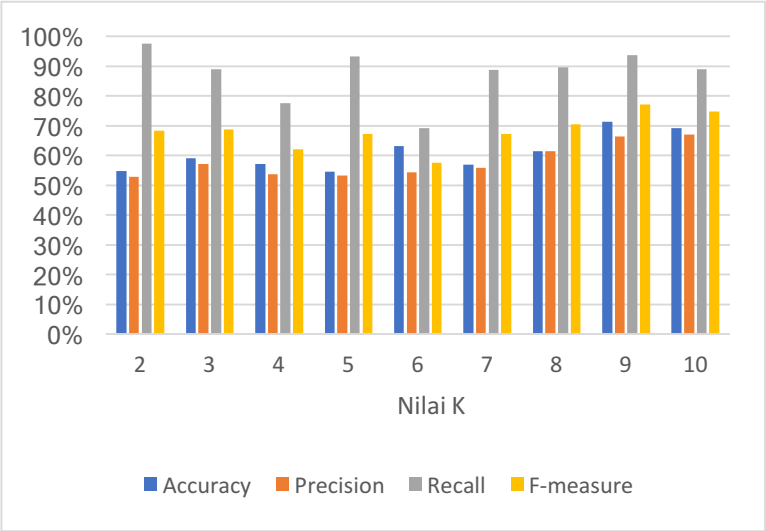
Nilai K	Accuracy	Precision	Recall	F-measure
2	62,8%	67,6%	55,6%	59,8%
3	73,2%	68,4%	90,8%	77,2%
4	81,4%	83,5%	86,3%	82,9%
5	76,8%	77,6%	81,4%	78,0%
6	**82,5%**	**82,7%**	**83,5%**	**82,4%**
7	78,0%	74,8%	86,7%	80,1%
8	81,6%	81,4%	87,5%	82,7%
9	78,0%	77,4%	82,0%	78,5%
10	81,5%	81,3%	84,0%	81,7%

Tabel 5. 6 Hasil perhitungan menggunakan fitur gabungan *POS Tagging* dan *Term Presence* dengan *kernel RBF*

Nilai K	Accuracy	Precision	Recall	F-measure
2	54,7%	52,8%	97,7%	68,4%
3	59,2%	57,2%	88,9%	68,7%
4	57,1%	53,8%	77,7%	62,1%
5	54,5%	53,2%	93,3%	67,3%
6	63,2%	54,3%	69,3%	57,5%
7	57,0%	55,8%	88,8%	67,3%
8	61,5%	61,5%	89,6%	70,4%
9	**71,4%**	**66,3%**	**93,7%**	**77,1%**
10	69,3%	67,0%	89,0%	74,9%



Gambar 5. 5 Plot data menggunakan fitur gabungan *POS Tagging* dan *Term Presence* dengan *kernel linear*



Gambar 5. 6 Plot data menggunakan fitur gabungan *POS Tagging* dan *Term Presence* dengan *kernel RBF*

Setelah mendapatkan hasil pengujian dari *kernel linear* dan *RBF,* maka dibandingan kedua *kernel* antara *linear dan RBF*. Berikut hasil perbandingan dari kedua *kernel* ditunjukkan pada Tabel 5.7, Tabel 5.8, dan Tabel 5.9.

Tabel 5. 7 Hasil perbandingan *kernel* menggunakan fitur gabungan *POS Tagging* dan *Term Presence*

Kernel	Accuracy	Precision	Recall	F-measure
Linear	**82,5%**	**82,7%**	**83,5%**	**82,4%**
RBF	71,4%	66,3%	93,7%	77,1%

Tabel 5. 8 Hasil perbadingan *kernel* menggunakan fitur *POS Tagging*

Kernel	Accuracy	Precision	Recall	F-measure
Linear	**63,0%**	**61,7%**	**76,9%**	**68,0%**
RBF	60,5%	57,4%	81,2%	67,0%

Tabel 5. 9 Hasil perbadingan *kernel* menggunakan fitur *Term Presence*

Kernel	Accuracy	Precision	Recall	F-measure
Linear	**77,0%**	**74,0%**	**87,0%**	**79,0%**
RBF	74,7%	70,8%	93,0%	79,2%

5.4 Evaluasi

Pengujian 1 diperoleh bahwa nilai terbaik untuk *accuracy, recall, precision,* dan *f-measure* didapat pada saat menggunakan nilai $K = 5$ untuk *kernel linear* dan $K = 2$ untuk *kernel RBF*. Dengan menggunakan fitur dari kata *POS Tagging* dari *corpus SentiWordNet* memberikan hasil masih belum cukup memuaskan. Hal ini disebabkan karena jumlah kata yang sesuai dengan *corpus* tidak banyak. Terdapat maksimal 8 fitur kata yang digunakan pada *tweet* data. Selain jumlah fitur yang sedikit, terdapat beberapa *tweet* yang tidak memiliki fitur karena jenis kata yang tidak dapat dikenali oleh *corpus*.

Pada pengujian 2 diperoleh bahwa hasil terbaik *accuracy* didapat ketika menggunakan nilai $K = 9$ untuk *kernel linear* dan $K = 10$ untuk *kernel RBF*. Dengan menggunakan fitur *Term Presence* pada dataset didapatkan 100 fitur TP. Memiliki jumlah matriks yang lebih besar dari pengujian 1, membuat hasil penggunaan fitur TP lebih baik. Namun fitur yang didapat pada pengujian ini belum cukup memuaskan. Masih banyak kata yang tidak dikenali oleh *corpus* kamus *SentiWordnet* karena penggunaan kata yang kurang baku.

Pada pengujian ke 3 diperoleh hasil terbaik *accuracy* didapat ketika menggunakan nilai *K = 6* untuk *kernel linear* dan *K = 9* untuk *kernel RBF*. Dengan menggunakan fitur gabungan antara *POS Tagging* dan *TP*, memberikan hasil yang lebih baik dibandingkan menggunakan fitur secara terpisah. Dengan menggunakan fitur gabungan, maka ukuran matriks input pada dataset semakin besar. Hal tersebut akan membuat proses perhitungan pada proses klasifikasi semakin akurat.

Namun masih terdapat beberapa kesalahan dalam melakukan klasifikasi dimana prediksi *tweet* yang tidak sesuai. Hal ini dikarenakan proses *preprocessing* yang kurang akurat. Banyak kata yang belum dapat dirubah menjadi kata baku dan tidak terbaca oleh *SentiWordNet*. Sehingga ada beberapa *tweet* yang tidak memiliki bobot sama sekali.

Setelah mendapatkan hasil dari pengujian 1 sampai 3, maka dilakukan perbandingan performa antara *kernel linear* dan *kernel RBF* dengan nilai terbaik. Dari perbandingan pada Tabel 5.7, 5.8, dan 5.9 tersebut diperoleh bahwa penggunaan parameter *kernel linear* pada metode *IMS* dan fitur gabungan antara *POS Tagging* dan *TP* sebagai fitur total pada kasus ini memberikan hasil terbaik. Penggunaan fitur gabungan memberikan hasil yang terbaik karena semakin banyak jumlah fitur pada proses *text mining* khususnya pada analisis sentimen akan memberikan hasil yang semakin baik. Selain penggunaan fitur gabungan, penggunaan *kernel linear* pada kasus ini memberikan hasil yang terbaik karena jumlah kelas data yang hanya dua dan dimensi metrik yang tidak terlalu besar. Dengan menggunakan parameter tersebut maka didapatkan hasil klasifikasi yang baik.

BAB VI
KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diperoleh selama pengerjaan tugas akhir dan saran mengenai pengembangan yang dapat dilakukan terhadap tugas akhir ini di masa yang akan datang

6.1. Kesimpulan

Dari hasil pengamatan selama proses perancangan, implementasi, dan pengujian yang dilakukan, dapat diambil kesimpulan sebagai berikut.

1. Pada data *tweet* melalui akun *twitter* XL, Telkomsel, dan IM3 pada bulan April 2016 dengan menggunakan fitur *POS Tagging,* hasil terbaik didapatkan menggunakan *kernel linear* dengan nilai $K = 5$ yaitu *accuracy* sebesar 63,0%, *precision* 61,7%, *recall* 76,9%, dan *f-measure* 68,0%. Sedangkan hasil terbaik *kernel RBF* didapat dengan nilai $K = 2$ dengan nilai *accuracy sebesar 60,5%, precision 57,4%, recall 81,2%, dan f-measure 67,0%.*
2. Pada data *tweet* melalui akun *twitter* XL, Telkomsel, dan IM3 pada bulan April 2016 dengan menggunakan fitur *Term Presence (TP),* hasil terbaik didapatkan menggunakan *kernel linear* dengan nilai $K = 9$ yaitu *accuracy* sebesar 77,0%, *precision* 74,0%*, recall* 87,0%, dan *f-measure* 79,0%. Sedangkan hasil terbaik *kernel RBF* didapat dengan nilai $K = 10$ dengan nilai *accuracy* 74,7%, *precision* 70,8%, *recall* 93,0%, *dan f-measure* 79,2%.
3. Pada data *tweet* melalui akun *twitter* XL, Telkomsel, dan IM3 pada bulan April 2016 dengan menggunakan fitur gabungan antara *Term Presence (TP)* dan *POS Tagging,* hasil terbaik didapatkan menggunakan *kernel linear* dengan nilai $K = 6$ yaitu *accuracy* sebesar 82,5%,

precision 82,7%, *recall* 83,5%, dan *f-measure* 82,4%. Sedangkan hasil terbaik *kernel RBF* didapat dengan nilai K = 9 dengan nilai *accuracy* sebesar 71,4%, *precision* 66,3%, *recall* 93,7%, dan *f-measure* 77,1%.

4. Penggunaan fitur gabungan antara *Term Presence (TP)* dan *POS Tagging* terbukti memberikan hasil lebih baik dibandingkan penggunaan fitur secara terpisah pada kasus *tweet* akun XL, Telkomsel, dan IM3 pada bulan April 2016.
5. Pengujian menggunakan parameter *kernel linear* pada SVM memberikan hasil yang lebih baik daripada penggunaan *kernel RBF* pada dataset yang digunakan.
6. Berdasarkan hasil uji coba yang telah didapat, dapat disimpulkan bahwa SWIMS dapat diimplementasikan untuk *dataset* Berbahasa Indonesia.

6.2. Saran

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang. Saran-saran ini didasarkan pada hasil perancangan, implementasi, dan pengujian yang telah dilakukan.

1. Selain menggunakan *kernel linear* dan *kernel RBF* dapat digunakan jenis *kernel* lain.
2. Menggunakan *corpus* Bahasa Indonesia lain untuk mendapatkan nilai bobot kata yang akan dijadikan sebagai fitur.
3. Menerapkan algoritma selain SVM ketika melakukan klasifikasi.
4. Melakukan *pre-processing* secara *detail* pada data agar memberikan arti kata yang lebih akurat dan dapat disesuaikan dengan *corpus* yang digunakan.

DAFTAR PUSTAKA

[1] Lee, B. Pang and Lillian, "Opinion Mining and Sentiment Analysis," *Foundations and Trends® in Information Retrieval,* vol. 2, no. 1-2, pp. 1-135, 2008.

[2] Korashy, W. Medhat and A. Hassan, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal,* vol. 5, no. 4, pp. 1093 - 1113, 2014.

[3] J. Bhaskar, K. Sruthi and P. Nedungadi, "Hybrid Approach for Emotion Classification of Audio Conversation Based on Text and Speech Mining," *Procedia Computer Science,* vol. 46, pp. 635-643, 2015.

[4] Li, J. Pan, X. Hu, Y. Zhang, P. Li, Y. Lin, H. Li, W. He and Lei, "Quadruple Transfer Learning: Exploiting both shared and non-shared concepts for text classification," *Knowledge-Based Systems,* vol. 90, pp. 199-210, 2015.

[5] Bashir, F. H. Khan, U. Qamar and Saba, "SWIMS: Semi-supervised subjective feature weighting and intelligent model selection for sentiment analysis," *Knowledge-Based Systems,* vol. 100, pp. 97 - 111, 2016.

[6] J. Brownlee, "Machine Learning Mastery," 10 2014. [Online]. Available: https://machinelearningmastery.com/an-introduction-to-feature-selection/. [Accessed 2 1 2017].

[7] D. Moeljadi, "Indonesian SentiWordNet," 2016. [Online]. Available: https://github.com/neocl/barasa. [Accessed 29 12 2017].

[8] M. Budi Wahyono, "Macam-Macam Kelas Kata," 04 06 2018. [Online]. Available: https://erlangga.co.id/materi-belajar/sma/8792-macam-macam-kelas-kata.html.

[9] N. Bambrick, "KDnuggets," 2016. [Online]. Available: https://www.kdnuggets.com/2016/07/support-vector-machines-simple-explanation.html. [Accessed 2 1 2017].

[10] S. Ray, "analytics vidhya," [Online]. Available: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/. [Accessed 03 06 2018].

[11] R. Xia, C. Zong and S. Lo, "Ensemble of feature sets and classification algorithms for sentiment classification," *Information Sciences,* vol. 181, pp. 1138-1152, 2011.

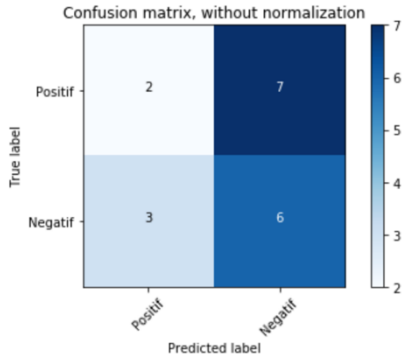
[12] P. Gupta, "Towards Data Science," [Online]. Available: https://towardsdatascience.com/cross-validation-in-machine-learning-72924a69872f. [Accessed 22 05 2018].

LAMPIRAN

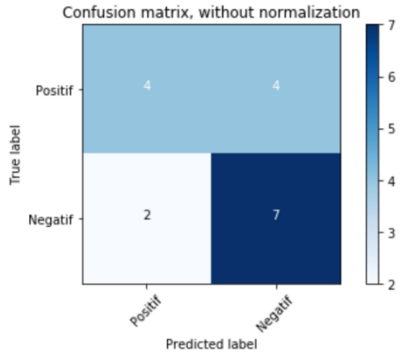
Confussion matrix pada setiap *cross validation* dengan hasil terbaik pada setiap skenario pengujian.

1. Pengujian 1 Menggunakan Fitur *POS Tagging*

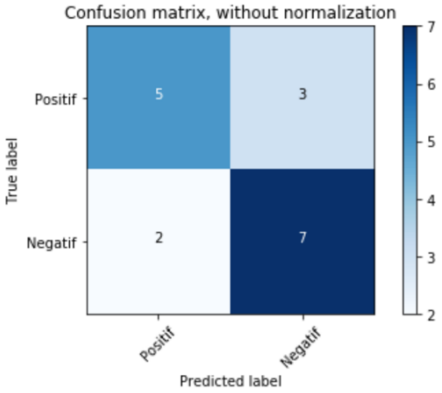
```
Best Final Accuracy on 1 CV: 0.444
```



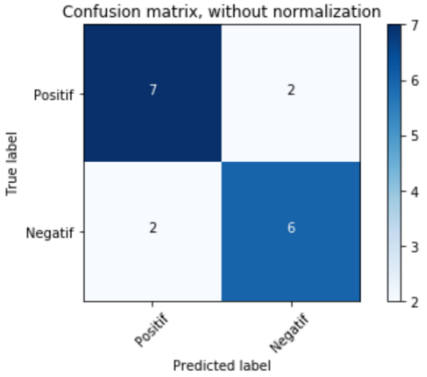

```
Best Final Accuracy on 2 CV: 0.647
```



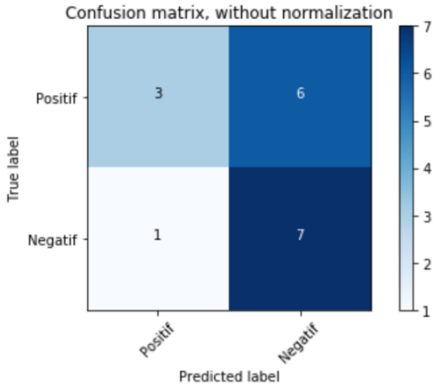
```
Best Final Accuracy on 3 CV: 0.706
```




```
Best Final Accuracy on 4 CV: 0.765
```

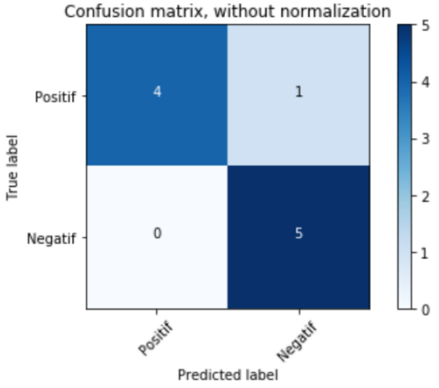


```
Best Final Accuracy on 5 CV: 0.588
```

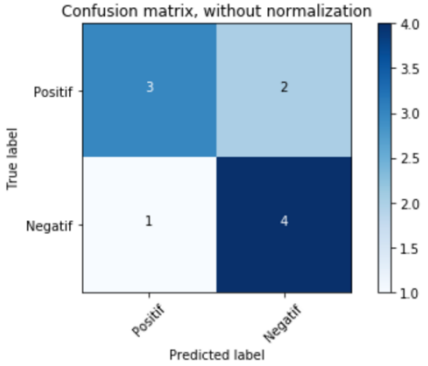


2. Pengujian 2 Menggunakan Fitur *Term Presence*

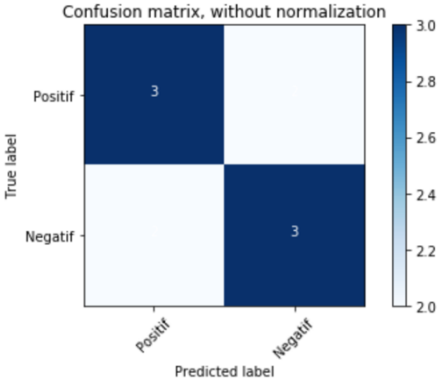
```
Best Final Accuracy on 1 CV: 0.900
```



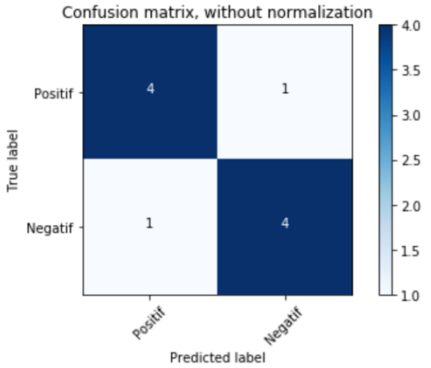
```
Best Final Accuracy on 2 CV: 0.700
```



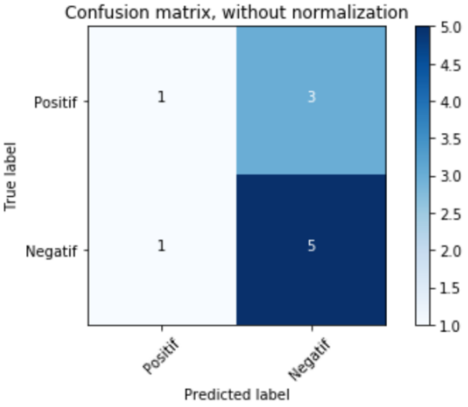

```
Best Final Accuracy on 3 CV: 0.600
```



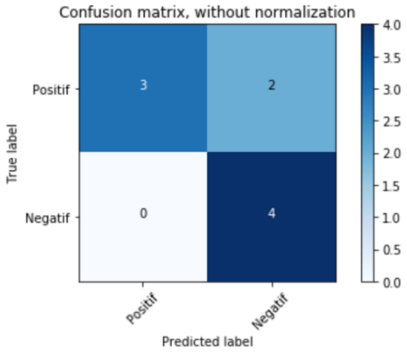
```
Best Final Accuracy on 4 CV: 0.800
```



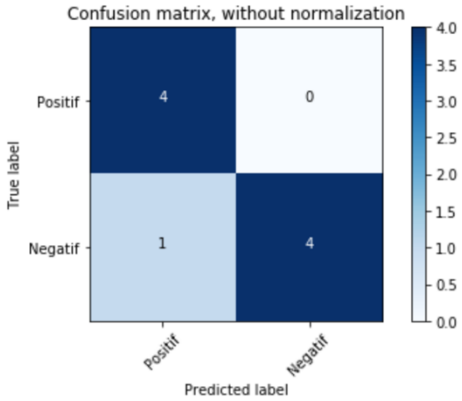

```
Best Final Accuracy on 5 CV: 0.600
```



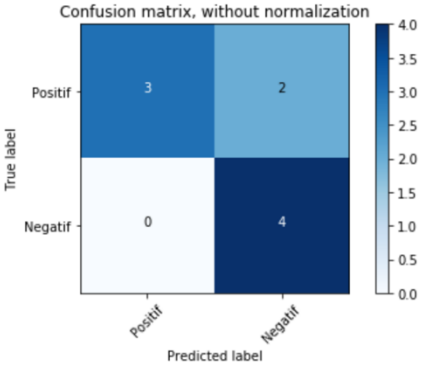
```
Best Final Accuracy on 6 CV: 0.778
```



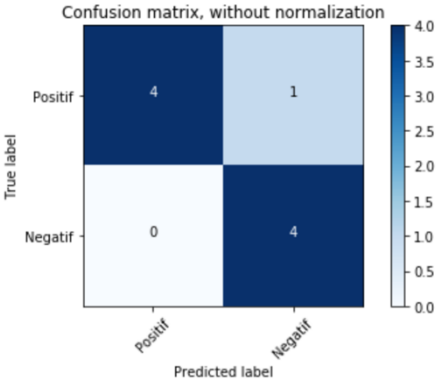

```
Best Final Accuracy on 7 CV: 0.889
```



```
Best Final Accuracy on 8 CV: 0.778
```

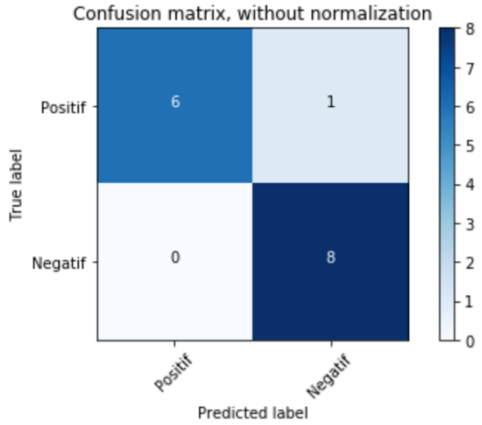



```
Best Final Accuracy on 9 CV: 0.889
```

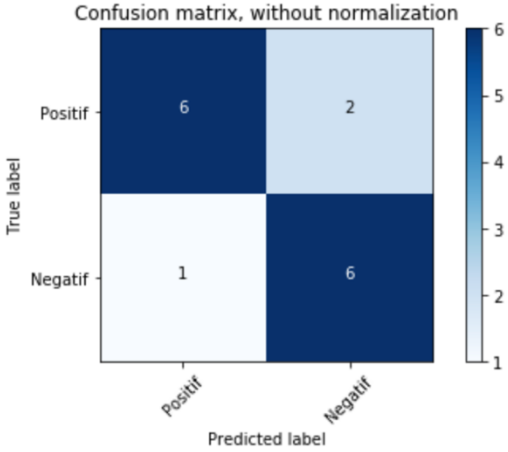


3. Pengujian 3 Menggunakan Fitur Gabungan

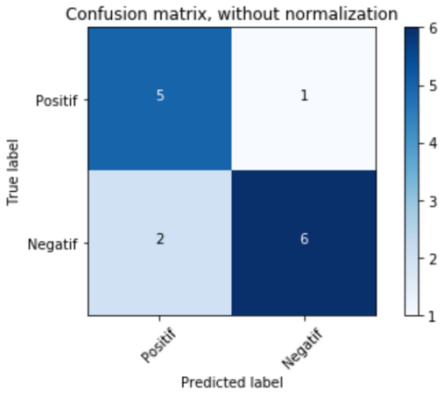
```
Best Final Accuracy on 1 CV: 0.933
```



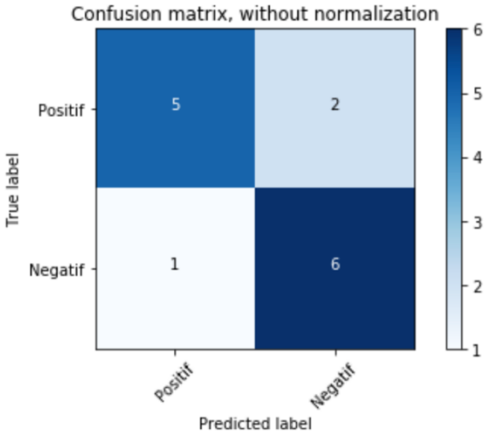

```
Best Final Accuracy on 2 CV: 0.800
```



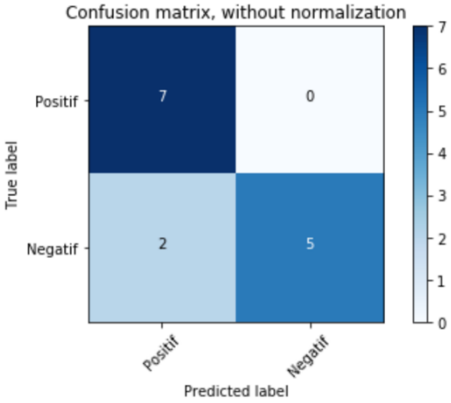
```
Best Final Accuracy on 3 CV: 0.786
```



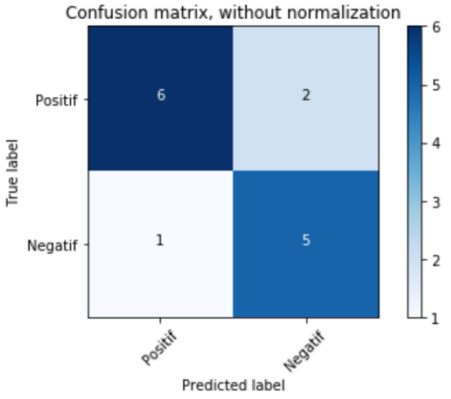

```
Best Final Accuracy on 4 CV: 0.786
```



```
Best Final Accuracy on 5 CV: 0.857
```




```
Best Final Accuracy on 6 CV: 0.786
```



Hasil performa pada setiap iterasi *k-fold* pada masing-masing skenario

1. Hasil performa pada Skenario 1. Penggunaan fitur *POS Tagging* dengan menggunakan *kernel linear*

	Accuracy	**Recall**	**Precision**	**F-measure**
Nilai K				
2	0,418	0,59	0,448	0,509
	0,558	0,952	0,526	0,677
3	0,448	0,733	0,478	0,578
	0,517	0,866	0,52	0,65
	0,392	0,692	0,409	0,514
4	0,222	0,454	0,4	0,4
	0,409	0,666	0,551	0,551
	0,571	1	0,689	0,689
	0,523	0,9	0,642	0,642
5	0,444	0,666	0,461	0,545
	0,647	0,777	0,636	0,7
	0,705	0,777	0,7	0,736
	0,764	0,75	0,75	0,75
	0,588	0,875	0,538	0,666
6	0,466	0,625	0,5	0,555
	0,4	0,857	0,428	0,571
	0,642	0,625	0,714	0,666
	0,5	0,714	0,5	0,588
	0,428	0	0	0
	0,35	0,833	0,384	0,526

7	0,538	0,538	0,555	0,625
	0,307	0,307	0,363	0,47
	0,333	0,333	0,375	0,428
	0,5	0,5	0,555	0,625
	0,666	0,666	0,666	0,666
	0,58	0,58	1	0,285
	0,333	0,333	0,363	0,5
8	0,636	0,833	0,625	0,714
	0,09	0,2	0,142	0,166
	0,545	0,8333	0,555	0,666
	0,636	0,666	0,666	0,666
	0,727	0,4	1	0,571
	0,54	0,5	0,6	0,545
	0,6	0,6	0,6	0,6
	0,3	0,75	0,333	0,461
9	0,6	0,8	0,571	0,666
	0,3	0,4	0,333	0,363
	0,4	0,8	0,444	0,571
	0,6	0,6	0,6	0,6
	0,5	0,5	0,6	0,545
	0,77	0,75	0,75	0,75
	0,555	0,2	1	0,333
	0,666	1	0,571	0,727
	0,444	0,75	0,428	0,545
10	0,666	0,8	0,666	0,727
	0,222	0,5	0,285	0,363
	0,555	0,4	0,666	0,5

	0,333	0,5	0,333	0,4
	0,555	0,6	0,6	0,6
	0,666	0,6	0,75	0,666
	0,625	0,75	0,6	0,666
	0,75	1	0,666	0,8
	0,5	1	0,428	0,6
	0,375	0,25	0,333	0,285

2. Hasil performa pada Skenario 1. Penggunaan fitur *POS Tagging* dengan menggunakan *kernel RBF*

	Accuracy	**Recall**	**Precision**	**F-measure**
Nilai K				
2	0,604	0,909	0,571	0,571
	0,604	0,714	0,576	0,638
3	0,413	0,666	0,454	0,54
	0,379	0,666	0,434	0,526
	0,464	1	0,464	0,634
4	0,454	0,727	0,47	0,571
	0,454	0	0	0
	0,476	1	0,476	0,645
	0,52	1	0,5	0,666
5	0,444	0,666	0,416	0,545
	0,529	0,777	0,538	0,636
	0,47	0	0	0
	0,529	1	0,5	0,666
	0,411	0,75	0,428	0,545
6	0,333	0,125	0,25	0,166

	0,466	1	0,466	0,636
	0,425	0	0	0
	0,785	0,714	0,833	0,769
	0,428	0,428	0,428	0,42
	0,42	0,666	0,666	0,5
7	0,307	0,142	0,25	0,181
	0,307	0,666	0,363	0,47
	0,25	0,5	0,333	0,4
	0,416	0	0	0
	0,548	1	0,545	0,705
	0,5	1	0,5	0,666
	0,333	0,8	0,363	0,5
8	0,636	0,833	0,625	0,714
	0,272	0,4	0,285	0,333
	0,545	0,333	0,666	0,444
	0,454	0	0	0
	0,545	0,8	0,5	0,615
	0,54	0	0	0
	0,7	1	0,625	0,769
	0,4	1	0,4	0,571
9	0,6	0,8	0,571	0,666
	0,3	0,6	0,375	0,461
	0,4	0,8	0,444	0,571
	0,3	0,6	0,375	0,461
	0,6	0,333	1	0,5
	0,444	0,75	0,428	0,545
	0,444	0	0	0

	0,444	1	0,444	0,615
	0,444	0,5	0,4	0,444
10	0,444	0,2	0,5	0,285
	0,333	0,75	0,375	0,5
	0,555	0,6	0,6	0,6
	0,333	0,5	0,333	0,4
	0,666	0,4	1	0,571
	0,666	0,4	1	0,571
	0,75	0,5	1	0,666
	0,625	0,5	0,666	0,571
	0,375	1	0,375	0,545
	0,375	0,5	0,4	0,444

3. Hasil performa pada Skenario 2. Penggunaan fitur *Term Presence* dengan menggunakan *kernel linear*

	Accuracy	**Recall**	**Precision**	**F-measure**
Nilai K				
2	0,651	0,909	0,606	0,727
	0,813	1	0,724	0,84
3	0,689	0,8	0,666	0,504
	0,72	0,8	0,705	0,75
	0,857	1	0,764	0,866
4	0,727	0,818	0,692	0,75
	0,727	0,666	0,8	0,727
	0,666	0,9	0,6	0,72
	0,857	1	0,769	0,869
5	0,833	0,888	0,8	0,842

	0,705	0,777	0,7	0,736
	0,588	0,666	0,6	0,631
	0,764	0,875	0,7	0,777
	0,823	1	0,727	0,842
6	0,8	0,875	0,777	0,823
	0,6	0,714	0,555	0,625
	0,714	0,625	0,833	0,714
	0,714	0,857	0,666	0,75
	0,785	0,857	0,75	0,8
	0,85	1	0,75	0,857
7	0,846	1	0,539	0,875
	0,538	0,666	0,5	0,571
	0,666	0,833	0,625	0,714
	0,416	0,571	0,5	0,533
	0,75	0,833	0,714	0,769
	0,833	0,833	0,833	0,833
	0,833	1	0,714	0,833
8	0,909	1	0,857	0,923
	0,636	0,8	0,571	0,666
	0,727	0,666	0,8	0,727
	0,636	0,5	0,75	0,6
	0,636	1	0,555	0,714
	0,72	0,666	0,8	0,727
	0,8	1	0,714	0,833
	0,9	1	0,8	0,888
9	0,9	1	0,833	0,909
	0,7	0,8	0,666	0,727

	0,6	0,6	0,6	0,6
	0,8	0,8	0,8	0,8
	0,6	0,833	0,625	0,71
	0,77	1	0,666	0,8
	0,88	0,8	1	0,888
	0,77	1	0,666	0,8
	0,88	1	0,8	0,888
10	0,888	1	0,833	0,909
	0,666	0,75	0,6	0,666
	0,666	0,8	0,666	0,727
	0,888	0,75	1	0,857
	0,444	0,4	0,5	0,444
	0,77	0,8	0,8	0,8
	0,75	1	0,666	0,8
	0,875	0,75	1	0,857
	0,75	1	0,6	0,75
	0,875	1	0,8	0,888

4. Hasil performa pada Skenario 2. Penggunaan fitur *Term Presence* dengan menggunakan *kernel RBF*

	Accuracy	**Recall**	**Precision**	**F-measure**
Nilai K				
2	0,697	0,772	0,68	0,723
	0,558	0,238	0,625	0,344
3	0,689	0,6	0,75	0,666
	0,689	0,666	0,714	0,689
	0,464	1	0,464	0,634

4	0,636	0,545	0,666	0,6
	0,636	0,833	0,625	0,714
	0,714	1	0,625	0,769
	0,476	1	0,476	0,645
5	0,777	1	0,629	0,818
	0,529	0,888	0,533	0,666
	0,647	0,777	0,636	0,7
	0,47	1	0,47	0,64
	0,47	1	0,47	0,64
6	0,666	1	0,615	0,761
	0,466	1	0,466	0,636
	0,785	0,75	0,857	0,8
	0,642	1	0,583	0,736
	0,857	1	0,777	0,875
	0,92	1	0,857	0,923
7	0,692	1	0,636	0,777
	0,538	0,833	0,5	0,625
	0,5	0,833	0,5	0,625
	0,5	0,714	0,555	0,625
	0,833	1	0,75	0,857
	0,666	0,833	0,625	0,714
	0,416	1	0,416	0,588
8	0,9	1	0,857	0,927
	0,545	1	0,5	0,666
	0,545	0,833	0,555	0,666
	0,727	0,666	0,8	0,727
	0,727	1	0,8	0,769

	0,81	0,833	0,833	0,833
	0,8	0,8	0,8	0,8
	0,4	1	0,4	0,57
9	0,9	1	0,833	0,909
	0,6	1	0,555	0,714
	0,4	0,8	0,444	0,571
	0,7	0,8	0,666	0,727
	0,6	0,833	0,625	0,714
	0,77	1	0,666	0,8
	0,777	1	0,714	0,833
	0,777	0,75	0,75	0,75
	0,888	1	0,8	0,888
10	0,888	1	0,833	0,909
	0,666	1	0,571	0,727
	0,555	1	0,555	0,714
	0,555	0,75	0,5	0,6
	0,666	0,8	0,666	0,727
	0,88	1	0,833	0,909
	0,625	1	0,571	0,727
	0,875	0,75	1	0,857
	0,75	1	0,75	0,857
	0,875	1	0,8	0,888

5. Hasil performa pada Skenario 3. Penggunan fitur gabungan *POS Tagging* dan *Term Presence* menggunakan *kernel linear*

	Accuracy	**Recall**	**Precision**	**F-measure**
Nilai K				
2	0,581	0,636	0,583	0,608
	0,674	0,476	0,769	0,588
3	0,724	1	0,652	0,789
	0,793	0,8	0,8	0,8
	0,678	0,923	0,6	0,727
4	0,727	1	0,647	0,785
	0,863	0,75	1	0,857
	0,761	0,9	0,692	0,782
	0,904	0,8	1	0,888
5	0,722	0,888	0,666	0,761
	0,764	0,777	0,777	0,777
	0,647	0,777	0,636	0,7
	0,823	0,625	1	0,769
	0,882	1	0,8	0,888
6	0,933	1	0,888	0,941
	0,8	0,857	0,75	0,8
	0,786	0,75	0,857	0,8
	0,786	0,875	0,75	0,8
	0,857	0,713	1	0,833
	0,786	0,833	0,714	0,769
7	0,769	0,857	0,75	0,8

	0,692	0,666	0,666	0,666
	0,916	1	0,857	0,923
	0,5	0,714	0,555	0,625
	0,915	1	0,857	0,923
	0,833	0,833	0,833	0,833
	0,833	1	0,714	0,833
8	0,909	1	0,857	0,923
	0,727	1	0,625	0,769
	0,727	0,666	0,8	0,727
	0,818	0,666	1	0,8
	0,818	1	0,714	0,833
	0,727	0,666	0,8	0,727
	0,8	1	0,714	0,833
	1	1	1	1
9	0,9	1	0,833	0,909
	0,7	0,8	0,666	0,727
	0,7	0,6	0,75	0,666
	0,8	0,8	0,8	0,8
	0,7	0,833	0,714	0,769
	0,666	0,75	0,6	0,666
	0,777	0,6	1	0,75
	0,888	1	0,8	0,888
	0,888	1	0,8	0,888
10	0,888	1	0,833	0,909
	0,777	0,75	0,75	0,75
	0,666	0,8	0,666	0,727
	0,888	0,75	1	0,857

	0,666	0,6	0,75	0,666
	0,888	1	0,833	0,909
	0,75	0,75	0,75	0,75
	0,875	0,75	1	0,857
	0,875	1	0,75	0,857
	0,875	1	0,8	0,888

6. Hasil performa pada Skenario 3. Penggunan fitur gabungan *POS Tagging* dan *Term Presence* menggunakan *kernel RBF*

	Accuracy	**Recall**	**Precision**	**F-measure**
Nilai K				
2	0,604	0,954	0,576	0,711
	0,488	1	0,488	0,656
3	0,655	0,8	0,631	0,705
	0,655	0,866	0,619	0,722
	0,464	1	0,464	0,634
4	0,409	0,272	0,375	0,315
	0,59	0,833	0,588	0,689
	0,809	1	0,714	0,833
	0,476	1	0,476	0,645
5	0,666	1	0,6	0,75
	0,529	0,888	0,533	0,666
	0,588	0,77	0,583	0,666
	0,47	1	0,47	0,64
	0,47	1	0,47	0,64
6	0,466	0	0	0

	0,4666	1	0,466	0,636
	0,857	0,875	0,875	0,875
	0,642	1	0,583	0,736
	0,571	0,285	0,666	0,4
	0,785	1	0,666	0,8
7	0,692	1	0,636	0,777
	0,461	1	0,461	0,631
	0,5	0,833	0,5	0,625
	0,5	0,714	0,555	0,625
	0,75	1	0,666	0,8
	0,666	0,666	0,666	0,666
	0,416	1	0,416	0,588
8	0,636	0,833	0,625	0,714
	0,545	1	0,5	0,666
	0,454	0,833	0,5	0,625
	0,727	0,833	0,714	0,769
	0,636	1	0,555	0,714
	0,81	0,666	1	0,8
	0,7	1		0,769
	0,4	1	0,625	0,571
9	0,8	1	0,714	0,833
	0,6	1	0,555	0,714
	0,4	0,8	0,444	0,571
	0,7	0,8	0,666	0,727
	0,7	0,833	0,714	0,769
	0,666	1	0,571	0,714
	0,888	1	0,8333	0,909

	0,888	1	0,8	0,888
	0,777	1	0,666	0,8
10	0,555	0,6	0,6	0,6
	0,444	1	0,444	0,615
	0,555	1	0,555	0,714
	0,555	0,75	0,5	0,6
	0,666	0,8	0,666	0,727
	0,777	1	0,714	0,8333
	0,75	1	0,666	0,8
	0,875	0,75	1	0,857
	0,875	1	0,75	0,857
	0,875	1	0,8	0,888

BIODATA PENULIS

Adam Widi Bagaskarta, lahir di Malang pada tanggal 14 September 1995. Lulus dari SMAN 1 Blitar pada tahun 2014 dan melanjutkan studi di Departemen Informatika Institut Teknologi Sepuluh Nopember Surabaya. Berpengalaman sebagai asisten dosen pada matakuliah sistem basis data. Mengikuti beberapa kompetisi sebagai finalis pada hackathon BCA dan BRI pada tahun 2016. Aktif mengikuti organisasi antara lain staff kewirausahaan Himpunan Mahasiswa Teknik Computer-Informatika (HMTC) 2015/2016, staff Riset dan Teknologi Badan Eksekutif Mahasiswa ITS 2015/2016, menteri Riset dan Teknologi Badan Eksekutif Mahasiwa ITS 2016/2017, dan trainer keilmiahan ITS angkatan ke 6. Pernah mengikuti kegiatan *study aboard* dalam acara *lab internship* Shibaura Institute of Technology Japan.

Dalam menyelesaikan pendidikan sarjana, penulis mengambil bidang minat Komputer Visi dan Cerdas (KCV) dan juga memiliki ketertarikan di bidang Data Science dan pemrograman. Penulis dapat dihubungi melalui alamat email adamwbagaskarta@gmail.com dan hello@adambagaskarta.com.