KLASIFIKASI KEMACETAN LALU LINTAS DI KOTA MALANG PADA SOSIAL MEDIA TWITTER MENGGUNAKAN METODE *IMPROVED K-NEAREST NEIGHBOR*

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Riska Dewi Nurfarida
NIM: 145150201111138

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

KLASIFIKASI KEMACETAN LALU LINTAS DI KOTA MALANG PADA SOSIAL MEDIA TWITTER MENGGUNAKAN METODE *IMPROVED K-NEAREST NEIGHBOR*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Riska Dewi Nurfarida
NIM: 145150201111138

Skripsi ini telah diuji dan dinyatakan lulus pada
17 Oktober 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Indriati, S.T, M.Kom	Rizal Setya Perdana, S.Kom, M.Kom
NIP: 19831013 201504 2 002	NIK: 201603 910118 1 001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 17 Oktober 2018

Riska Dewi Nurfarida

NIM: 145150201111138

KATA PENGANTAR

Puji syukur atas kehadirat Allah SWT atas segala karunianya yang telah melimpahkan rahmat, taufik, dan hidayah-Nya sehingga laporan penelitian skripsi ini yang berjudul "Klasifikasi Kemacetan Lalu Lintas Di Kota Malang Pada Sosial Media Twitter Menggunakan Metode *Improved K-Nearest Neighbor*" dapat terselesaikan dengan baik.

Melalui kesempatan ini, penulis menyadari penulisan skripsi ini tidak akan dapat terselesaikan jika tanpa bantuan dari berbagai pihak. Oleh sebab itu, penulis ingin menyampaikan rasa terimakasih dan hormat yang sebesar-besarnya kepada segala pihak yang telah mendukung, memberikan bantuan, serta doa selama proses penulisan skripsi, diantaranya:

1. Ibu Indriati, S.T, M.Kom dan Bapak Rizal Setya Perdana, S.Kom, M.Kom selaku dosen pembimbing skripsi yang telah membimbing dan mengarahkan penulis dengan sabar sehingga penelitian skripsi ini dapat terselesaikan
2. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D., Bapak Ir. Heru Nurwarsito, M.Kom, Bapak Drs. Marji, M.T, dan Bapak Edy Santoso, S.Si, M.Kom selaku Dekan, Wakil Dekan I, Wakil Dekan II dan Wakil Dekan III Fakultas Ilmu Komputer Universitas Brawijaya.
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D, Bapak Agus Wahyu Widodo, S.T, M.Cs dan Bapak Muhammad Tanzil Furqon, S.Kom, M.CompSc selaku Ketua Jurusan, Ketua Program Studi dan Sekretaris Program Studi Teknik Informatika.
4. Bapak Subarkah dan Almarhumah Ibu Rismiyati dan Ibu Marsini yang telah memberikan motivasi, dukungan, kasih sayang, perhatian, serta senantiasa tiada hentinya memberikan doa demi kelancaran dan terselesaikannya skripsi ini.
5. Saudara-saudara saya, Elli Retno Ambarwati, Dwi Astuti Wahyusari, Andrian Wahyu Setyo Aji, Dedi Cahyo Saputro, Vera Rakhmawati Nugraheni, Vina Umi Faiqoh dan Muhammad Rofi'ul Anam, serta seluruh keluarga besar tercinta yang selalu mendukung dan memberikan doa demi kelancaran skripsi ini.
6. Seluruh civitas akademika Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberikan bantuan dan dukungan selama penulis menempuh studi dan selama penyelesaian skripsi di Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
7. Teman-teman terdekat saya, Putu Amelia Vennanda W., Chandra Ayu A. P., Nurul Muslimah, dan Nana Nofiana yang telah membantu dan memberikan dukungan selama proses penyelesaian penelitian skripsi ini.

8. Teman-teman Teknik Informatika angkatan 2014 dan Komputasi Cerdas serta seluruh pihak yang telah membantu kelancaran penulisan skripsi yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa penyusunan skripsi ini masih memiliki banyak kekurangan, sehingga penulis membutuhkan adanya kritik maupun saran yang bersifat membangun. Akhir kata dari penulis, saya harap skripsi ini dapat memberikan manfaat bagi semua pihak yang menggunakannya.

Malang, 17 Oktober 2018

Penulis

nurfaridard@gmail.com

ABSTRAK

Twitter merupakan layanan jejaring sosial yang diminati banyak pengguna internet yang digunakan sebagai media komunikasi dan juga mendapatkan informasi. Banyak informasi yang bisa didapatkan dari Twitter yaitu berupa pertanyaan, opini atau komentar yang bersifat postitif maupun negatif. Dengan perkembangan teknologi tersebut masyarakat Kota Malang dapat mencari informasi bahkan bertukar informasi mengenai keadaan lalu lintas di Kota Malang melalui sosial media Twitter. Seperti yang terdapat pada akun Twitter @PuspitaFM, kita dapat mendapatkan informasi atau bertukar informasi mengenai keadaan lalu lintas yang ada di Kota Malang. Namun terdapat kerancuan dalam menentukan kategori manakah *tweet* tersebut, apakah *tweet* tersebut masuk kategori macet atau masuk kategori tidak macet. Sehingga, pada penelitian ini akan mengklasifikasikan kategori kemacetan lalu lintas berdasarkan pada *tweet* yang diharapkan dapat mempermudah dalam menentukan kategori kemacetan lalu lintas pada *twitter.* Untuk melakukan proses klasifikasi ini dilakukan beberapa proses yaitu dimulai dengan proses *prepocessing text* yang terdiri dari beberapa tahapan-tahapa, yaitu *cleansing, case folding,* tokenisasi, *filtering* dan *stemming*. Kemudian dilanjutkan dengan proses pembobotan (*term weighting*), normalisasi, *cosine similiarity* hingga proses klasifikasi yang mana digunakan metode *Improved K-Nearest Neighbor*. Hasil yang didapatkan dari pross klasifikasi tersebut didapatkan yaitu *recall* sebesar 0.42857, *precision* sebesar 0.71428, *f-measure* sebesar 0.53571 dan hasil akurasi sebesar 65.33%. Jumlah data latih yang digunakan adalah 600 dokumen dan data uji yang digunakan adalah 150 dokumen.

Kata kunci : *Information Retrieval,* Twitter, Kemacetan Lalu Lintas, *Improved K-Neares* Neighbor

ABSTRACT

Twitter is a social networking service that many internet users are interested in that is used as a communication medium and also gets information. A lot of information can be obtained from Twitter, in the form of positive or negative questions, opinions or comments. With the development of these technologies, the people of Malang City can seek information and even exchange information about the traffic conditions in Malang City through Twitter social media. As found on the @PuspitaFM Twitter account, we can get information or exchange information about the traffic conditions in Malang City. But there is confusion in determining which category the tweet is, whether the tweet is categorized as a traffic jam or in the non-jammed category. So, in this study we will classify traffic congestion categories based on tweets which are expected to make it easier to determine the category of traffic jams on twitter. To carry out this classification process several processes are carried out, namely starting with the prepocessing text process which consists of several stages, namely cleansing, case folding, tokenisation, filtering and stemming. Then proceed with the weighting process (term weighting), normalization, cosine similiarity to the classification process which is used the Improved K-Nearest Neighbor method. The results obtained from the classification process are obtained, namely recall of 0.42857, precision of 0.71428, f-measure of 0.53571 and accuracy of 65.33%. The amount of training data used is 600 documents and the test data used is 150 documents.

Key Words : Information Retrieval, Twitter, Traffic congestion, Improved K-Neares Neighbor

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB 1 PENDAHULUAN

1.1 Latar belakang

Kemacetan lalu lintas merupakan masalah yang tak kunjung usai. Kemacetan lalu lintas adalah masalah yang sering terjadi di kota-kota besar, begitu juga dengan Kota Malang. Kota Malang sendiri merupakan kota termacet ketiga di Indonesia setelah Kota Jakrta dan Kota Bandung (Ramadhiani, 2018). Penyebab terjadinya kemacetan di Kota Malang juga beragam diantaranya yaitu perbandingan ruas jalan dengan banyaknya kendaraan yang tidak seimbang, adanya kecelakaan lau lintas, meningkatnya pengguna kendaraan pribadi, serta *attitude* dari pengguna jalan itu sendiri. Seperti yang terdapat pada web suryamalang.tribunnews.com pada tanggal 31 Maret 2017, ruas jalan di Kota Malang tidak bertambah sejak tahun 2016 dan tidak adanya pembangunan jalan baru pada tahun 2017. Artinya, Kota Malang membutuhkan jalan baru untuk mengatasi kemacetan lalu lintas yang terjadi. Faktor lain penyebab kemacetan yang terjadi di Kota Malang adalah banyaknya mahasiswa di Kota Malang. Berdasarkan data yang terdapat pada web suryamalang.tribunnews.com terdapat 57 perguruan tinggi dengan ribun mahasiswa. Jadwal kuliah mereka yang tak teratur memuat jalan setiap saat dipenuhi mahasiswa. Hal tersebut juga menjadi penyebab terjadinya kemacetan di Kota Malang (Suryanik, 2017).

Perkembangan teknologi saat ini dapat membantu rutinitas kegiatan sehari-hari, diantaranya adalah memantau kemacetan lalu lintas melalui sosial media Twitter. Twitter merupakan layanan jejaring sosial yang diminati banyak pengguna internet yang digunakan sebagai media komunikasi dan juga mendapatkan informasi. Banyak informasi yang bisa didapatkan dari Twitter yaitu berupa pertanyaan, opini atau komentar yang bersifat postitif maupun negatif (Nurjanah, 2017). Dengan perkembangan teknologi tersebut masyarakat Kota Malang dapat mencari informasi bahkan bertukar informasi mengenai keadaan lalu lintas di Kota Malang melalui sosial media Twitter. Berdasarkan data yang diambil dari www.kominfo.go.id Indonesia menempati peringkat kelima pengguna Twitter terbesar di dunia dengan 19,5 juta pengguna. Dengan banyaknya pengguna tersebut besar kemungkinan semakin mudahnya akses dalam mendapatkan informasi atau bertukarnya informasi mengenai keadaan lalu lintas yang ada di Kota Malang. Seperti yang terdapat pada akun Twitter @PuspitaFM, kita dapat mendapatkan informasi atau bertukar informasi mengenai keadaan lalu lintas yang ada di Kota Malang. Setiap harinya akun @PuspitaFM akan memberi informasi keadaan lalu lintas di beberapa ruas jalan di Kota Malang. Kemudian para *follower* dari akun tersebut juga bisa memberi informasi tentang keadaan lalu lintas di sekitar mereka dengan mentweet dan memention akun @PuspitaFM yang kemudian akan di *retweet* untuk dibagikan ke *follower* akun tersebut. Dari akun tersebut diambil *tweet* yang berupa kemacetan lalu lintas yang kemudian akan diklasifikasikan mengggunakan metode *Improved K-Nearest Neighbor.*

Beragamnya jenis *tweet* yang terdapat pada akun tersebut terkadang memiliki jenis *tweet* yang ambigu dalam menentukan keterangan macet atau tidak macet, sehingga diperlukan adanya suatu penelitian untuk mengklasifikasikan *tweet* dalam menentukan kategori macet atau tidak macet. Terdapat berbagai macam metode dalam melakukan pengklasifikasian teks, adapun metode tersebut terbagi menjadi 3 kelompok yakni klasifikasi teks dengan dasar statistic (*Naïve Bayes*, KNN CCV, SVM, dsb), kemudian klasifikasi teks berdasar koneksi (*Artifical Neural Network*), serta klasifikasi teks dengan dasar *rule based* (Decision Tree). Pada penelitian Yang Yiming dan Xin Liu pada tahun 1999, metode klasifikasi berbasis statistik, terutama KNN dan SVM terbukti memiliki kinerja yang lebih baik dibandingkan lainnya (Ridok dan Latifah, 2015).

Dalam penelitian ini menggunakan metode *Improved K-Nearest Neighbor* yang merupakan modifikasi dari metode *K-Nearest Neighbor* (K-NN). Metode K-NN merupakan metode yang digunakan untuk mengelompokkan objek berdasarkan jarak terdekat dari objek masing-masing kategori (Sreemathy dan Balamurugan, 2012). Namun terdapat kekurangan dalam penerapan metode K-NN yaitu hasil yang diperoleh dalam penentuan kelas dari data kandidat hasil yang didapat masih kurang tepat, maka dengan adanya metode *Improved K-Nearest Neighbor* dapat menjadi solusi yang tepat untuk mengatasi masalah tersebut (Megantara et all, 2010). Perbedaan antara metode K-NN dan *Improved K-Nearest Neighbor* terdapat dalam penentuan nilai *k*, pada K-NN nilai *k* yang ditentukan pada tiap kategori ialah memiliki nilai yang sama, sedangkan pada *Improved K-Nearest Neighbor* digunakan nilai *k* yang berbeda pada tiap kategori yang sesuai dengan banyaknya data latih (Puspitasari et all, 2017). Sehingga nilai akurasi yang didapatkan akan lebih tinggi dan maksimal. Metode ini dirasa tepat untuk melakukan klasifikasi sehingga dapat menghasilkan kelas-kelas yang sesuai.

Penelitian yang menjadi acuan dalam penelitian ini yaitu memiliki judul "Klasifikasi Posting Twitter Kemacetan Lalu Lintas Kota Bandung Mengunakan *Naive Bayes Classification"* yang dilakukan oleh Sandi afajar Rodiansyah dan Edi Winarko pada tahun 2013. Penelitian tersebut dilakukan dengan metode Naive Bayes Classification karena metode tersebut merupakan algortima klasifikasi sederhana yang mempunyai nilai akurasi yang tinggi (Rodansyah dan Winarko, 2013). Berdasarkan penelitian tersebut hasil akurasi terkecil memiliki nilai 78% dengan sampel sebanyak 100 dan hasil akurasi terbesar memiliki nilai 91,60% dengan sebanyak 13160. Pada penelitian ini juga dilakukan dengan menggunakan metode *Support Vector Machine* (SVM) yang memiliki hasil akurasi terkecil 92% dengan sampel sebanyak 100 dan hasil tertinggi 99,11% dengan sampel sebanyak 13160.

Pada penelitian yang berjudul "Klasifikasi Spam Pada Twitter Menggunakan Metode *Improved K-Nearest Neighbor"* yang dilakukan oleh Dea Zakia Nathania, Indriarti dan Fitra Abdurrachman Bachtiar pada tahun 2018. Penelitian tersebut membandingkan kinerja dari metode KNN dan metode Improved KNN dalam implementasi klasifikasi spam pada twitter. Pada penelitian tersebut digunakan data latih sebanyak 500 dokumen. Hasil dari penelitian tersebut adalah metode

Improved KNN memiliki akurasi yang lebih tinggi dengan nilai akurasi 92% dibandingkan dengan metode KNN dengan nilai akurasi 88%.

Berdasarkan penelitian di atas, untuk menentukan kelas-kelas pada suatu kategori dilakukan beberapa proses. Pada penelitian ini akan dilakukan proses untuk mengetahui kelas-kelas pada kategori kemacetan lalu lintas di Kota Malang pada sosial media *Twitter*.

1.2 Rumusan masalah

Berdasarkan uraian di atas, maka dapat dirumuskan permasalahan-permasalahan yang ada pada skripsi adalah sebagai berikut:

1. Bagaimana penerapan metode Improved KNN dalam melakukan klasifikasi kemacetan di Kota Malang pada sosial media twitter?
2. Bagaimana hasil akurasi klasifikasi kemacetan lalu lintas kota Malang pada sosial media Twitter dengan metode Improved KNN ?

1.3 Tujuan

Adapun tujuan dari penelitian klasifikasi kemacetan di Kota Malang pada sosial media twitter menggunakan metode improved KNN adalah sebagai berikut:

1. Menerapkan metode Improved KNN ke dalam melakukan klasifikasi kemacetan lalu di Kota Malang pada sosial media twitter.
2. Mendapatkan hasil akurasi klasifikasi kemacetan lalu lintas kota Malang pada sosial media Twitter dengan metode Improved KNN.

1.4 Manfaat

Penelitian ini diharapkan memiliki manfaat yang baik serta berguna bagi pembaca dan penulis. Adapun manfaatnya adalah sebagai berikut:

Bagi Penulis

1. Sebagai media untuk mengimplementasikan ilmu pengetahuan teknologi dalam bidang Information Retrieval dan Data Mining terutama Klasifikasi.
2. Mendapatkan pengetahuan dan wawasan terkait metode-metode yang digunakan dalam skripsi ini.

Bagi Pengguna

1. Mendapatkan wawasan akan pengimplementasian dari metode Improved KNN dalam klasifikasi data.

1.5 Batasan masalah

Dalam penelitian Klasifikasi Kemacetan Lalu Lintas di Kota Malang pada Sosial Media Twitter Menggunakan Metode Improved KNN ini batasan penelitian yang digunakan adalah sebagai berikut:

1. Data Twitter yang digunakan adalah dari akun @PuspitaFM dengan jumlah 600 tweet, *tweet* yang digunakan adalah *tweet* tentang keadaan lalu lintas di Kota Malang.
2. Lokasi tempat yang digunakan dalam penelitian terdiri dari beberapa ruas jalan di Kota Malang.
3. Hasil dari penelitian ini belum bisa digunakan untuk di seluruh daerah Kota Malang, hanya dari beberapa ruas jalan saja.

1.6 Sistematika pembahasan

Adapun sistematika penulisan dalam skripsi ini adalah sebagai berikut:

BAB I Pendahuluan

Bab ini berisi tentang latar belakang, rumusan masalah, tujuan, batasan masalah, manfaat, dan sistematika penulisan dalam penelitian Klasifikasi Kemacetan Lalu Lintas di Kota Malang Pada Sosial Media Twitter dengan Menggunakan Metode Improved KNN.

BAB II Tinjauan Pustaka

Tinjauan pustaka menjelaskan tentang kajian pustaka terkait dengan penelitian Klasifikasi Kemacetan Lalu Lintas di Kota Malang Pada Sosial Media Twitter dengan Menggunakan Metode Improved KNN.

BAB III Metodologi Penelitian

Metodologi menjelaskan tentang metode yang digunakan dalam penelitian Klasifikasi Kemacetan Lalu Lintas di Kota Malang Pada Sosial Media Twitter dengan Menggunakan Metode Improved KNN.

BAB IV Perancangan

Perancangan menjelaskan tentang analisis kebutuhan serta perancangannya, yaitu aplikasi Klasifikasi Kemacetan Lalu Lintas di Kota Malang Pada Sosial Media Twitter dengan Menggunakan Metode Improved KNN.

BAB V Implementasi

Implementasi menjelaskan tentang pengimplementasian dari metode yang digunakan yaitu Improved KNN pada Klasifikasi Kemacetan Lalu Lintas di Kota Malang pada Sosial Media Twitter.

BAB VI Pengujian dan Analisis

Pengujian dan analisis menjelaskan tentang suatu proses dengan hasil pengujian pada Klasifikasi Kemacetan Lalu Lintas di Kota Malang Pada Sosial Media Twitter dengan Menggunakan Metode Improved KNN.

BAB VII Penutup

Penutup berisi kesimpulan yang telah diperoleh dari perancangan, implementasi, dan pengujian pembuatan serta saran-saran untuk pengembangan sistem lebih

lanjut pada Klasifikasi Kemacetan Lalu Lintas di Kota Malang Pada Media Sosial Twitter dengan Menggunakan Metode Improved KNN.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Pada penelitian ini membahas mengenai pengklasifikasian kemacetan lalu lintas di Kota Malang pada media sosial twitter yang dilakukan dengan menggunakan metode. Penelitian ini memiliki judul "Klasifikasi Kemacetan Llu Lintas Di Kota Malang Pada Media Sosial Twitter Dengan Menggunakan Metode *Improved K-Nearest Neighbor*". Metode yang digunakan pada perhitungan dalam penelitian ini yaitu metode *Improved K-Nearest Neighbor* (KNN). Penelitian ini dilakukan dengan mengacu beberapa penelitian lain yang telah dilakukan sebelumnya yang berkaitan dengan penelitian ini. Publikasi penelitian yang digunakan sebagai acuan yaitu berupa dua buah jurnal yang digunakan untuk membantu pemecahan permasalahan pada penelitian ini.

Penelitian sebelumnya yang digunakan sebagai acuan dalam penelitian ini berjudul "Klasifikasi Posting Twitter Kemacetan Lalu Lintas Kota Bandung Mengunakan *Naive Bayes Classification"* yang dilakukan oleh Sandi afajar Rodiansyah dan Edi Winarko pada tahun 2013. Penelitian tersebut dilakukan dengan metode Naive Bayes Classificatin karena metode tersebut merupakan algortima klasifikasi sederhana yang mempunyai nilai akurasi yang tinggi (Rodansyah dan Winarko, 2013). Berdasarkan penelitian tersebut hasil akurasi terkecil memiliki nilai 78% dengan sampel sebanyak 100 dan hasil akurasi terbesar memiliki nilai 91,60% dengan sebanyak 13160. Pada penelitian ini juga dilakukan dengan menggunakan metode *Support Vector Machine* (SVM) yang memiliki hasil akurasi terkecil 92% dengan sampel sebanyak 100 dan hasil tertinggi 99,11% dengan sampel sebanyak 13160.

Untuk penelitian selanjutnya yang dijadikan acuan berjudul "Klasifikasi Spam Pada Twitter Menggunakan Metode *Improved K-Nearest Neighbor"* yang dilakukan oleh Dea Zakia Nathania, Indriarti dan Fitra Abdurrachman Bachtiar pada tahun 2018. Penelitian tersebut membandingkan kinerja dari metode KNN dan metode Improved KNN dalam implementasi klasifikasi spam pada twitter. Pada penelitian tersebut digunakan data latih sebanyak 500 dokumen. Hasil dari penelitian tersebut adalah metode Improved KNN memiliki akurasi yang lebih tinggi dengan nilai akurasi 92% dibandingkan dengan metode KNN dengan nilai akurasi 88%.

2.2 Dasar Teori

2.2.1 Kemacetan

Kemacetan merupakan kejadian terganggunya lalu lintas yang dapat menyebabkan tersedatnya hingga terhentinya lalu lintas yang dapat terjadi karena jumlah kendaraan yang ada melebihi kapasitas jalan. Penyebab terjadinya kemacetan juga bisa disebabkan karena perbandingan ruas jalan dengan banyaknya kendaraan yang tidak seimbang, adanya kecelakaan lalu lintas,

meningkatnya pengguna kendaraan pribadi, serta *attitude* dari pengguna jalan itu sendiri. Kemacetan lalu lintas sangatlah menggangu masyarakat karena dapat mengakibatkan terbuangnya waktu di jalan juga dapat menyebabkan pemborosan bahan bakar. Upaya dalam mengurangi kemacetan yang ada saat ini yaitu dengan mengurangi jumlah volume kendaraan, misal dengan pengaturan genap-ganjil. Upaya lain yang ada yaitu pembangunan jalan tol, penambahan kapasitas jalan.

Dari kemacetan lalu lintas ada bebapa informasi yang bersangkutan yaitu kapan hari terjadinya kemacetan, kedua kapan waktu tepatnya terjadinya kemacetan dan yang ketiga adalah lokasi dan arah kemacetan. Kemacetan lalu lintas bisa terjadi secara acak pada hari tertentu. Begitu juga dengan waktu, kemacetan bisa terjadi pada jam-jam tertentu baik agi siang maupun malam. Lokasi dan arah terjadinya kemacetan lalu lintas juga beragam, bisa saja kemacetan terjadi di jalan tertentu namun hanya arah sisi sebelah yangterjadi kemacetan.

2.2.2 Media Sosial Twitter

Twitter merupakan layanan jejaring sosial berbasis *microblog* yang memungkinkan didirikan oleh Evan Williams, Christoper "Biz" Stone, Jack Dorsey dan Noah Glass pada tahun 2006. Jejaring sosial ini memungkinkan para penggunanya untuk saling mengirim dan membaca pesan dengan *tweet* atau kicauan (Claudy, 2018). Pada awalnya Twitter hanya bisa mengirimkan pesan dengan batas 140 karakter saja hingga pada akhir September 2017 Twitter mengeluarkan sebuah wacana penambahan batas karakter menjadi 280 karakter dan itu terjadi sampai saat ini.

Dari data yang terdapat pada media berita BeritaSatu, di Indonesia Twitter merupakan salah satu sosial media yang memiliki pengguna terbanyak nomor lima sedunia. Pada data yang dirilis oleh Twitter Indonesia pada akhir 2016, disebutkan pegguna aktis Twitter di Indonesia mencapai 77 persen. Selain itu, pengguna Twitter di Indonesia juga termasuk yang paling cerewet. Hal tersebut dapat dilihat dari jumlah tweet yang dihasilkan sepanjang 2016 yang mencapai 4,1 miliar tweet.

2.2.3Akun Twitter @PuspitaFM

Akun Twitter @PuspitaFM adalah akun radio yang ada di Kota Malang yang mulai bergabung dengan Twitter pada Maret 2012. Pada akun ini sering memberikan informasi tentang kondisi lalu lintas, informasi cuaca, pengetahuan random, waktu adzan, berita kehilangan dan lain-lain. Informasi kemacetan pada akun ini didapatkan dari kicauan langsung dari akun tersebut maupun dari hasil *retweet* yang dilakukan oleh akun @PuspitaFM dari kicaun para pengikut akun tersebut.

2.2.3 Data Mining

Menurut Larose, 2005 data mining adalah sebuah analisis dari peninjauan kumpulan data untuk menemukan hubungan yang tidak diduga dan meringkas data yang berbeda dengan cara yang berbeda denagn sebelumnya, yang dapat

dipahami dan bermanfaat bagi pemilik data. Data Mining merupakan bidang dari beberapa bidang keilmuan yang menyatukan teknik dari pembelajaran mesin, pengenalan pola, statistic, database, dan visualisasi untuk penanganan permasalahan pengambilan informasi dari database yang besar. Sedangkan menurut Pramudiono, 2006 data mining adalah serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual. Data mining merupakan anilisis otomatis dari data yang jumlah besar atau kompleks dengan tujuan menemukan pola atau kecenderungan yang penting yang biasanya tidak disadari keberadaannya.

Menurut Larose, 2005 data mining dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat di lakukan, yaitu:

2.2.3.1.1.1.1 Deskripsi, secara sederhana terkadang peneliti ingin mencoba mencari cara untuk mnggambarkan pola da kecenderungan yang terdapat dalam data.

2.2.3.1.1.1.2 Estimasi, hampir sama dengan klasifikasi, kecuali variabel target estimasi lebih ke arah numerik dari pada ke arah kategori.

2.2.3.1.1.1.3 Prediksi, hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil akan ada di masa mendatang. Beberapa metode dan teknik yang digunakan dalam klasifikasi dan estimasi dapat pula digunakan untuk prediksi.

2.2.3.1.1.1.4 Klasifikasi, terdapat target variabel kategori. Sebagai contoh, penggolongan pendapatan dapat dipisahkan dalam tiga kategori, yaitu pendapatan tinggi, pendapatan sedang, dan pendapatan rendah. Dalam penelitian ini klasifikasi digunakan dalam pemrosesan data.

2.2.3.1.1.1.5 Clustering, pengklusteran merupakan pengelompokan record, pengamatan, atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan. Kluster adalah kumpulan record yang memiliki kemiripan satu dengan yang lainnya dan memiliki ketidakmiripan dengan record-record dalam kluter lain.

2.2.3.1.1.1.6 Asosiasi, tugas aosiasi dalam data mining adalah menemukan atribut yang muncul dalam suatu waktu.

2.2.4 Klasifikasi

Klasifikasi adalah metode data mining yang dapat digunakan untuk proses pencarian sekumpulan model (fungsi) yang dapat menjelaskan dan membedakan kelas-kelas data atau konsep, yang tujuannya supaya model tersebut dapat digunakan memprediksi objek kelas yang labelnya tidak diketahui atau dapat memprediksi kecenderungan data-data yang muncul di masa depan. Metode klasifikasi juga bertujuan untuk melakukan pemetaan data ke dalam kelas yang sudah didefinisikan sebelumnya berdasarkan pada nilai atribut data (Han dan Kamber, 2006).

2.2.5 *Improved K-Nearest Neighbor*

Algortima Improved KNN melakukan modifikasi pada penentuan nilai *k-values*. Penentuan nilai *k-values* yang tepat dilakukan untuk mendapatkan nilai akurasi yang tinggi. Dalam *Improved KNN* penetapan nilai *k-values* dilakukan dengan menetapkan nilai yang berbeda pada setiap kategori. Perbedaan tersebut dilakukan dengan menyesuaikan besar kecilnya jumlah dokumen latih pada setiap kategori tersebut. Hasilnya jika nilai *k-values* semakin tinggi maka hasil kategori tidak terpengaruh oleh kategori yang mempunyai jumlah dokumen latih yang memiliki nilai lebih besar (Herdiawan, 2015). Improved K-Nearest Neighbor merupakan algoritma yang menggunakan nilai k-values (jumlah tetangga terdekat) yang berbeda pada setiap kategorinya. Perbedaan metode Improved Knn dengan metode K-Nearest Neighbor yaitu pada metode KNN algoritma yang menggunakan satu nilai k-values untuk seluruh kategori yang ada (Baoli, Shiwen dan Qin, 2003).

Setelah melakukan perhitungan Cosine similarity maka akan dilanjutkan dengan melakukan pengurutan hasil perhitungan Cosine similarity secara menurun pada setiap kategori. Dilanjutkan dengan perhitungan nilai k-values yang baru (n). dimana persamaan Cosine similarity menjelaskan mengenai proporsi dari penetapan nilai k-values yang baru (n) pada tiap-tiap kategori (Herdiawan, 2015).

$$n = [\frac{k * N(C_m)}{maks\ \{N(C_m) |\ j=1.....N_c\}}] \quad (2.1)$$

Dimana:

- n : nilai k-values baru
- k : nilai k-values yang ditetapkan di awal
- $N(Cm)$: jumlah dokumen/data latih pada kategori m
- maks N cm j=1……Nc} : jumlah dokumen/data latih terbanyak pada semua kategori yang ada.

Dalam menentukan hasil kategori pada dokumen uji, maka dilakukan perbandingan similaritas pada masing-masing kategori. Persamaan (2.2) dibawah ini menyatakan nilai maksimum dari perbandingan antara kemiripan dokumen X dengan dokumen latih dj sejumlah top n tetangga pada suatu kategori dengan dokumen X terhadap dokumen latih dj sejumlah top n tetangga pada training set (Baoli, Shiwen dan Qin, 2003).

$$p(x, c_m) = argmax_m\ \frac{\sum d_j \in top\ n\ KNN(c_m) sim(x,d_j) y(d_j.c_m)}{\sum d_j \in top\ n\ KNN(c_m) sim(x,d_j)}] \quad (2.2)$$

Dimana:

- p(x,cm) : probabilitas dokumen X menjadi anggota kategori cm
- $sim(x,dj)$: kemiripan antara dokumen X dengan dokumen latih dj
- top n kNN : top n tetangga

- y(dj,cm) : fungsi atribut yang memenuhi suatu kategori tertentu, akan bernilai 1 apabila dokumen latih dj masuk ke dalam anggota cm, jika tidak maka akan bernilai 0.

Berdasarkan perhitungan dengan menggunakan persamaan diatas, maka selanjutnya akan dibandingkan hasil probabilitas untuk masing-masing kategori. Hasil kategori akan mengacu kepada hasil probabilitas terbesar (Putri, 2013).

2.2.6 Sistem Temu Kembali Informasi

Sistem Temu Kembali Informasi atau bisa juga disebut dengan *Infromation Retrieval* adalah suatu sistem yang menerima query dari pengguna yang kemudian akan dilakukan perankingan terhadap dokumen berdasarkan kesesuaian terhadap query. Hasil ranking berupa dokumen yang memiliki relevansi terhadap query, namun tingkat relevansi merupakan hal yang subjektif tergantung dari pengguna yang dipengaruhii oleh berbagai macam faktor seperti topik, pewaktuan, sumber informasi maupun tujuan pengguna. Model sistem temu kembali menentukan detail sistem temu yaitu meliputi representasi dokumen maupun query, fungsi pencarian (retrieval function), dan notasi kesesuaian (relevance notation) dokumen terhadap query (DwijaWisnu, 2015).

2.2.7 Prepocessing Teks

Prepocessing teks dilakukan untuk menyiapkan teks kedalam bentuk data yang kemudain akan diolah pada proses berikutnya (Puspitasari, 2018). Terdapat beberapa jenis prepocessing dalam konteks data teks, yaitu:

2.2.7.1.1.1.1.1 *Cleansing*

Cleansing merupakan tahapan dalam *prepocessing text* yang bertujuan untuk mengurangi *noise* yang terdapat pada data. Pada tahapan ini dilakukan proses penghapusan URL, *hashtag* (#aaa), *username* (@aaa), angka, karakter selain huruf alfabet dan tanda baca (Nathania, 2018)

2.2.7.1.1.1.1.2 Case Folding

Case folding merupakan tahapan dalam *preprocessing text* yang dilakukan untuk mengubah inputan yang berupa huruf kapital (*upper case)* menjadi huruf kecil (*lower case)*.

2.2.7.1.1.1.1.3 Tokenisasi

Tokenisasi merupakan tahapan dalam *prepocessing text* yang dilakukan dengan memotong setiap kata yang terdapat dalam kalimat dengan menggunakan spasi yang dijadikan sebagai delimiter yang kemudian menghasilkan token yang berupa kata (DwijaWisnu, Hetami., 2015).

2.2.7.1.1.1.1.4 *Filtering*

Filtering merupakan tahapan dalam *prepocessing text* yang dilakukan untuk menghilangkan kata yang dianggap tidak penting atau kata yang tidak bermakan yang termasuk dalam *stoplist. Stoplist* merupakan kata-kata yang

sering muncul dalam dokumen namun tidak mempunyai kaitan dengan tema tertentu (DwijaWisnu, Hetami., 2015).

2.2.7.1.1.1.1.5 *Stemming*

Stemming merupakan tahapan dalam *prepocessing text* yang dilakukan untuk mengubah bentuk kata hasil dari proses *filtering* yang berupa kata imbuhan baik imbujan awal maupun imbuhan akhir menjadi kata dasar. Hasil dari proses *stemming* yaitu berupa *root word* (DwijaWisnu, Hetami., 2015).

2.2.8 Pembobotan (*Term Weghting)*

Term Weighting merupakan proses pembobotan yang dilakukan pada setiap kata atau *term* untuk mendapatkan nilai dari kata aTau *term* yang sudah diproses sebelumnya (Puspitasari, Santoso, Indriarti, 2018). Metode pembobotan dalam penelitian ini yaitu *Term Frequency-Inverse Document (*TF-IDF).

Term Frequency (TF) merupakan jumlah kemunculan suatu kata atau *term* dalam dokumen. Semakin sering kata atau *term* muncul dan sama dengan *term* dalam dokumen, maka nilai *Term Frequency* TF akan semakin bertambah. Jika suatu kata atau *term* mempunyai nilai frekuensi kemunculan yang tinggi dalam dokumen, maka kata atau *term* tersebut memiliki pengaruh besar dalam dokumen tersebut (Puspitasari, Santoso, Indriarti, 2018). Pembobotan kata atau *term* dalam sebuah dokumen yaitu:

$$W_{tf}(t,d) = 1 + f(t,d) = \begin{cases} 1 + \log_{10} tf_{t,d}, & if\ tf_{t,d} > 0 \\ 0, & otherwise \end{cases} \quad (2.3)$$

Dimana:

- $W_{tf_{t,d}}$: Hasil dari pembobotan $tf_{t,d}$
- f(d,t) : frekuensi kemunculan *term t* pada dokumen *d.*
- $tf_{t,d}$: Frekuensi kemunculan *t* pada dokumen *d*

Inverse Document Frequency (IDF) adalah jumlah kemunculan suatu kata atau *term* dalam kumpulan dokumen (Fauzi, Arifin, Yuniarti, 2014). Perhitungan IDF yaitu:

$$IDF(t) = 1 + \log\ (\frac{Nd}{df(t)}) \quad (2.4)$$

Dimana:

- Nd: jumlah seluruh dokumen
- df(t): jumlah dokumen yang memiliki *term* t.

Pembobotan TF-IDF merupakan pembobotan yang menerapkan pembobtan TF dan IDF yang dihitung dengan mengalikan bobot TF dengan IDF. Adapun rumusnya yaitu:

$$w(t,d) = W_{t,f}(t,d) \times idf_t \quad (2.5)$$

Dimana:

- $w(t,d)$: pembobotan TF.IDF
- $w_{tf}(t,d)$: hasil pembobotan tft,d
- $idft$: hasil invers dari dft

Setelah menghitung pembobotan TF-IDF, maka dilanjutkan dengan perhitungan normalisasi terhadapa nilai w(t,d). Adapun untuk menghitung normalisasi dihitung dengan menggunakan rumus:

$$w_{t,d} = \frac{w_{t,d}}{\sqrt{\sum_{t=1}^{n} {w_{t,d}}^2}} \quad (2.6)$$

2.2.9 Cosine Similiarity

Proses *cosine similiarity* dilakukan setelah melakukan pembobotan kata dan normalisasi. *Cosine similiarity* merupakan metode yang digunakan untuk menghitung tingkat kemiripan antara dokumen dengan *query* (Fauzi, Arifin, Yuniarti, 2014). Adapun rumus perhitungannya adalah :

$$Cosine\ (d_i, q_i) = \frac{q_i, d_i}{|q_i||d_i|} = \frac{\sum_j^t{}_{=i}(q_{ij}, d_{ij})}{\sqrt{\sum_j^t{}_{=i}(q_{ij})^2 . \sum_j^t{}_{=i}(d_{ij})^2}} \quad (2.7)$$

Dimana:

- q_{ij} = Bobot *j* pada dokumen i
- d_{ij} = Bobot *j* pada dokumen i

2.2.10 Confusion Matrix

Penelitian ini menggunakan metode klasifikasi, oleh karena itu diperlukan evaluasi untuk mengetahui akurasi dan kinerja dari metode yang digunakan ini. *Confusion matrix* merupakan tabel yang digunakan untuk membandingkan kategori hasil prediksi dengan kategori aktual (Nathania, Indriarti dan Bachtiar, 2018). Pada Tabel 2.1 ditunjukan *confession matrix* yang digunakan pada penelitian ini.

Tabel 2. 1 *Confusion Matrix*

		Hasil Aktual	
		Macet	Tidak Macet
Hasil Prediksi	Macet	TP	FP
	Tidak Macet	FN	TN

Keterangan:

- TP : *True Positive*, banyaknya jumlah data hasil prediksi merupakan kategori macet dan data hasil aktual merupakan kategori macet.
- FP : *False Positive,* banyaknya jumlah data hasil prediksi merupakan kategori macet dan data hasil aktual merupakan kategori tidak macet.

- FN : *False Negative,* banyaknya jumlah data hasil prediksi merupakan kategori tidak macet dan data hasil aktual meupakan kategori macet.
- TN : *True Negative,* banyaknya jumlah data hasl prediksi merupakan kategori tidak macet dan data hasil aktual merupakan kategori tidak macet.

2.2.11 *Precision, Recall, F-Measure* dan Akurasi

Dalam penelitian ini digunakan evaluasi untuk menghitung akurasi sistem dengan menggunakan parameter *precision, recall* dan *f-measure*. *Precision* merupakan tingkat keakuratan dari hasil klasifikasi pada seluruh dokumen untuk mengetahui kategori data yang diklasifikasikan sesuai dengan kategori sebenarnya. *Recall* merupakan parameter untuk mengetahui tingkat keberhasilan suatu sistem dalam mengenali kategori. *F-measure* adalah gambaran mengenai pengaruh antara *precision* dan *recall* (Puspitasari, Santoso, Indriarti, 2018). Adapun rumus untuk menghitung *precision, recall* dan *f-measure* ditunjukkan pada persamaan 2.8, 2.9 dan 2.10.

$$precision = \frac{TP}{TP+FP} \quad (2.8)$$

$$recall = \frac{TP}{TP+FN} \quad (2.9)$$

$$F1 = \frac{2 \times recall \times precision}{recall+precision} \quad (2.10)$$

Untuk mendapatkan hasil akurasi dari penelitian ini, maka digunakan persamaan 2.11.

$$akurasi = \frac{TP+TN}{TP+FP+TN+FN} * 100\% \quad (2.11)$$

BAB 3 METODOLOGI

Dalam bab ini akan dijelaskan mengenai metode, teknik dan proes dalam penelitian yang digunakan dalam pembuatan Klasifikasi Kemacetan Lalu Lintas di Kota Malang pada Sosial Media Twitter Menggunakan Metode *Improved K-Nearest Neighbor*. Pada bab ini akan dijelaskan mengenai Tipe Penelitian dan Strategi dan Rancangan Penelitian.

3.1 Tipe Penelitian

Dalam penelitian ini, tipe penelitian yang digunakan adalah tipe penelitian nonimplementatif. Tipe penelitian nonimplementatatif merupakan proses yang lebih mengutamakan pada penggalian informasi dari informasi yang digunakan untuk mengidentifikasi elemen penting objek penelitian. Dari proses tersebut akan dijadikan dasar dalam mengambil keputusan atau penelitian lanjut. Segingga, hasil dari penggalian informasi dapat menjawab dai identifikasi masalah penelitian secara kuantitatif maupun kualitatif.

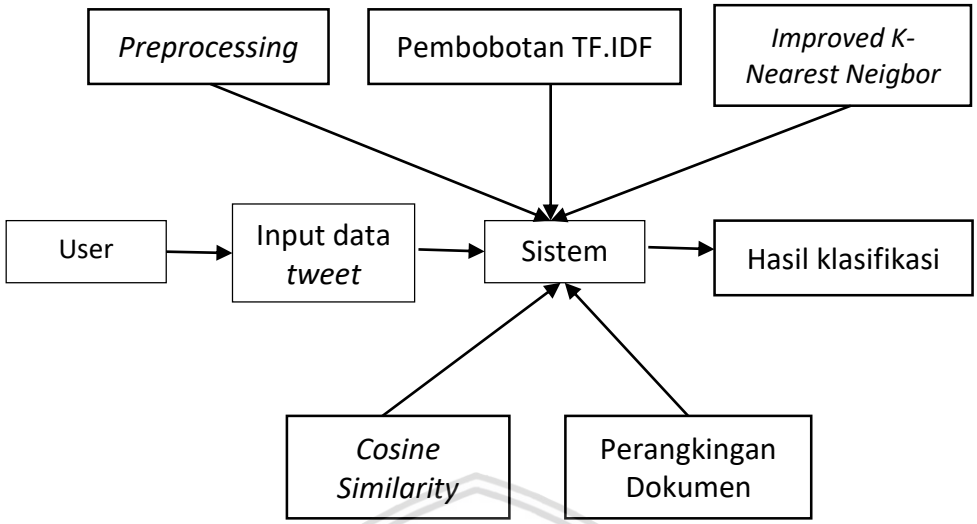
Pendekatan pada tipe penelitian ini yang ditinjau dari kegiatan penenlitian merupakan penelitian nonimplmentatif analitik (*analytical/explanatory*). Tipe ini menjelaskan derajat hubungan antar elemen dari objek penelitian.

3.2 Strategi Penelitian

Strategi dalam penelitian ini yaitu eksperimen. Startegi penelitian eksperimen merupakan penelitian yang terfokus pada satu atau lebih variabel . strategi eksperimen digunakan untuk melihat pengaruh variabel yang satu tehadap yang lain. Dalam penelitian ini digunakan beberapa algoritma yaitu *prepocessing, term weighting, cosine similiarity* dan *Imprved K-Nearest Neighbor.*

3.3 Rancangan Penelitian

Rancangan peneltian ini menjelaskan mengenai gambaran rancangan sistem yang digunakan dalam peneltian ini yang berupa bagaimana cara kerja sistem mulai dari input, proses dan output. Gambaran sistem pada penelitian ini ditunjukkan pada Gambar 3.1.



Gambar 3. 1 Model Perancangan Arsitektur

Berdasarkan pada Gambar 3.1 *user* akan menginputkan data *tweet* yang kemudian akan melalui proses *prepocesing,* pembobotan TF.IDF, *cosine similiarity* dan klasifikasi *Improved k-nearest neigbor* yang kemudian akan menghasilkan output berupa hasil klasifikasi.

3.3.1 Partisipan Penelitian

Partisipan penelitian yang terlibat dalam penelitian ini yaitu akun @PuspitaFM, baik dari admin maupun para *follower* dari akun tersebut yang men*tweet* tentang keadaan lalu lintas di kota Malang.

3.3.2 Lokasi Penelitian

Pada penelitian ini tidak terdapat lokasi khusus pada penelitian yang dilakukan. Penelitian ini dilakukan pada sosial media Twitter dengan nama akun @PuspitaFM yang bisa diakses melalui *smartphone* dan web. Pada proses-proses tertentu penelitian ini dilakukan di Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya.

3.3.3 Teknik Pengumpulan Data

Teknik pengumpulan data pada penelitian ini menggunakan metode data sekunder. Data sekunder merupakan data primer yang sudah diolah lebih lanjut. Metode data sekunder disebut juga dengan meode penggunaan bahan dokumen, yaitu tidak mengambil data secara langsung tetapi meggunakan dokumen atau data yang sudah dihasilkan dari pihak pihak lain. Dalam peneltian ini data ddapatkan dari sosial media twitter dengan nama akun @PupitaFM. Datanya berupa *tweet* langsung dari akun @PuspitaFM maupun *tweet* tidak langsung yang berupa hasil *retweet* dari *follower* akun tersebut. Kemudian data-data tersebut akan dilakukan proses mulai dari *prepocesing,* pembobotan TF.IDF, *cosine similiarity* hingga klasifikasi *Improved K-Nearest Neigbor*.

3.3.4 Teknik Pengumpulan Data

Pengujian yang dilakukan dengan melakukan pengujian validasi, pengujian akurasi sistem dan pengujian sensifitas metode *Improved KNN* pada sistem yang telah dibuat pada tahap implementasi. Pengujian validasi dilakukan dengan memeriksa cara kerja sistem, apakah sudah baik dan tidak adanya kesalahan dan sesuai kebutuhan. Serta solusi yang diberikan sesuai dengan pengetahuan yang dimasukan ke dalam system. Pengujian dilakukan dengan menghitung nilai *precission, recall, f-measure* dan akurasi. Sealin itu dilakukan pengujian dengan membandngkan hasil akurasi dari metode *Improved K-Nearest Neighbor* dengan *K-Nearest Neigbor (KNN).*

3.3.5 Analisis Kebutuhan

Analisa kebutuhan digunakan untuk mengetahui kebutuhan yang dibutuhkan oleh sistem dan yang diperlukan dalam perancangan sistem, implementasi, serta pengujian sistem. Adapun kebutuhan yang dibutuhkan ialah sebagai berikut :

3.3.5.1.1.1.1 Perangkat Keras (Hardware), meliputi:

- Laptop ASUS
- Processor Intel Core i3 4030U, 1.9GHz
- RAM 2GB

3.3.5.1.1.1.2 Perangkat Lunak (Software), meliputi:

- Sistem Operasi *Windows* 10 64 bit
- Bahasa Pemrogramman PHP dan HTML
- XAMPP
- Notepad++
- MySQL
- Browser Google Chrome

3.4 Penarikan Kesimpulan dan Saran

Kesimpulan akan dilakukan setelah semua tahapan perancangan, implementasi dan pengujian metode yang diterapkan telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis metode yang diterapkan. Saran didapatkan untuk membantu dalam proses pengembangan penelitian selanjutnya supaya penelitian ini bisa lebih baik.

3.5 Jadwal Penelitian

Penelitian ini dilakukan dari bulan Februari hingga Oktober. Adapaun jadwal penelitian ditunjukkan pada Tabel 3.1.

Tabel 3. 1 Jadwal Penelitian

No	Uraian	Minggu ke-																											
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
		Februari				Maret				April				Mei				Juni				Juli				Oktober			
1	Konsultasi dan Penyusunan Laporan																												
2	Penyerahan Proposal																												
3	Perbaikan/ Revisi Proposal Penelitian																												
4	Pengumpulan Data																												
5	Penyusunan Laporan Penelitian																												
6	Bimbingan dan Konsultasi Hasil Penelitian																												
7	Seminar Hasil Penelitian																												
8	Perbaikan Hasil Penelitian																												
9	Sidang Skripsi																												
10	Perbaikan Hasil Sidang Penelitian																												

BAB 4 PERANCANGAN DAN IMPLEMENTASI

4.1 Deskripsi Masalah

Kemacetan lalu lintas adalah masalah yang sering terjadi di kota-kota besar, begitu juga dengan Kota Malang. Kota Malang sendiri merupakan kota termacet ketiga di Indonesia setelah Kota Jakrta dan Kota Bandung (Ramadhiani, 2018). Penyebab terjadinya kemacetan di Kota Malang juga beragam diantaranya yaitu perbandingan ruas jalan dengan banyaknya kendaraan yang tidak seimbang, adanya kecelakaan lau lintas, meningkatnya pengguna kendaraan pribadi, serta *attitude* dari pengguna jalan itu sendiri. Faktor lain penyebab kemacetan yang terjadi di Kota Malang adalah banyaknya mahasiswa di Kota Malang. Berdasarkan data yang terdapat pada web suryamalang.tribunnews.com terdapat 57 perguruan tinggi dengan ribun mahasiswa. Jadwal kuliah mereka yang tak teratur memuat jalan setiap saat dipenuhi mahasiswa. Hal tersebut juga menjadi penyebab terjadinya kemacetan di Kota Malang (Suryanik, 2017).

Perkembangan teknologi saat ini dapat membantu rutinitas kegiatan sehari-hari, diantaranya adalah memantau kemacetan lalu lintas melalui sosial media Twitter. Kemacetan lalu lintas saat ini dapat dirasakan mengganggu rutinitas kegiatan masyarakat, begitu juga dengan masyarakat Kota Malang. Dengan perkembangan teknologi tersebut masyarakat Kota Malang dapat mencari informasi bahkan bertukar informasi mengenai keadaan lalu lintas di Kota Malang melalui sosial media Twitter. Berdasarkan data yang diambil dari www.kominfo.go.id Indonesia menempati peringkat kelima pengguna Twitter terbesar di dunia dengan 19,5 juta pengguna. Dengan banyaknya pengguna tersebut besar kemungkinan semakin mudahnya akses dalam mendapatkan informsasi atau bertukarnya informasi mengenai keadaan lalu lintas yang ada di Kota Malang. Dalam *tweet* yang ada terkadang terdapat kata yang ambigu dalam menentukan kategori macet atau tidak macet. Oleh karena itu, penelitian ini dilakukan untuk mendapatkan hasil klasifikasi yang lebih akurat dengan menggunakan metode *Improved K-Nearest Neighbor (KNN)*.

4.2 Deskripsi Sistem

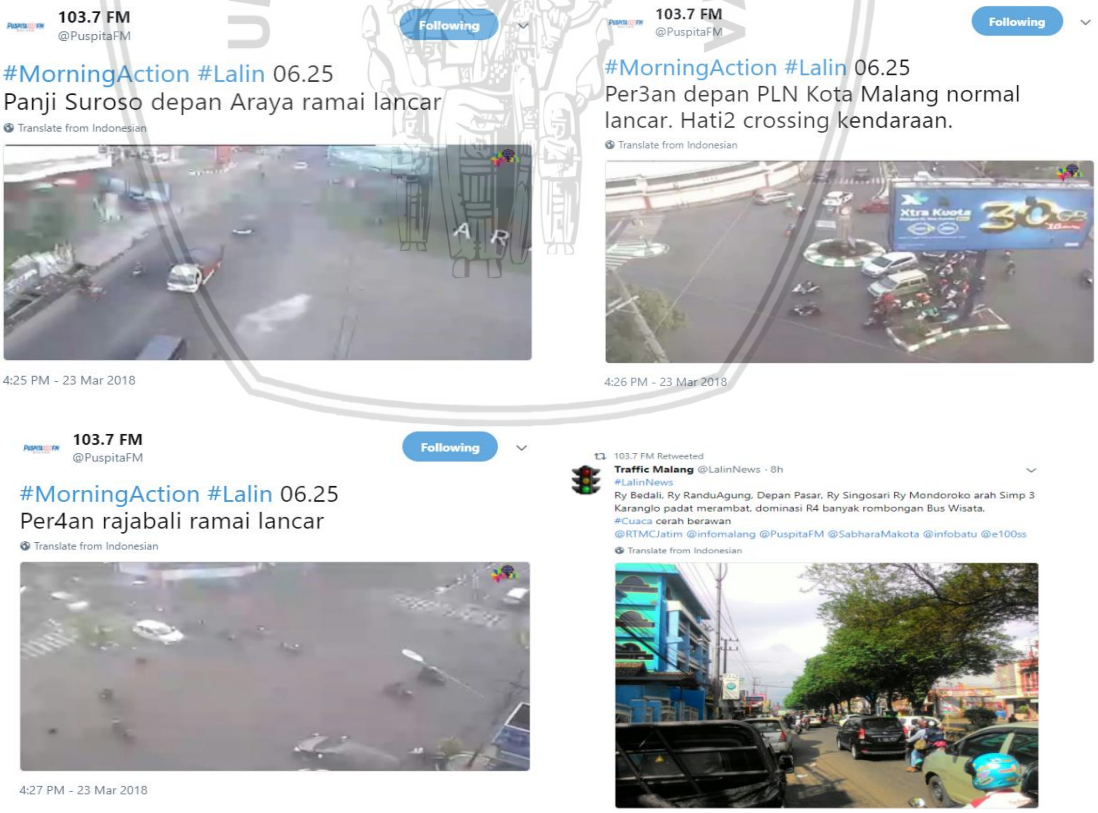
Klasifikasi Kemacetan Lalu Lintas di Kota Malang pada Sosial Media Twitter Menggunakan Metode *Improved K-Nearest Neighbor (KNN)* merupakan suatu sistem yang dirancang guna pengklasifikasian pada *tweet* kemacetan lalu lintas di Kota Malang untuk mengetahui kategori *tweet* tersebut masuk kategori macet atau tidak macet dengan menggunakan metode *Improved K-Nearest Neighbor (KNN)*. Adapun hasil dari klasifikasi ini dipengaruhi oleh banyaknya jumlah data latih yang digunakan. Data latih yang digunakan merupakan data latih yang sudah melalui tahapan-tahapan dalam *prepocessing*, *term weighting* dan normalisasi hasil dari *term weighting*. Dari data latih tersebut kemudian dilakukan klasifikasi dengan menggunakan metode *Improved K-Nearest Neighbor (KNN)*.

4.3 Manualisasi

4.3.1 Proses Pengumpulan Data

Proses pengumpulan data yang diambil dari akun @PuspitaFM dilakukan secara manual karena data yang dibutuhkan tidak terlalu banyak sehingga data tersebut nantinya akan langsung disimpan ke dalam *database*. Pemilihan *tweet* tentang keadaan lalu lintas diambil dari *tweet* langsung dari akun @PuspitaFM dan hasil *retweet* dan *liked* dari *followers* akun tersebut. Data yang digunakan adalah sejak tanggal 15 Maret 2018. Setiap harinya akun @PuspitaFM akan men*tweet* tentang keadaaan lalu lintas di Kota Malang sekitar 20 lebih *tweet* dalam sehari. Apabila pada hari liburan tertentu keadaan lalu lintas di Kota Malang akan semakin padat daripada hari-hari biasa, sehingga banyak dari *followers* akun @PuspitaFM yang men*tweet* keadaan lalu lintas tersebut yang kemudian akan di*retweet* oleh akun @PuspitaFM. Sehingga pada hari-hari liburan tertentu yang membuat lalu lintas Kota Malang semakin padat, akun @PuspitaFM akan men*tweet* lebih banyak tentang keadaan lalu lintas di Kota Malang lebih banyak daripada hari biasanya.

Berikut merupakan contoh beberapa *tweet* dari akun @PuspitaFM tentang keadaan lalu lintas di Kota Malang pada tanggal 23 Maret 2018 yang ditunjukkan pada Gambar 4.1.



Gambar 4. 1 Contoh *Tweet* Keadaan Lalu Lintas

Dalam pelabelan keadaan lalu lintas yang terjadi ini dibagi menjadi dua kategori yaitu macet dan tidak macet. Proses pelabelan ini dilakukan dengan melakukan kuisioner kepada beberapa narasumber yaitu mahasiswa Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya. Kusioner dibagikan kepada tiga mahasiswa denga hasil pelabelan yang sama untuk semua data latih. Berikut merupakan pemberian label pada *tweet* yang disesuakan pada *tweet*, *retweet* dan *liked* dari akun @PuspitaFM tentang keadaan lalu lintas yang ditunjukkan pada Tabel 4.1.

Tabel 4. 1 Contoh Pelabelan *Tweet* Keadaan Lalu Lintas

No.	*Tweet*	Kategori
1.	15.34 Pertigaan PLN ramai lancar cuaca panas Mar 15 2018	Tidak Macet
2.	15.27 Wib arus lalu lintas simp PDAM Malang terpantau ramai lancar tetap hati2 selama berkendara Mar 15 2018	Tidak Macet
3.	17.43 Perempatan Dieng ramai lancar Mar 15 2018	Tidak macet
4.	#MorningAction #Lalin 06.27 Pertigaan depan PLN Kota Malang padat. Terutama di JA Suprapto. Belum ada petugas yg mengatur lalin. MAR 15 2018	Macet
5.	#MorningAction #Lalin 06.27 Jalan A. Yani, Kecamatan Lawang depan bakpao telo arah Surabaya masih padat. | CCTV Malang Today MAR 15 2018	IPMacet

4.3.2 *Prepocessing Text*

Prepocesing Text merupakan tahapan yang dilakukan untuk menyiapkan teks ke dalam bentuk data yang kemudian akan dolah pada proses berikutnya (Puspitasari, 2018). Tahapan ini juga bertujuan untuk mendapatkan data terstruktur yang kemudian dapat memudahkan dalam proses perhitungan berikutnya (Nathania, 2018). Terdapat beberapa tahapan dalam *prepocesing text* yaitu *cleansing, case folding,* tokenisasi*, filtering* dan *stemming.* Dalam tahapan ini digunakan data latih sebanyak 10 buah yang terdiri dari 5 *tweet* tidak macet dan 5 *tweet* macet. Berikut merupakan data latih yang ditunjukkan pada Tabel 4.2.

Tabel 4. 2 Data Latih yang digunakan

No.	*Tweet*	Kategori
1.	15.34 Pertigaan PLN ramai lancar cuaca panas Mar 15 2018	Tidak Macet
2.	15.27 Wib arus lalu lintas simp PDAM Malang terpantau ramai lancar tetap hati2 selama berkendara Mar 15 2018	Tidak Macet

3.	15.35 Perempatan Dieng ramai lancar cuaca panas Mar 15 2018	Tidak macet
4.	#MorningAction #Lalin 06.28 Panji Suroso depan Araya ramai lancar Mar 18 2018	Tidak Macet
5.	#MorningAction #Lalin 06.28 Perempatan rajabalinormal Mar 18 2018	Tidak Macet
6.	#MorningAction #Lalin 06.27 Tidar ramai lancar	Tidak Macet
7.	#MorningAction #Lalin 06.28 Pertigaan depan PLN Kota Malang padat. Terutama di JA Suprapto. Hati2 crossing kendaraan Mar 18 2018	Macet
8.	18.16 Perempatan Dieng cenderung padat Mar 18 2018	Macet
9	16.07 depan Araya padat arah Surabaya cuaca gerimis @infomalang Mar 18 2018	Macet
10.	#MorningAction #Lalin 07.02 Jalan A. Yani, Kecamatan Lawang depan bakpao telo arah ke surabaya masih padat. | CCTV Malang Today Mar 15 2018	Macet
11.	#MorningAction #Lalin 06.27 Pertigaan depan PLN Kota Malang padat. Terutama di JA Suprapto. Belum ada petugas yg mengatur lalin. Mar 15 2018	Macet

4.3.2.1 *Cleansing*

Cleansing merupakan tahapan dalam *prepocessing text* yang bertujuan untuk mengurangi *noise* yang terdapat pada data. Pada tahapan ini dilakukan proses penghapusan URL, *hashtag* (#aaa), *username* (@aaa), angka, karakter selain huruf alfabet dan tanda baca (Nathania, 2018). Berikut merupakan hasil tahapan *cleansing* dari data latih yang digunakan yang ditunjukkan pada Tabel 4.3.

Tabel 4. 3 Hasil *Cleansing*

No.	*Tweet*	Kategori
1.	15.34 Pertigaan PLN ramai lancar cuaca panas Mar 15 2018	Tidak Macet
2.	15.27 Wib arus lalu lintas simp PDAM Malang terpantau ramai lancar tetap hati2 selama berkendara Mar 15 2018	Tidak Macet
3.	15.35 Perempatan Dieng ramai lancar cuaca panas Mar 15 2018	Tidak macet
4.	06.28 Panji Suroso depan Araya ramai lancar Mar 18 2018	Tidak Macet

5.	06.28 Perempatan rajabalinormal Mar 18 2018	Tidak Macet
6.	06.27 Tidar ramai lancar Mar 15 2018	Tidak Macet
7.	06.28 Pertigaan depan PLN Kota Malang padat. Terutama di JA Suprapto. Hati2 crossing kendaraan Mar 18 2018	Macet
8.	18.16 Perempatan Dieng cenderung padat Mar 18 2018	Macet
9.	16.07 depan Araya padat arah Surabaya cuaca gerimis Mar 18 2018	Macet
10.	07.02 Jalan A. Yani, Kecamatan Lawang depan bakpao telo arah ke surabaya masih padat. CCTV Malang Today Mar 15 2018	Macet
11.	06.27 Pertigaan depan PLN Kota Malang padat. Terutama di JA Suprapto. Belum ada petugas yg mengatur lalin. Mar 15 2018	Macet

4.3.2.2 *Case Folding*

Case folding merupakan tahapan dalam *preprocessing text* yang dilakukan untuk mengubah inputan yang berupa huruf kapital (*upper case)* menjadi huruf kecil (*lower case)*. Berikut merupakan hasil tahapan *case folding* dari data latih yang digunakan yang ditunjukkan pada Tabel 4.5.

Tabel 4. 4 Hasil *Case Folding*

No.	*Tweet*	Kategori
1.	15.00-16.00 pertigaan pln ramai lancar cuaca panas kamis	Tidak Macet
2.	15.00-16.00 wib arus lalu lintas simp pdam malang terpantau ramai lancar tetap hati2 selama berkendara kamis	Tidak Macet
3.	15.00-16.00 perempatan dieng ramai lancar cuaca panas kamis	Tidak macet
4.	06.00-07.00 panji suroso depan araya ramai lancar sabtu	Tidak Macet
5.	06.00-07.00 perempatan rajabali normal sabtu	Tidak Macet
6.	06.00-07.00 tidar ramai lancar kamis	Tidak Macet
7.	06.00-07.00	Macet

	pertigaan depan pln kota malang padat. terutama di ja suprapto. hati crossing kendaraan sabtu	
8.	18.00-19.00 perempatan dieng cenderung padat sabtu	Macet
9.	16.00-17.00 depan araya padat arah surabaya cuaca gerimis sabtu	Macet
10.	07.00-08.00 jalan a. yani, kecamatan lawang depan bakpao telo arah ke surabaya masih padat. cctv malang today kamis	Macet
11.	06.00-07.00 pertigaan depan pln kota malang padat. terutama di ja suprapto. belum ada petugas yg mengatur lalin. kamis	Macet

4.3.2.3 Tokenisasi

Tokenisasi merupakan tahapan dalam *prepocessing text* yang dilakukan dengan memotong setiap kata yang terdapat dalam kalimat dengan menggunakan spasi yang dijadikan sebagai delimiter yang kemudian menghasilkan token yang berupa kata (DwijaWisnu, Hetami., 2015). Berikut merupakan hasil tahapan *case folding* dari data latih yang digunakan yang ditunjukkan pada Tabel 4.6.

Tabel 4. 5 Hasil Tokenisasi

No.	*Tweet*	Kategori
1.	15.00-16.00/pertigaan/pln/ramai/lancar/cuaca/panas/kamis	Tidak Macet
2.	15.00-16.00/wib/arus/lalu/lintas/simp/pdam/malang/terpantau/ ramai/lancar/tetap/hati/selama/berkendara/kamis	Tidak Macet
3.	15.00-16.00/perempatan/dieng/ramai lancar/cuaca/panas/kamis	Tidak macet
4.	06.00-07.00/panji/suroso/depan/araya/ramai/lancar/sabtu	Tidak Macet
5.	06.00-07.00/perempatan/rajabali/normal/sabtu	Tidak Macet
6.	06.00-07.00/tidar/ramai/lancar/kamis	Tidak Macet
7.	06.00-07.00/pertigaan/depan/pln/kota/malang/padat/terutama/di/ja/suprapto/ /hati/crossing/kendaraan/sabtu	Macet
8.	18.00-19.00/perempatan/dieng/cenderung/padat/sabtu	Macet
9.	16.00-17.00/depan/araya/padat/arah/surabaya/cuaca/gerimis/sabtu	Macet
10.	07.00-08.00/jalan/a/yani/kecamatan/lawang/depan/bakpao/tel	Macet

	o/arah/ke/surabaya/masih/padat/cctv/malang/today/kamis	
11.	06.00-07.00/pertigaan/depan/pln/kota/malang/padat/terutama/di/ja/suprapto/belum/ada/petugas/yg/mengatur/lalin/kamis	Macet

4.3.2.4 *Filtering*

Filtering merupakan tahapan dalam *prepocessing text* yang dilakukan untuk menghilangkan kata yang dianggap tidak penting atau kata yang tidak bermakna yang termasuk dalam *stoplist. Stoplist* merupakan kata-kata yang sering muncul dalam dokumen namun tidak mempunyai kaitan dengan tema tertentu (DwijaWisnu, Hetami., 2015). Berikut merupakan hasil tahapan *filtering* dari data latih yang digunakan yang ditunjukkan pada Tabel 4.7.

Tabel 4. 6 Hasil *Filtering*

No.	*Tweet*	Kategori
1.	15.00-16.00/pertigaan/pln/ramai/lancar/kamis	Tidak Macet
2.	15.00-16.00/simp/pdam/ramai/lancar/kamis	Tidak Macet
3.	15.00-16.00/perempatan/dieng/ramai lancar/kamis	Tidak macet
4.	06.00-07.00/panji/suroso/ramai/lancar/sabtu	Tidak Macet
5.	06.00-07.00/perempatan/rajabali/normal/sabtu	Tidak Macet
6.	06.00-07.00/tidar/ramai/lancar/kamis	Tidak Macet
7.	06.00-07.00/pertigaan/pln/padat/sabtu	Macet
8.	18.00-19.00/perempatan/dieng/padat/sabtu	Macet
9.	16.00-17.00/depan/araya/padat/sabtu	Macet
10.	07.00-08.00/jalan/a/yani/padat/kamis	Macet
11.	06.00-07.00/pertigaan/pln/padat/kamis	Macet

4.3.2.5 *Stemming*

Stemming merupakan tahapan dalam *prepocessing text* yang dilakukan untuk mengubah bentuk kata hasil dari proses *filtering* yang berupa kata imbuhan baik imbuhan awal maupun imbuhan akhir menjadi kata dasar. Hasil dari proses *stemming* yaitu berupa *root word* (DwijaWisnu, Hetami., 2015). Berikut merupakan hasil tahapan *filtering* dari data latih yang digunakan yang ditunjukkan pada Tabel 4.8.

Tabel 4. 7 Hasil *Stemming*

No.	*Tweet*	Kategori
1.	15.00-16.00/pertigaan/pln/ramai/lancar/kamis	Tidak Macet
2.	15.00-16.00/simp/pdam/ramai/lancar/kamis	Tidak Macet
3.	15.00-16.00/perempatan/dieng/ramai lancar/kamis	Tidak macet
4.	06.00-07.00/panji/suroso/ramai/lancar/sabtu	Tidak Macet
5.	06.00-07.00/perempatan/rajabali/normal/sabtu	Tidak Macet
6.	06.00-07.00/tidar/ramai/lancar/kamis	Tidak Macet
7.	06.00-07.00/pertigaan/pln/padat/sabtu	Macet
8.	18.00-19.00/perempatan/dieng/padat/sabtu	Macet
9.	16.00-17.00/depan/araya/padat/sabtu	Macet
10.	07.00-08.00/jalan/a/yani/padat/kamis	Macet
11.	06.00-07.00/pertigaan/pln/padat/kamis	Macet

4.3.3 Proses Pembobotan (*Term Weighting)*

Proses perhitungan pembobotan dilakukan dengan cara menggunakan TF-IDF yang mana IDF dapat mengisyaratkan tingkat keunikan atau tingkat perbedaan kemunculan kata pada suatu kumpulan dokumen yang ada (Nathania, 2018). Berikut merupakan proses pembobotan (*term weighting)* yang dilakukan mulai dari *prepocessing* terhadap data yang digunakan. Data uji yang digunakan dalam proses ini ditunjukkan pada Tabel 4.9.

Tabel 4. 8 Data Uji yang digunakan

No.	*Tweet*	Kategori
1.	06.37 perempatan dieng sabtu normal	Tidak Macet

Berikut merupakan tabel data latih yang didapatkan dari proses *prepocessing* yang ditunjukkan pada Tabel 4.10.

Tabel 4. 9 Data Latih setelah tahapan *Prepocessing*

No.	*Tweet*	Kategori
1.	15.00-16.00/pertigaan/pln/ramai/lancar/kamis	Tidak Macet
2.	15.00-16.00/simp/pdam/ramai/lancar/kamis	Tidak Macet
3.	15.00-16.00/perempatan/dieng/ramai/lancar/kamis	Tidak macet
4.	06.00-07.00/panji/suroso/ramai/lancar/sabtu	Tidak Macet
5.	06.00-07.00/perempatan/rajabali/normal/sabtu	Tidak Macet
6.	06.00-07.00/tidar/ramai/lancar/kamis	Tidak Macet
7.	06.00-07.00/pertigaan/pln/padat/sabtu	Macet

8.	18.00-19.00/perempatan/dieng/padat/sabtu	Macet
9.	16.00-17.00/araya/padat/sabtu	Macet
10.	07.00-08.00/jalan/a/yani/padat/kamis	Macet
11.	06.00-07.00/pertigaan/pln/padat/kamis	Macet

Berikut merupakan data uji yang sudah dilakuakan tahap *prepocessing* yang ditunjukkan pada Tabel 4.11.

Tabel 4. 10 Tabel Data Uji setelah tahapan *Prpocessing*

No.	*Tweet*	Kategori
1.	06.00-07.00/perempatan/dieng/sabtu/normal	Tidak Macet

Kemudian setelah melalui tahapan *prepocessing* dilakukan proses pembobotan (*term weighting).* Langkah awak dari proses pembobotan yaitu menghitung banyaknya kemunculan kata (*term)* dalam setiap dokumen. Berikut merupakan contoh menghitung TF untuk kata (*term)* pertigaan pada d1:

$$W_{tf_{t,d}} = 1 + log_{10} tf_{t,d} = 1 + log_{10} 1 = 1 + 0 = 1$$

Kemudian dilanjutkan dengan menghitung banyaknya dokumen (d) yang di dalamnya terdapat kata (df). Pada tahap ini data uji tidak termasuk dalam perhitungan df. Selanjutnya dilakukan perhitungan IDF. Berikut merupakan contoh menghitung IDF untuk kata (*term*) pertigaan:

$$idf_t = log_{10} \frac{11}{6} = 0{,}56427143$$

Hasil perhitungan TF dan IDF ditunjukkan pada Tabel 4.12.

Tabel 4. 11 Hasil Perhitungan TF dan IDF

Term	TF											DF	IDF	data uji
	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11			
15.00-16.00	1	1	1	0	0	0	0	0	0	0	0	3	0,56427143	0
pertigaan	1	0	0	0	0	0	1	0	0	0	1	3	0,56427143	0
pln	1	0	0	0	0	0	1	0	0	0	1	3	0,56427143	0
ramai	1	1	1	1	0	1	0	0	0	0	0	5	0,34242268	0
lancar	1	1	1	1	0	1	0	0	0	0	0	5	0,34242268	0
kamis	1	1	1	0	0	1	0	0	0	1	1	6	0,26324143	0
simp	0	1	0	0	0	0	0	0	0	0	0	1	1,04139269	0
pdam	0	1	0	0	0	0	0	0	0	0	0	1	1,04139269	0
perempatan	0	0	1	0	1	0	0	1	0	0	0	3	0,56427143	1

dieng	0	0	1	0	0	0	0	1	0	0	0	2	0,74036269	1
06.00-07.00	0	0	0	1	1	1	1	0	0	0	1	5	0,34242268	1
panji	0	0	0	1	0	0	0	0	0	0	0	1	1,04139269	0
suroso	0	0	0	1	1	0	0	0	0	0	0	2	0,74036269	0
sabtu	0	0	0	1	1	0	1	1	1	0	0	5	0,34242268	1
rajabali	0	0	0	0	1	0	0	0	0	0	0	1	1,04139269	0
normal	0	0	0	0	1	0	0	0	0	0	0	1	1,04139269	0
tidar	0	0	0	0	0	1	1	0	0	0	0	1	1,04139269	0
padat	0	0	0	0	0	0	1	1	1	1	1	5	0,34242268	0
18.00-19.00	0	0	0	0	0	0	0	1	0	0	0	1	1,04139269	0
16.00-17.00	0	0	0	0	0	0	0	0	1	0	0	1	1,04139269	0
araya	0	0	0	0	0	0	0	0	1	0	0	1	1,04139269	0
07.00-08.00	0	0	0	0	0	0	0	0	0	1	0	1	1,04139269	0
jalan	0	0	0	0	0	0	0	0	0	1	0	1	1,04139269	0
a	0	0	0	0	0	0	0	0	0	1	0	1	1,04139269	0
yani	0	0	0	0	0	0	0	0	0	0	0	1	1,04139269	0

Kemudian dilanjutkan dengan melakukan perhitungan TF-IDF *Weighting* yang merupakan hasil perkalian TF dan IDF. Berikut merupakan contoh perhitungan TF-IDF *Weighting* pada kata (term) pertigaan pada d1:

$$w_{t,d} = w_{tf_{t,d}} * idf_t = 1 * 0{,}564271 = 0{,}564271$$

Hasil dari perhitungan TF-IDF *Weighting* ditunjukkan pada Tabel 4.13.

Tabel 4. 12 Hasil Perhitngan TF-IDF *Weighting*

Wt,d											
d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11	data uji
0,5643	0,5643	0,5643	0	0	0	0	0	0	0	0	0
0,5643	0	0	0	0	0	0,5643	0	0	0	0,5643	0
0,5643	0	0	0	0	0	0,5643	0	0	0	0,5643	0
0,3424	0,3424	0,3424	0,3424	0	0,3424	0	0	0	0	0	0
0,3424	0,3424	0,3424	0,3424	0	0,3424	0	0	0	0	0	0
0,2632	0,2632	0,2632	0	0	0,2632	0	0	0	0,2632	0,2632	0
0	1,0414	0	0	0	0	0	0	0	0	0	0
0	1,0414	0	0	0	0	0	0	0	0	0	0
0	0	0,5643	0	0,5643	0	0	0,5643	0	0	0	0,5643
0	0	0,7404	0	0	0	0	0,7404	0	0	0	0,7404
0	0	0	0,3424	0,3424	0,3424	0,3424	0	0	0	0,3424	0,3424
0	0	0	1,0414	0	0	0	0	0	0	0	0
0	0	0	0,7404	0,7404	0	0	0	0	0	0	0
0	0	0	0,3424	0,3424	0	0,3424	0,3424	0,3424	0	0	0,3424
0	0	0	0	1,0414	0	0	0	0	0	0	0

0	0	0	0	1,0414	0	0	0	0	0	0	0
0	0	0	0	0	1,0414	1,0414	0	0	0	0	0
0	0	0	0	0	0	0,3424	0,3424	0,3424	0,3424	0,3424	0
0	0	0	0	0	0	0	1,0414	0	0	0	0
0	0	0	0	0	0	0	0	1,0414	0	0	0
0	0	0	0	0	0	0	0	1,0414	0	0	0
0	0	0	0	0	0	0	0	0	1,0414	0	0
0	0	0	0	0	0	0	0	0	1,0414	0	0
0	0	0	0	0	0	0	0	0	1,0414	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Kemudian dilanjutkan dengan perhitungan normalisasi dari hasil perhitungan TF-IDF *Weighting*. Berikut merupakan contoh perhitungan normalisasi kata (*term)* pertigaan pada d1:

$$w_{t,d} = \frac{w_{t,d}}{\sqrt{\sum_{t=1}^{n} {w_{t,d}}^2}} = \frac{0{,}564271}{\sqrt{6{,}023675429}} = 0{,}449278$$

Hasil perhitungan normalisasi dari hasil perhitungan TF-IDF *Weighting* ditunjukkan pada Tabel 4.14.

Tabel 4. 13 Hasil Normalisasi Perhitungan TF-IDF *Weighting*

Normalisasi											
d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11	data uji
0,4493	0,3377	0,4625	0	0	0	0	0	0	0	0	0
0,4493	0	0	0	0	0	0,3919	0	0	0	0,5818	0
0,4493	0	0	0	0	0	0,3919	0	0	0	0,5818	0
0,2726	0,205	0,2806	0,2362	0	0,4202	0	0	0	0	0	0
0,2726	0,205	0,2806	0,2362	0	0,4202	0	0	0	0	0	0
0,2096	0,1576	0,2157	0	0	0,323	0	0	0	0,1419	0,2714	0
0	0,6233	0	0	0	0	0	0	0	0	0	0
0	0,6233	0	0	0	0	0	0	0	0	0	0
0	0	0,4625	0	0,312	0	0	0,3817	0	0	0	0,5378
0	0	0,6068	0	0	0	0	0,5008	0	0	0	0,7056
0	0	0	0,2362	0,1894	0,4202	0,2378	0	0	0	0,3531	0,3263
0	0	0	0,7183	0	0	0	0	0	0	0	0
0	0	0	0,5107	0,4094	0	0	0	0	0	0	0
0	0	0	0,2362	0,1894	0	0,2378	0,2316	0,2209	0	0	0,3263
0	0	0	0	0,5759	0	0	0	0	0	0	0
0	0	0	0	0,5759	0	0	0	0	0	0	0
0	0	0	0	0	1,2778	0,7233	0	0	0	0	0
0	0	0	0	0	0	0,2378	0,2316	0,2209	0,1846	0,3531	0

0	0	0	0	0	0	0	0,7044	0	0	0	0
0	0	0	0	0	0	0	0	0,6717	0	0	0
0	0	0	0	0	0	0	0	0,6717	0	0	0
0	0	0	0	0	0	0	0	0	0,5615	0	0
0	0	0	0	0	0	0	0	0	0,5615	0	0
0	0	0	0	0	0	0	0	0	0,5615	0	0
0	0	0	0	0	0	0	0	0	0	0	0

4.3.4 Proses Klasifikasi *Improved K-Nearest Neighbor (KNN)*

Hasil perhitungan normalisasi dari TF-IDF *Weighting* akan digunakan dalam proses klasifikasi *Improved K-Nearest Neighbor (KNN)*. Langkah awal yang dilakukan dalam tahapan proses klasifikasi *Improved K-Nearest Neighbor (KNN)* adalah menghitung *cosine similiarity. Cosine similiarity* merupakan hasil perkalian data latih dengan data uji yang ingin diketahui hasil kategorinya. Berikut merupakan contoh perhitungan *cosine similiarity* dari d1 pada kata (*term)* pertigaan terhadap data uji:

$$Cosine\ (d_i, q_i) = 0{,}4493 * 0 = \ 0$$

Hasil perhitungan *cosine similiarity* ditunjukkan pada Tabel 4.15.

Tabel 4. 14 Hasil Perhitungan *Cosine Similiarity*

Cosine Similiarity										
d1 x d.uji	d2 x d.uji	d3 x d.uji	d4 x d.uji	d5 x d.uji	d6 x d.uji	d7 x d.uji	d8 x d.uji	d9 x d.uji	d10 x d.uji	d11 x d.uji
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0,2487	0	0,1678	0	0	0,2053	0	0	0
0	0	0,4281	0	0	0	0	0,3534	0	0	0
0	0	0	0,0771	0,0618	0,1371	0,0776	0	0	0	0,1152
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0,0771	0,0618	0	0,0776	0,0756	0,0721	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
JUMLAH										
0	0	0,6768	0,1542	0,2914	0,1371	0,1552	0,6342	0,0721	0	0,1152

Langkah selanjutnya dari perhitungan *cosine similiarity* akan dilakukan pengurutan tingkat kemiripan dari nilai terbesar ke nilai terkecil. Berikut merupakan hasil urutan tingkat kemiripan dari nilai *cosine similiarity* yang ditunjukkan pada Tabel 4.16.

Tabel 4. 15 Urutan Tingkat Kemiripan terhadap Data Uji

Peringkat		
d3	0,6768236	Tidak Macet
d8	0,6341923	Macet
d5	0,2913896	Tidak Macet
d7	0,1552194	Macet
d4	0,1541602	Tidak Macet
d6	0,1371105	Tidak Macet
d11	0,1152174	Macet
d9	0,0720775	Macet
d1	0	Tidak Macet
d2	0	Tidak Macet
d10	0	Macet

Langkah selanjutnya yaitu menghitung nilai *k-values* baru (n) untuk masing-masing kategori. Nilai *k-values* awal ditetapkan senilai 6 untuk masing-masing kategori. Berikut merupakan tabel jumlah data latih yang digunakan dalam penelitian ini yang ditunjukkan pada Tabel 4.17.

Tabel 4. 16 Jumlah Data Latih

Data Latih		
Tidak Macet	Macet	Jumlah
6	5	11

Dari Tabel 4.17 akan digunakan untuk menghitung nilai *k-values* baru untuk masing-masing kategori. Berikut merupakan perhitungan nilai *k-values* baru untuk masing-masing kategori:

1. Kategori Tidak Macet

$$n = \left[\frac{6 * 6}{6}\right] = 6$$

Jadi, nilai *k-values* baru untuk kategori tidak macet adalah 6.

2. Kategori Macet

$$n = \left[\frac{6 * 5}{6}\right] = 5$$

Jadi, nilai *k-values* baru untuk kategori macet adalah 5.

Dari nilai *k-values* baru untuk masing-masing kategori tersebut kemudian akan dilanjutkan dengan menghitung nilai probabilitas dokumen uji dari masing-masing kategori. Berikut merupakan contoh perhitungan probabilitas dari masing-masing kategori dari dokumen uji:

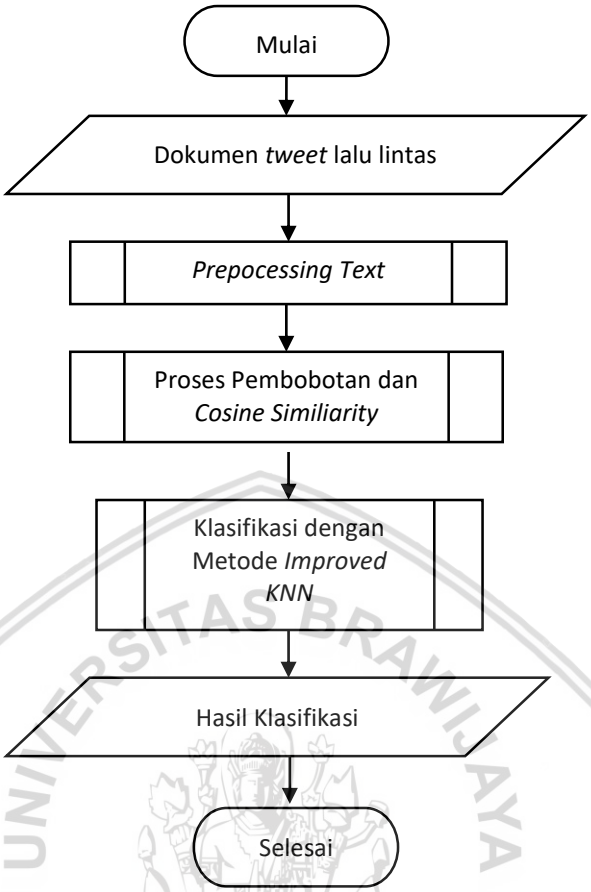
$$p(x, c_{TidakMacet}) = \frac{\big((0{,}676823 * 1) + (0{,}634192 * 0) + (0{,}291389 * 1) + (0{,}155219 * 0) + (0{,}154160 * 1) + (0{,}137110 * 1)\big)}{(0{,}676823 + 0{,}634192 + 0{,}291389 + 0{,}155219 + 0{,}154160 + 0{,}137110)}$$
$$= 2{,}697025$$

$$p(x, c_{Macet}) = \frac{\big((0{,}676823 * 0) + (0{,}634192 * 1) + (0{,}291389 * 0) + (0{,}155219 * 1) + (0{,}154160 * 0)\big)}{(0{,}676823 + 0{,}634192 + 0{,}291389 + 0{,}155219 + 0{,}154160)}$$
$$= 2{,}024373$$

Dari hasil perhitungan probabilitas tersebut dapat disimpulkan bahwa dokumen uji termasuk ke dalam kategori Tidak Macet dengan nilai probabilitas sebesar 2,697025.

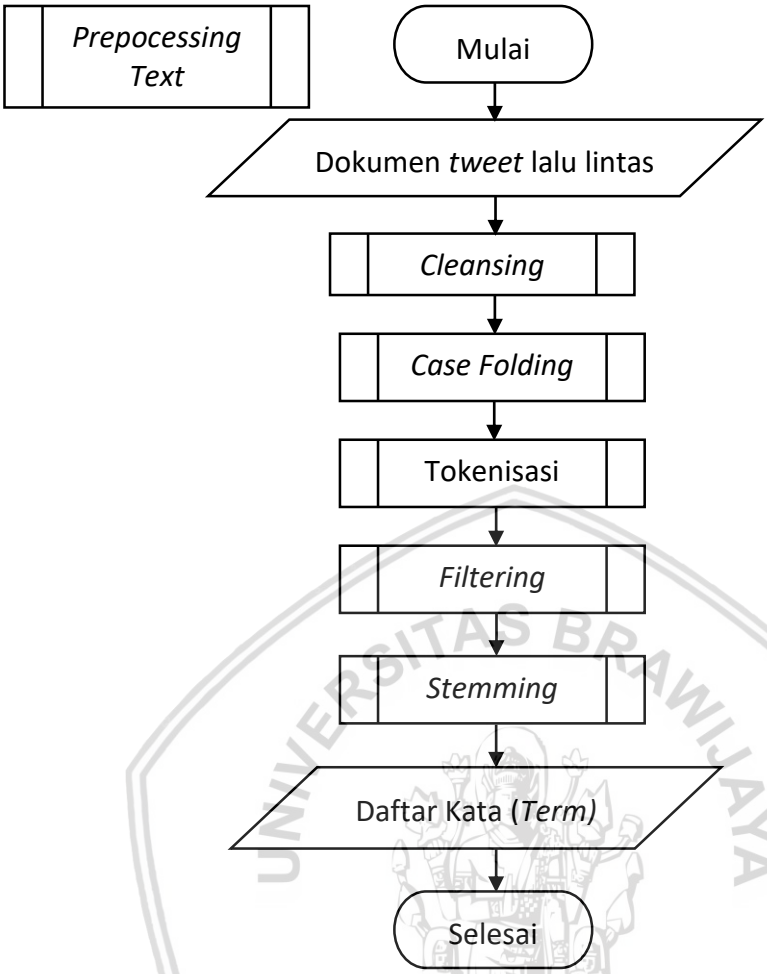
4.4 Diagram Alir Sistem

Dalam tahapan ini akan dijelaskan langkah-langkah yang dilakukan dalam proses penelitian ini. Pada Gambar 4.2 dijelaskan gambaran umum sistem pada penelitian ini, yaitu memasukkan dokumen berupa *tweet* yang akan digunakan, dilanjutkan dengan proses *prepocesing text*, proses pembobotan (*term weighting)*, proses perhitungan *cosine similiarity* dan proses klasifikasi dengan menggunakan metode *Improved K-Nearest Neighbor* (KNN). Hasil keluaran dari penelitian ini yaitu berupa hasil kategori Tidak Macet dan Macet dari tahapan-tahapan yang sudah dilakukan.



Gambar 4. 2 Diagram Alir Sistem

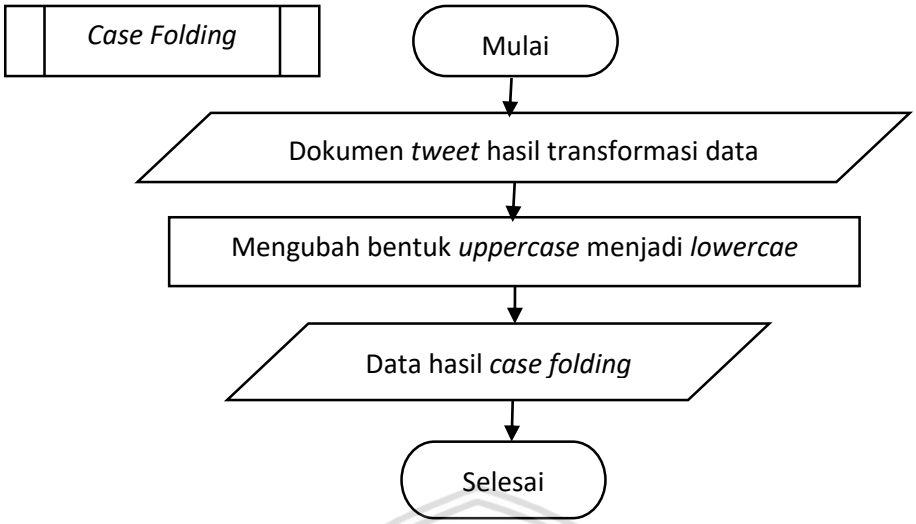
Pada Gambar 4.2 dijelaskan tentang diagram alur sistem pada proses *prepocessing text*. Dalam tahapan *prepocessing text* terdapat beberapa tahapan yang dilakukan dimulai dari tahapan *cleansing,* transformasi data, tokenisasi, *filtering* dan tahapan terakhir yaitu *stemming.* Dimana proses *cleansing* digunakan untuk mengurangi *noise* yang ada pada dokumen, *case folding* digunakan untuk mengubah huruf yang berupa *uppercase* diubah menjadi huruf yang berupa *lowercase*, tokenisasi digunakan untuk memotong setiap kata yang terdapat dalam kalimat dengan menggunakan spasi, *filtering* digunakan untuk menghilangkan kata yang tidak diperlukan dan proses *stemming* digunakan untuk mengubah bentuk kata menjadi kata dasar. Hasil dari *prepocessing text* yaitu berupa kata (*term)* yang akan digunakan dalam proses selanjutnya yaitu proses pembobotan (*tem weighting)*, perhitungan normalisasi, perhitungan *cosine similiarity* hingga proses kalsifikasi dengan menggunakan metode *Improved K-Nearest Neighbor* (KNN).



Gambar 4. 3 Diagram Alir *Prepocessing Text*

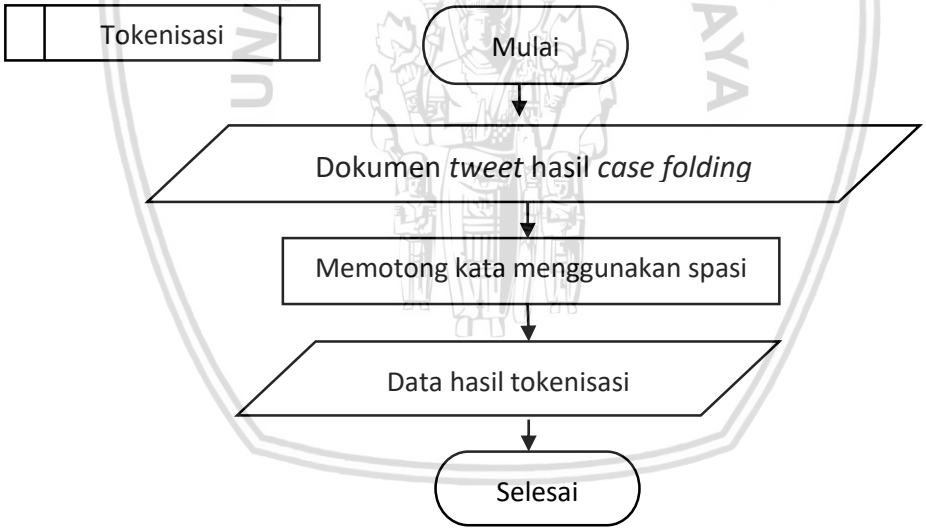
Pada Gambar 4.3 ditunjukkan diagram alir dari tahapan *cleansing. Cleansing* merupakan tahapan yang bertujuan untuk mengurangi *noise* yang terdapat pada data. Pada tahapan ini dilakukan proses penghapusan URL, *hashtag* (#aaa), *username* (@aaa), angka, karakter selain huruf alfabet dan tanda baca.

Pada Gambar 4.4 ditunjukkan diagram alir tahapan *case folding*. *Case folding* merupakan proses mengubah bentuk huruf berupa *uppercase* menjadi *lowercase.*



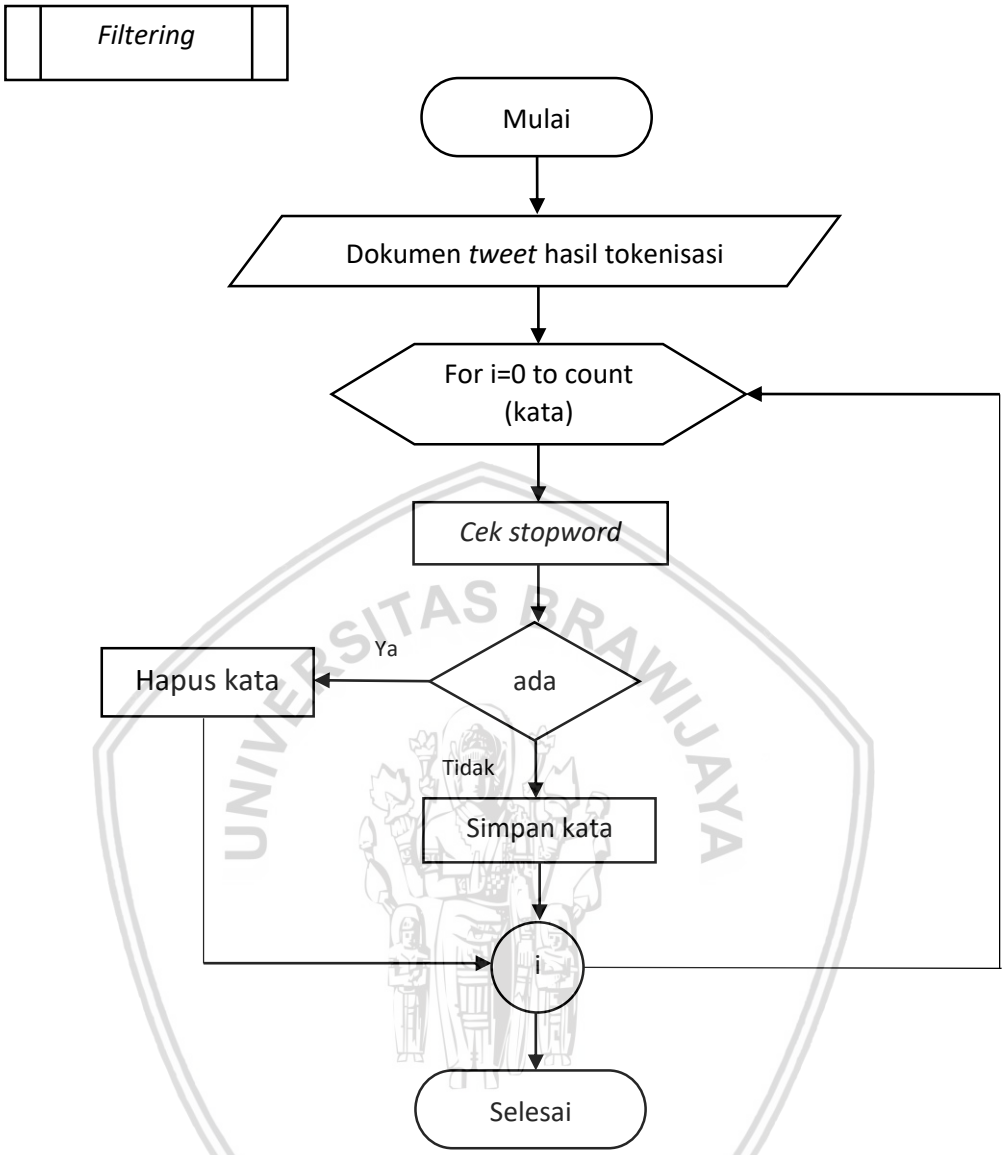
Gambar 4. 4 Diagram Alir *Case Folding*

Pada Gambar 4.5 ditunjukkan diagram alir dari tahapan tokenisasi. Tokenisasi merupakan tahapan memotong setiap kata yang terdapat dalam kalimat dengan menggunakan spasi.



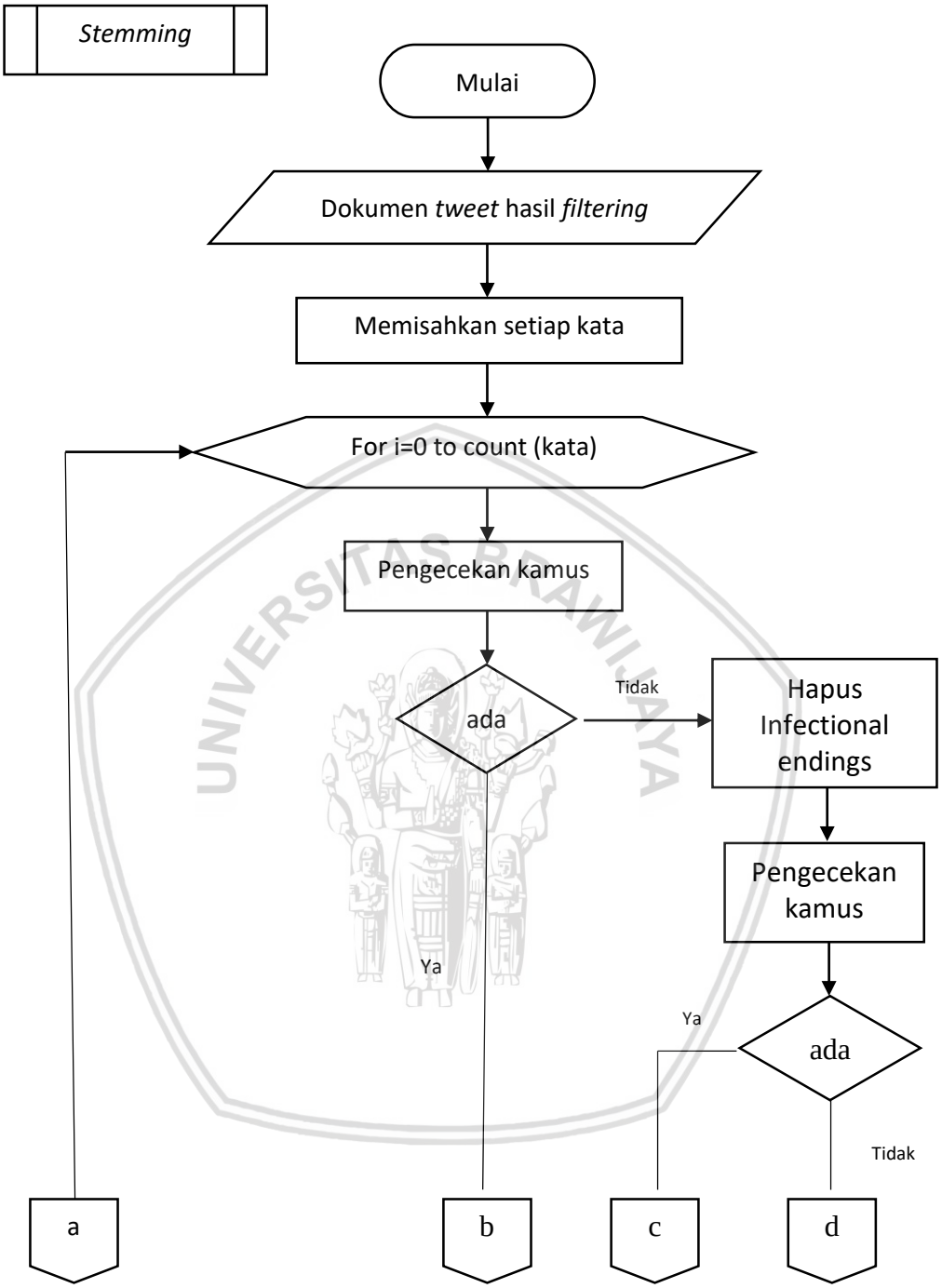
Gambar 4. 5 Diagram Alir Tokenisasi

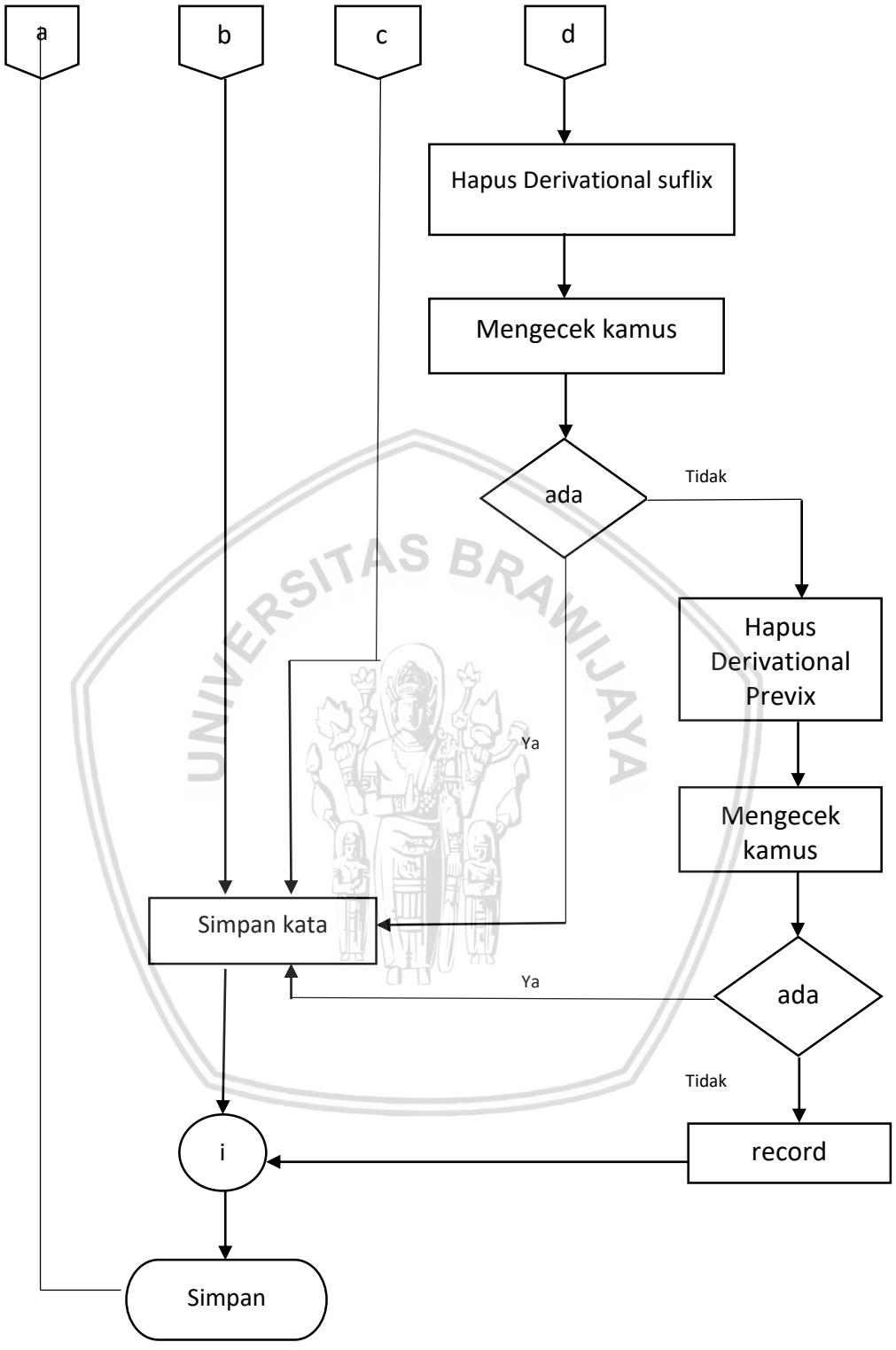
Pada Gambar 4.6 ditunjukkan diagram alir *Filtering. Filtering* merupakan tahapan menghilangkan kata yang dianggap tidak penting atau kata yang tidak bermakna yang termasuk dalam *stoplist*. Kata-kata yang ada pada dokumen akan dicocokkan dengan daftar *stoplist*, jika pada dokumen terdapat kata yang ada pada *stoplist* maka akan kata tersebut akan dihilangkan dan jika pada dokumen kata-katany tidak terdapat pada *stoplist* maka akan dibiarkan dan dilanjutkan ke proses selanjutnya.



Gambar 4. 6 Diagram Alir *Filtering*

Pada Gambar 4.7 ditunjukkan digram alir dari proses *stemming*. *Stemming* merupakan proses untuk mengubah bentuk kata menjadi kata dasar. Tahapan yang dilakukan pada proses *stemming* yaitu menggunakan algortima Nazief dan Andriani. Proses *stemming* dalam algoritma ini mempunyai beberapa langkah yaitu menghapus *Infectional suffixes, Derivational suffix, Derivational Previx* dan pada setiap tahapan dilakukan pengecekan dari daftar kata dasar yang digunakan. Apabila tidak ditemukan kata dasar setelah melakukan semua tahapan yang dibutuhkan, maka kata tersebut akan kembali seperti sebelum dilakukan proses *stemming*.

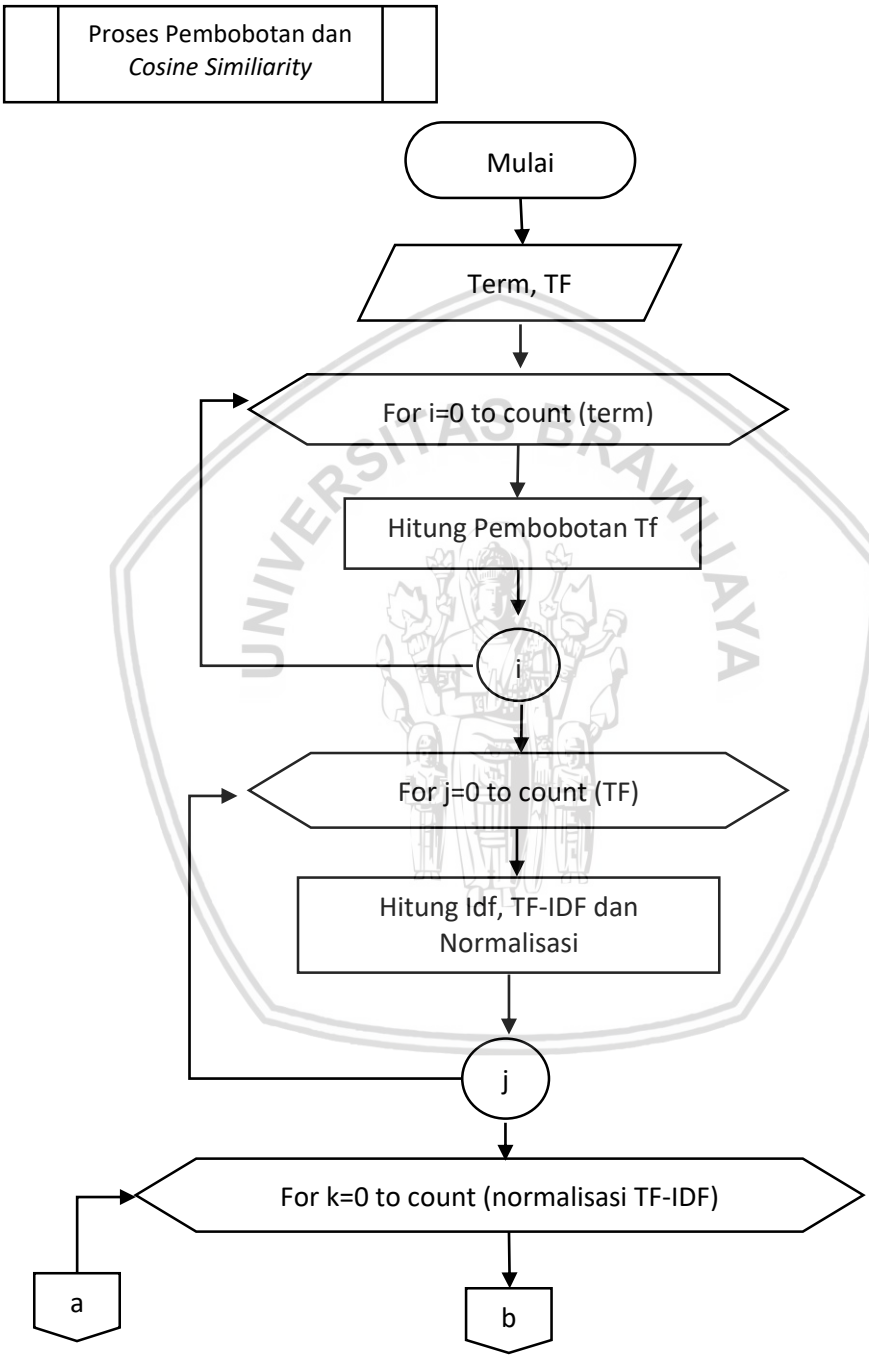

Stemming
Mulai
Dokumen *tweet* hasil *filtering*
Memisahkan setiap kata
For i=0 to count (kata)
Pengecekan kamus
ada
Tidak
Hapus Infectional endings
Pengecekan kamus
ada
Ya
Ya
Tidak
a
b
c
d

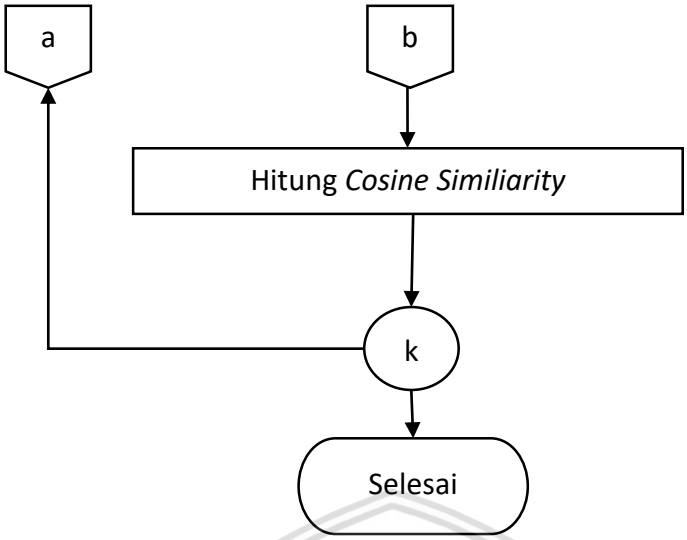


Gambar 4. 7 Diagram Alir *Stemming*

Pada Gambar 4.8 ditunjukkan diagram alir dari proses perhitungan TF-IDF dan *Cosine Similiarity*. Langkah awal pada proses ini yaitu memasukkan kata (term) yang sudah melalui tahapan *stemming*, kemudian dilanjutkan dengan proses permbobotan (*term weighting)*. Proses pembobotan dilakukan dengan

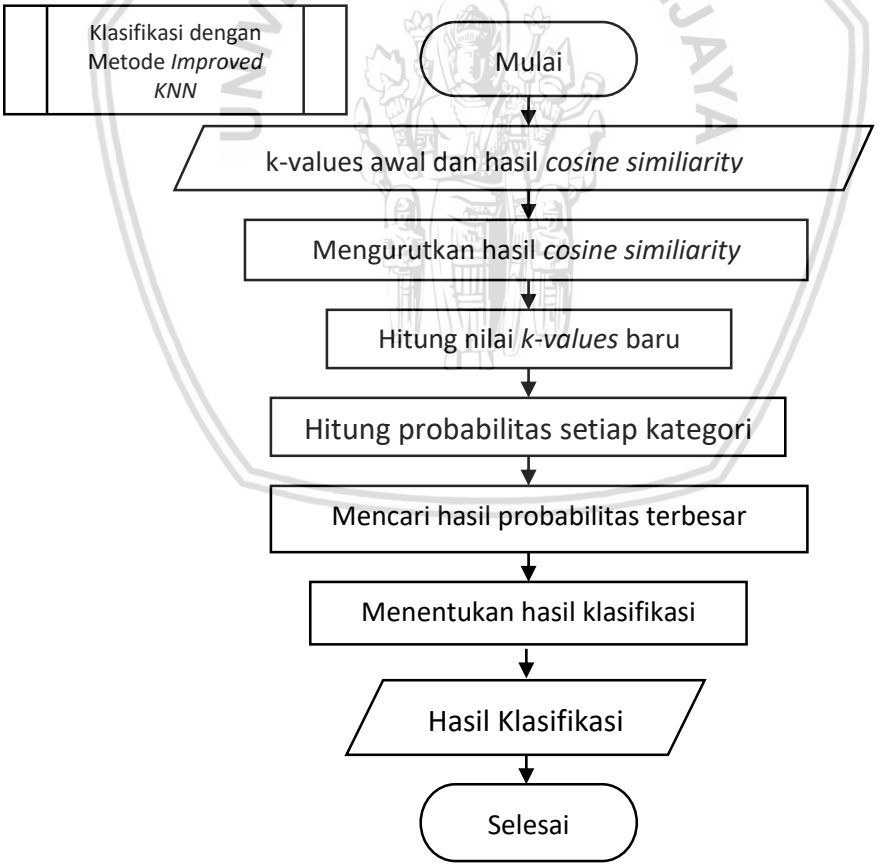
menghitung pembobotan TF dari hasil TF. Kemudian dilanjutkan dengan perhitungan TF-IDF yang didapatkan dari hasil dokumen latih dikalikan dengan hasil pembobotan TF. Langkah selanjutnya yaitu menghitung normalisasi dari proses perhitungan TF-IDF. Proses normalisasi dilakukan untuk memudahkan dalam proses perhitungan selanjutnya yaitu perhitungan *cosine similiarity*.





Gambar 4. 8 Diagram Alir Perhitungan TF-IDF dan *Cosine Similiarity*

Pada Gambar 4.9 ditunjukkan diagram alir dari klasifikasi dengan menggunakan metode *Improved K-Nearest Neighbor* (KNN). Proses klasifikasi bisa dilakukan setelah melakukan proses perhitungan *cosine similiarity*.



Gambar 4. 9 Diagram Alir Klasifikasi *Improved K-Nearest Neighbor*

Proses klasifikasi dengan *Improved K-Nearest Neighbor* dilakukan dengan beberapa langkah yaitu diawali dengan input nilai *k-values* awal dan hasil *cosine*

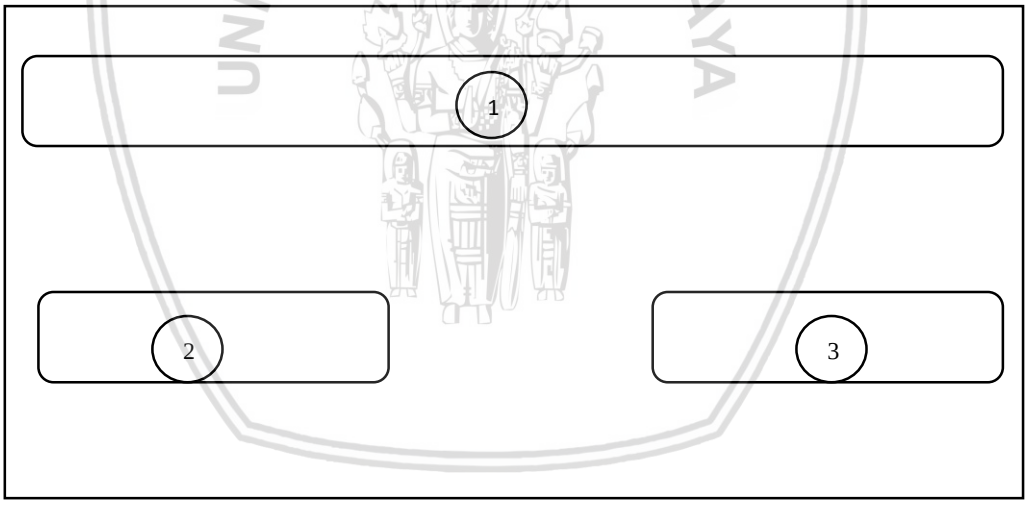
similiarity yang didapatkan dari hasil normalisasi perhitungan TF-IDF. Kemudian hasil dari *cosine similiarity* diurutkan. Hitung nilai *k-values* baru dari masing-masing kategori yang ada. Setelah mendapatkan nilai *k-values* baru, hitung probabilitas masing-masing kategori yang kemudian dipilih nilai probabilitas terbesar. Hasil klasifikasi didapatkan dari pemilihan nilai probabilitas terbesar dari dokumen yang ada sebelumnya.

4.5 Perancangan Antarmuka (*User Interface)*

Perancangan antarmuka atau *user interface (*UI) digunakan untuk menghubungkan interaksi antara sistem dengan pengguna. Perancangan yang dibuat meliputi perancangan antarmuka halaman awal, perancangan antarmuka data latih, perancangan antarmuka pengujian dan perancangan antar muka hasil pengujian.

4.5.1 Perancangan Antarmuka Halaman Awal

Perancangan antarmuka halaman awal yang ditunjukkan pada Gambar 4.10 akan menampilkan judul dari sistem, menu data latih yang digunakan untuk melihat data latih yang ada dan menu pengujian yang digunakan untuk melakukan pengujian dari pengguna dari data uji yang ditentukan.



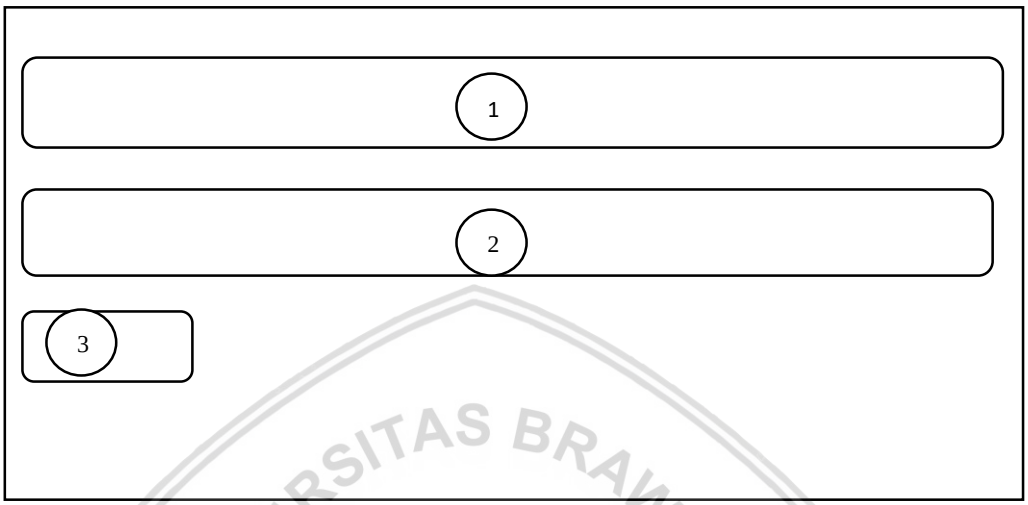
Gambar 4. 10 Perancangan Antarmuka alaman Awal

Keterangan:

1. Judul dari aplikasi
2. Tombol menu halaman pengguna
3. Tombol menu halaman pengujian

4.5.2 Perancangan Antarmuka Halaman Pengguna

Perancangan antarmuka halaman pengguna yang ditunjukkan pada Gambar 4.11 akan menampilkan informasi tentang data latih yang digunakan dalam proses klasifikasi.



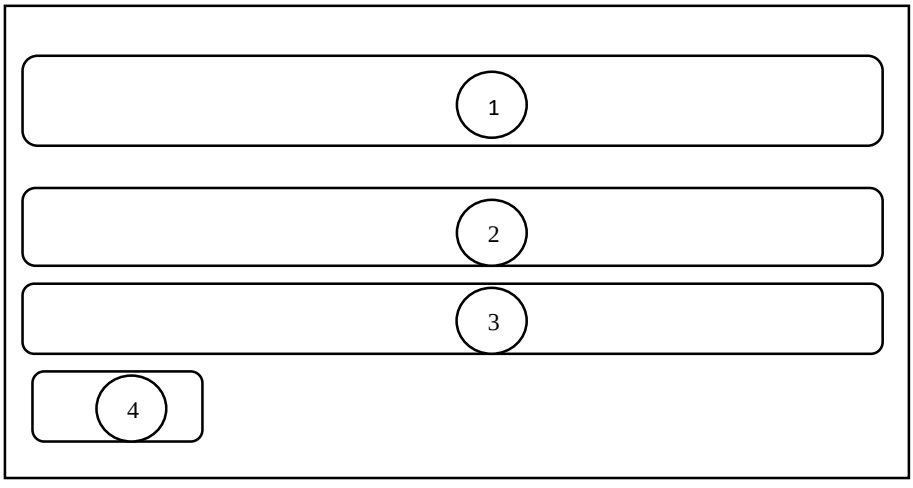
Gambar 4. 11 Perancangan Antarmuka Halaman Pengguna

Keterangan:

1. Judul Aplikasi
2. Kolom untuk memasukkan *tweet*
3. Tombol submit

4.5.3 Perancangan Antarmuka Pengujian

Perancangan antarmuka pengujian yang ditunjukkan pada Gambar 4.12 digunakan untuk melakukan pengujian dengan memasukkan dokumen *tweet* sebagai data uji yang akan digunakan sehingga akan diproses untuk mendapatkan hasil klasifikasi berdasarkan kategori yang ada.



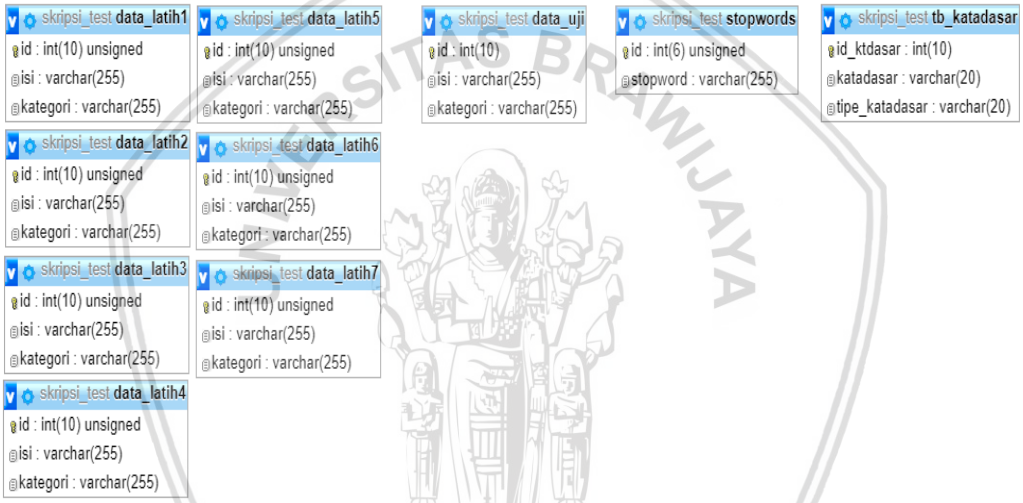
Gambar 4. 12 Perancangan Antarmuka Pengujian

Keterangan:

1. Judul aplikasi
2. Kolom memasukkan data uji berupa *tweet*
3. Kolom untuk memasukkan skenario
4. Tombol submit

4.6 Perancangan Database

Perancangan database dibangun untuk menyimpan data yang berupa data latih dan data uji, data dari hasil *prepocessing text* hingga data dari hasil perhitungan yang telah dilakukan. Tabel dari perancangan database ditunjukkan pada Gambar 4.13.



Gambar 4. 13 Perancangan Database

4.6.1 Tabel Data Latih1

Tabel Data Latih1 merupakan tabel yang digunakan untuk melakukan penyimpanan berupa data latih dengan jumlah 300 data latih yan terdiri dari 100 data macet dan 200 data tidak macet. Pada Tabel 4.17 ditunjukkan struktur dari tabel data latih.

Tabel 4. 17 Tabel Data Latih1

No.	Nama Field	Type	Size
1.	Id	Int	10
2.	Isi	Varchar	255
3.	Kategori	varchar	255

4.6.2 Tabel Data Latih2

Tabel Data Latih2 merupakan tabel yang digunakan untuk melakukan penyimpanan berupa data latih dengan jumlah 365 data latih yan terdiri dari 115 data macet dan 250 data tidak macet. Pada Tabel 4.18 ditunjukkan struktur dari tabel data latih.

Tabel 4. 18 Tabel Data Latih2

No.	Nama Field	Type	Size
1.	Id	Int	10
2.	Isi	Varchar	255
3.	Kategori	varchar	255

4.6.3 Tabel Data Latih3

Tabel Data Latih3 merupakan tabel yang digunakan untuk melakukan penyimpanan berupa data latih dengan jumlah 475 data latih yan terdiri dari 125 data macet dan 350 data tidak macet. Pada Tabel 4.19 ditunjukkan struktur dari tabel data latih.

Tabel 4. 19 Tabel Data Latih3

No.	Nama Field	Type	Size
1.	Id	Int	10
2.	Isi	Varchar	255
3.	Kategori	varchar	255

4.6.4 Tabel Data Latih4

Tabel Data Latih4 merupakan tabel yang digunakan untuk melakukan penyimpanan berupa data latih dengan jumlah 550 data latih yan terdiri dari 150 data macet dan 400 data tidak macet. Pada Tabel 4.20 ditunjukkan struktur dari tabel data latih.

Tabel 4. 20 Tabel Data Latih2

No.	Nama Field	Type	Size
1.	Id	Int	10
2.	Isi	Varchar	255
3.	Kategori	varchar	255

4.6.5 Tabel Data Latih5

Tabel Data Latih5 merupakan tabel yang digunakan untuk melakukan penyimpanan berupa data latih dengan jumlah 600 data latih yan terdiri dari 185

data macet dan 415 data tidak macet. Pada Tabel 4.21 ditunjukkan struktur dari tabel data latih.

Tabel 4. 21 Tabel Data Latih5

No.	Nama Field	Type	Size
1.	Id	Int	10
2.	Isi	Varchar	255
3.	Kategori	varchar	255

4.6.6 Tabel Data Latih6

Tabel Data Latih6 merupakan tabel yang digunakan untuk melakukan penyimpanan berupa data latih dengan jumlah 400 data latih yan terdiri dari 200 data macet dan 200 data tidak macet. Pada Tabel 4.22 ditunjukkan struktur dari tabel data latih.

Tabel 4. 22 Tabel Data Latih6

No.	Nama Field	Type	Size
1.	Id	Int	10
2.	Isi	Varchar	255
3.	Kategori	varchar	255

4.6.7 Tabel Data Latih7

Tabel Data Latih7 merupakan tabel yang digunakan untuk melakukan penyimpanan berupa data latih dengan jumlah 300 data latih yan terdiri dari 200 data macet dan 100 data tidak macet. Pada Tabel 4.23 ditunjukkan struktur dari tabel data latih.

Tabel 4. 23 Tabel Data Latih7

No.	Nama Field	Type	Size
1.	Id	Int	10
2.	Isi	Varchar	255
3.	Kategori	varchar	255

4.6.8 Tabel Data Uji

Tebel Data Uji merupakan tabel yang digunakan untuk melakukan penyimpanan berupa data uji yang digunakan pada penelitian ini. Pada Tabel 4.24 ditunjukkan struktur dari tabel data uji.

Tabel 4. 24 Tabel Data Uji

No.	Nama Field	Type	Size

1.	Id	Int	10
2.	Isi	Varchar	255
3.	Kategori	varchar	255

4.6.9 Tabel *Stopwords*

Tebel *stopwords* merupakan tabel yang digunakan untuk melakukan penyimpanan berupa kata-kata yang dianggap tidak penting pada penelitian ini. Pada Tabel 4.25 ditunjukkan struktur dari tabel data pengujian.

Tabel 4. 25 Tabel *Stopwords*

No.	Nama Field	Type	Size
1.	id	Int	6
2.	stopword	Varchar	255

4.6.10 Tabel Kata Dasar

Tebel kata dasar merupakan tabel yang digunakan untuk melakukan penyimpanan berupa kata-kata dasar yang digunakan pada penelitian ini. Pada Tabel 4.26 ditunjukkan struktur dari tabel data pengujian.

Tabel 4. 26 Tabel *Stopwords*

No.	Nama Field	Type	Size
1.	Id_katadasar	Int	10
2.	katadasar	Varchar	20
3.	Tipe_katadasar	varchar	20

4.7 Perancangan Pengujian dan Analisis

Perancangan pengujian merupakan perancangna yang digunakan untuk bisa mengetahui adanya atau tidak kesalahan yang ada pada saat pengimplementasian metode *Impoved K-Nearest Neighbor*. Untuk pengujian analisis akan digunakan tabel *confusion matrix* guna mempermudah proses perhitungan *Precision, Recall, F-Measure* dan tingkat akurasi. Dari proses pengujian analisis nantinya akan diketahui faktor-faktor yang mempengaruhi ketepatan dari hasil klasifikasi menggunakan metode *Improved K-Nearest Neigbor*. Pada Tabel 4.27 ditunjukkan perancangan tabel skenario. Pengujian ini dilakukan dengan menggunakan beberapa skenario dengan jumlah data latih berbeda-beda. Pada pengujian masing-masing skenario dilakukan dengan menggunakan nilai *k-values* awal yang sama. Pada Tabel 4.28 akan ditunjukkan tabel perancangan pengujian.

Tabel 4. 27 Perancangan Tabel Skenario

Skenario	Data Latih			Data Uji		
	Tidak Macet	Macet	Jumlah	Tidak Macet	Macet	Jumlah
1.						
2.						
3.						
4.						
5.						

Tabel 4. 28 Perancangan Pengujian

k-values	*k-values baru*		*Precision*	*Recall*	*F-Measure*	Akurasi
	Tidak Macet	Macet				

4.8 Kesimpulan

Kesimpulan akan didapatkan setelah proses-proses dalam tahapan klasifikasi telah dilakukan yaitu tahapan perancangan, implementasi dan pengujian. Penarikan kesimpulan ini didapatkan dari hasil analisa pengujian yang telah dilakukan dengan harapan adanya saran yang bisa berguna untuk memperbaiki kekurangan yang ada pada penelitian ini.

4.9 Spesifikasi Sistem

Untuk membuat sistem yang memiliki fungsi sesuai dengan kebutuhan yang dibutuhkan maka pengimplementasian sistem mengacu pada proses dan hasil analisis kebutuhan dan perancangan yang telah dibahas pada bab sebelumnya. Spesifikasi sistem dibagi menjadi dua yakni spesifikasi *hardware* (perangkat keras) dan *software* (perangkat lunak).

1.1.1 Spesifikasi Perangkat Keras

Pada proses pengembangan sistem Klasifikasi Kemacetan Lalu Lintas Di Kota Malang Pada Sosial Media Twitter Menggunakan Metode *Improved K-Nearest Neighbor* ini menggunakan komputer yang memiliki spesifikasi *hardware* (perangkat keras) sebagai berikut :

a. Processor Intel Core i3 4030U, 1,9GHz

b. Kapasitas Memori (RAM) 2.00 GB

1.1.2 Spesifikasi Perangkat Lunak

Pada proses pengembangan sistem Klasifikasi Kemacetan Lalu Lintas Di Kota Malang Pada Sosial Media Twitter Menggunakan Metode *Improved K-Nearest Neighbor* ini menggunakan komputer yang memiliki spesifikasi *software* (perangkat lunak) sebagai berikut :

a. OS Windows 10 Profesional 64 bit

b. Bahasa Pemrograman PHP dan HTML

c. XAMPP v3.2.2

d. Notepad ++ Text Editor

e. Google Chrome

1.2 Batasan-batasan Implementasi

Batasan-batasan implementasi merupakan batasan-batasan yang mengacu pada kemampuan yang yang dimiliki oleh sistem yang dibangun. Batasan-batasan implementasi ini juga berguna untuk menujukkan ruang lingkup dari sistem yang dibangun. Adapun batasan-batasan implementasi sistem Klasifikasi Kemacetan Lalu Lintas Di Kota Malang Pada Sosial Media Twitter Menggunakan Metode *Improved K-Nearest Neighbor* adalah sebagai berikut:

a. Klasifikasi Kemacetan Lalu Lintas Di Kota Malang Pada Sosial Media Twitter Menggunakan Metode *Improved K-Nearest Neighbor* dapat diakses dengan aplikasi berbasis web.

b. Metode yang digunakan adalah metode *Improves K-Nearest Neighbor (K-NN)*.

c. Data latih dan data uji yang digunakan merupakan data yang berasal dari Twitter yang berupa *tweet* dan *retweet* dari akun @PuspitaFM.

d. *Output* yang dihasilkan berupa hasil klasifikasi kemacetan lalu lintas, yaitu macet dan tidak macet.

e. Penentuan klasifikasi berasal dari *tweet* itu sendiri.

4.10 Implementasi

Sistem Klasifikasi Kemacetan Lalu Lintas Di Kota Malang Pada Sosial Media Twitter Menggunakan Metode *Improved K-Nearest Neighbor* terdiri dari beberapa proses yaitu mulai dari *preprocessing*, *term weighting*, *cosine similiarity* hingga klasifikasi dengan metode *Improve K-NN*.

4.10.1 *Preprocessing*

Proses *preprocessing* terdiri dari beberapa tahapan yaitu, *cleansing* yang merupakan proses untuk mengurangi *noise* yang berupa URL, *hashtag* (#aaa),

username (@aaa), angka, karakter selain huruf alfabet dan tanda baca yang ada pada data*, case folding* yang merupakan tahapn untuk mengubah bentuk semua huruf menjadi *lowercase,* tokenisasi, *filtering* yang merupakan proses menghapus kata-kata yang tidak diperlukan*,* dan *stemming* yang merupakan mengubah bentuk kata berimbuhan menjadi kata dasar.

4.10.1.1 Cleansing

Pada tahapan ini bertujuan untuk mengurangi *noise* yang terdapat pada data. Pada tahapan ini dilakukan proses penghapusan URL, *hashtag* (#aaa), *username* (@aaa), angka, karakter selain huruf alfabet dan tanda baca. Pada *Source code* 5.1 ditunjukkan implementasi dari proses *cleansing.*

```
1   $tag = substr($kata, 0, 1);
2         $link = substr($kata, 0, 4);
3         // echo "link: ".$link."<br/>";
4         if ($tag === '@' || $tag === '#'){
5               return '';
6         }
7         if($link === 'www.' || $link === 'http'){
8         // echo "masuk!";
9               return '';
10        }
```

Source code* 5.1 Implementasi Tahap *Cleansing

4.10.1.2 Transformasi Data

Pada tahapan ini bertujuan untuk mengubah teks yang berisi tentang informasi waktu menjadi kategori rentang waktu dan mengubah teks yang berisi informasi tentang tanggal menjadi informasi hari. Tahapan ini juga dilakukan guna mengganti kata-kata singkatan dan typo menjadi kata baku sehingga ke depannya memudahkan proses selanjutnya. Pada *Source code* 5.2 ditunjukkan implementasi dari proses transformasi data.

```
1   $time = substr($kata, 0, 2);
2   $isNumeric = is_numeric($time);
3   if($isNumeric && strlen($kata) == 5){
4         $hasil = $time + 1;
5         $kata = sprintf('%02d', $time) . '.00-' .
6   sprint('%02d', $hasil) . '.00';
7               return  $kata;
8         }
9
10  if($kata == 'per3an'){
11        $kata = 'pertigaan';
12        }
13        if($kata == 'per4an'){
14        $kata = 'perempatan';
15        }
16        if($kata == 'simp'){
17        $kata = 'simpang';
18        }
```

***Source code 5.2* Implementasi Transformasi Data**

4.10.1.3 Case Folding

Pada tahapan ini bertujuan untuk mengubah inputan yang berupa huruf kapital (*upper case)* menjadi huruf kecil (*lower case)*. Pada *Source code* 5.3 ditunjukkan implementasi dari proses *case folding.*

```
1   $kata = strtolower($kata);
2   $kata = preg_replace('/[^a-z]+/i', '', $kata);
```

Source code* 5.3 Implementasi Tahap *Case Folding

4.10.1.4 Tokenisasi

Pada tahapan ini dilakukan dengan memotong setiap kata yang terdapat dalam kalimat dengan menggunakan spasi yang dijadikan sebagai delimiter yang kemudian menghasilkan token yang berupa kata. Pada *Source code* 5.4 ditunjukkan implementasi dari proses tokenisasi.

```
1   $tokenisasi = explode(' ', $casefolding);
2   $tokenisasi = array_filter($tokenisasi);
3   sort($tokenisasi);
```

***Source code* 5.4 Implementasi Tahap Tokenisasi**

4.10.1.5 Filtering

Pada tahapan ini dilakukan untuk menghilangkan kata yang dianggap tidak penting atau kata yang tidak bermakan yang termasuk dalam *stoplist. Stoplist* merupakan kata-kata yang sering muncul dalam dokumen namun tidak mempunyai kaitan dengan tema tertentu. Pada *Source code* 5.5 ditunjukkan implementasi dari proses *filtering*.

```
1   $stopwords = GetOneStopWords($kata);
2       if(sizeof($stopwords)>0){
3         return '';
4       }
    function GetOneStopWords($kata){
          global $conn;
          $sql = "SELECT * FROM stopwords where stopword =
    '".$kata."'";
          $data =  $conn->query($sql);
          return $data->fetch_assoc();
    }
```

Source code* 5.5 Implementasi Tahap *Filtering

4.10.1.6 Stemming

Pada tahapan ini dilakukan untuk mengubah bentuk kata hasil dari proses *filtering* yang berupa kata imbuhan baik imbujan awal maupun imbuhan akhir menjadi kata dasar. Hasil dari proses *stemming* yaitu berupa *root word*. Pada *Source code* 5.6 ditunjukkan implementasi dari proses *stemming*.

```
1   /* 1. Cek Kata di Kamus jika Ada SELESAI */
2         if(cekKamus($kata)){ // Cek Kamus
3               return $kata; // Jika Ada kembalikan
4         }
5         /* 2. Buang Infection suffixes (\-lah", \-kah", \-
6   ku", \-mu", atau \-nya") */
```

```
7         $kata = Del_Inflection_Suffixes($kata);
8
9       /* 3. Buang Derivation suffix (\-i" or \-an") */
10      $kata = Del_Derivation_Suffixes($kata);
11
12      /* 4. Buang Derivation prefix */
13      $kata = Del_Derivation_Prefix($kata);
```

Source code* 5.6 Implementasi Tahap *Stemming

4.10.2 Pembobotan (*Term Weighting*)

Pembobotan (*term weighting)* dilakukan dengan cara menggunakan TF.IDF. Proses ini bertujuan untuk mengidentifikasi kemunculan dan keunikan setiap kata. Pada *Source code* 5.7 ditunjukkan implementasi dari proses pembobotan (*term weighting)*.

```
1   function tf_idf ($list_kata, $hasil_data_latih){
2         $tf_idf = array ();
3         foreach ($list_kata as $kata) {
4               $count = array();
5               $df = 0;
6               foreach ($hasil_data_latih as $data) {
7                     if(strlen($kata)>2){
8                           $cnt = substr_count($data, $kata);
9                     }
10                    else{
11                          $cnt = substr_count($data, "
12  ".$kata." ");
13                    }
14                    $df = $df + $cnt ;
15                    array_push($count, $cnt);
16              }
17              if($df==0){
18                    $df = 1;
19              }
20              $idf = round(log(sizeof($hasil_data_latih) /
21  $df, 10), 4);
22              array_push($count, $df);
23              array_push($count, $idf);
24              array_push($tf_idf, $count);
25        }
26        return $tf_idf;
27  }

28  //langkah - 4
29  function wtd ($tf_idf){
30        $wtd = array();
31        foreach ($tf_idf as $row) {
32              $list_row = array();
33              for ($x = 0; $x < sizeof($row)-2; $x++) {
34                    $d = $row[$x] * $row[sizeof($row)-1];
35                    array_push($list_row, $d);
36              }
37              array_push($wtd, $list_row);
38        }
39        return $wtd;
40  }
```

```
41  //data uji
42  function tf_idf_dt_uji ($list_kata, $hasil_data_uji){
42        $tf_idf = array ();
43        foreach ($list_kata as $kata) {
44              if(strlen($kata)>2){
45                    $cnt_data_uji =
46  substr_count($hasil_data_uji, $kata);
47              }
48              else{
49                    $cnt_data_uji =
50  substr_count($hasil_data_uji, " ".$kata." ");
51              }
52              array_push($tf_idf, $cnt_data_uji);
53        }
54        return $tf_idf;
55  }
56
57  //langkah - 3
58  function wtd_dt_uji ($tf_idf, $tf_idf_dt_uji){
59        $wtd = array();
60        $i=0;
61        foreach ($tf_idf as $row) {
62              $d = round($row[sizeof($row)-1] *
63  $tf_idf_dt_uji[$i], 4);
64              array_push($wtd, $d);
65              $i++;
66        }
67        return $wtd;
68  }
```

Source code* 5.7 Implementasi Tahap Pembobotan (*Term Weighting)

4.10.3 Klasifikasi *Improved K-NN*

Klasifikasi *Improved KNN* dilakukan setelah menghitung pembobotan (*term weighting)*, perhitungan normalisasi dan perhitungan *cosine similiarity* yang kemudian akan dilakukan perankingan hasil dari perhitungan *cosine similiarity*.

4.10.3.1 Perhitungan *Cosine Similarity*

Perhitungan nilai *cosine similarity* dilakukan pada data latih terhadap data uji pada kategori yang akan diketahui. *Cosine similiarity* digunakan untuk mengukur tingkat kemiripan. Pada *Source code* 5.8 ditunjukkan implementasi dari proses *cosine similiarity.*

```
1   function cos_sim($normalisasi, $normalisasi_dt_uji) {
2         $cos_sim = array();
3               for ($x = 0; $x < sizeof($normalisasi); $x++) {
4                     $list_row = array();
5                     $row = $normalisasi[$x];
6                     $row_dt_uji = $normalisasi_dt_uji[$x];
7                     for ($y = 0; $y < sizeof($row); $y++) {
8                           $d = round($row[$y] * $row_dt_uji,
9   4);
10                          array_push($list_row, $d);
11                    }
12                    array_push($cos_sim, $list_row);
13              }
14              return $cos_sim;
15        }
```

```
16
17        function tkt_mirip($cos_sim){
18              $tkt_mirip = array();
19              for ($x = 0; $x < sizeof($cos_sim[0]); $x++) {
20                    $d=0;
21                    for ($y = 0; $y < sizeof($cos_sim); $y++)
22                          $d = $d + $cos_sim[$y][$x];
23                    }
24                    array_push($tkt_mirip, $d);
25              }
26              return $tkt_mirip;
27        }
28        function ranking($tkt_mirip, $dataLatih){
29              $rank = array();
30              $i = 0;
31              foreach ($dataLatih as $data) {
32                    $rank[''.$data['id']] = $tkt_mirip[$i];
33                    $i++;
34              }
35              arsort($rank);
36              // var_dump($rank);
37              return $rank;
38        }
```

Source code* 5.8 Implementasi Tahap Perhitungan *Cosine Similarity

4.10.3.2 Klasifikasi dengan Metode Improved K-NN

Klasifikasi dengan metode *Improved KNN* diawali dengan menetapkan nilai k awal, kemudian akan dicari nilai k baru untuk masing-masing kategori. Setelah mandapat nilai k baru maka akan dilanjutkan dengan menghitung probabilita masing-masing kategori untuk mengeahui hasil klasifikasi. Pada penelitian ini digunakan niali k awal yaitu 2, 4, 6, 8, 10, 15, 20, 25, 30, 40, 45, 50, 75, 100. Pada *source code* 5.9 ditunjukan implementasi kalsifikasi denga metode *Improved KNN*.

```
1     function hitungIKNN($ranking, $dataLatih, $kValue,
2     $kesimpulan){
3                       $cnt_macet = $this->data_latih-
4     >CountKategori('Macet');
5                       $cnt_tdk_macet = $this->data_latih-
6     >CountKategori('Tidak Macet');
7                       $maks = 0;
8                         $k = $kValue; /*Ketetapan sendiri*/
9                         if($cnt_macet >= $cnt_tdk_macet){
10                              $maks = $cnt_macet;
11                        }else{
12                              $maks= $cnt_tdk_macet;
13                        }
14                        $N_macet = round((($k * $cnt_macet) /
15    $maks));
16                        $N_tdk_macet = round((($k *
17    $cnt_tdk_macet) / $maks));
18                        echo "k-Value Macet: ".$N_macet."<br/>";
19                        echo "k-Value Tidak Macet:
20    ".$N_tdk_macet."<br/>";
21                        $i=0;
22                        //hitung probabilitas
23                        $prob_macet = 0;
```

```
                    $pembilang_macet = 0;
                    $penyebut_macet=0;
                    $prob_tdk_macet = 0;
                    $pembilang_tdk_macet = 0;
                    $penyebut_tdk__macet=0;
                    foreach ($ranking as $key=>$val) {
                          // echo "key: ".$key."<br/>";
                          $kategori = $this->data_latih-
>getKategori($key);
                          if($i<$cnt_tdk_macet){
                                if($kategori[0]["kategori"]
=="Tidak Macet"){
                                      // echo "masuk tdk?";
                                      $pembilang_tdk_macet =
$pembilang_tdk_macet + ($val*1);
                                }
                                else{
                                      $pembilang_tdk_macet =
$pembilang_tdk_macet + ($val*0);
                                }
                                $penyebut_tdk__macet=
$penyebut_tdk__macet + $val;
                          }
                          if($i<$cnt_macet){
                                if($kategori[0]["kategori"]
== "Macet"){
                                      $pembilang_macet =
$pembilang_macet + ($val*1);
                                }
                                else{
                                      $pembilang_macet =
$pembilang_macet + ($val*0);
                                }
                                $penyebut_macet=
$penyebut_macet + $val;
                          }
                          $i++;
                    }
                    echo "HASIL I-KNN: <br/>";
                    // echo "pembilang_macet:
".$pembilang_macet."<br/>";
                    // echo "penyebut_macet:
".$penyebut_macet."<br/>";

                    // echo "pembilang_tdk_macet:
".$pembilang_tdk_macet."<br/>";
                    // echo "penyebut_tdk__macet:
".$penyebut_tdk__macet."<br/>";

                    $prob_macet = $pembilang_macet /
$penyebut_macet;
                    $prob_tdk_macet = $pembilang_tdk_macet /
$penyebut_tdk__macet;

                    // echo "prob_macet:
".$prob_macet."<br/>";
                    // echo "prob_tdk_macet:
".$prob_tdk_macet."<br/>";
                    $observ = "";
```

```
83                      if($prob_macet >= $prob_tdk_macet){
84                            $observ = "Macet";
85                            if($kesimpulan == $observ){
86                                  $this->tp= $this->tp + 1;
87                            }else{
88                                  $this->fp = $this->fp + 1;
89                            }
90                            echo "hasil: Masuk Kategori
91   Macet<br/>";
92                      }else{
93                            $observ = "Tidak Macet";
94                            if($kesimpulan == $observ){
95                                  $this->tn = $this->tn + 1;
96                            }else{
97                                  $this->fn = $this->fn + 1;
98                            }
99                            echo "hasil: Masuk Kategori Tidak
10   Macet<br/>";
101                     }
102                     echo "hasil asli: ".$kesimpulan."<br/>";
103               }
104
105               function hitungKNN($ranking, $dataLatih){
106                     $k = 6; /*Ketetapan sendiri*/
107                     $i=0;
108                     //hitung probabilitas
109                     $prob_macet = 0;
110                     $pembilang_macet = 0;
111                     $penyebut_macet=0;
112                     $prob_tdk_macet = 0;
113                     $pembilang_tdk_macet = 0;
114                     $penyebut_tdk__macet=0;
115                     foreach ($ranking as $key=>$val) {
116                           // echo "key: ".$key."<br/>";
117                           $kategori = $this->data_latih-
118   >getKategori($key);
119                           // echo "hasil:
120   ".$kategori[0]["kategori"]."<br/>";
121                           if($i<$k){
122                                 if($kategori[0]["kategori"]
123   =="Tidak Macet"){
124                                       $pembilang_tdk_macet =
125   $pembilang_tdk_macet + ($val*1);
126                                 }
127                                 else{
128                                       $pembilang_tdk_macet =
129   $pembilang_tdk_macet + ($val*0);
130                                 }
131                                 $penyebut_tdk__macet=
132   $penyebut_tdk__macet + $val;
133                           }
134                           if($i<$k){
135                                 if($kategori[0]["kategori"]
136   == "Macet"){
137                                       // echo "masuk ?";
138                                       $pembilang_macet =
139   $pembilang_macet + ($val*1);
140                                 }
141                                 else{
```

```
142                                         $pembilang_macet =
143 $pembilang_macet + ($val*0);
144                                 }
145                                 $penyebut_macet=
146 $penyebut_macet + $val;
147                             }
148                             $i++;
149                     }
150                     echo "HASIL KNN: <br/>";
151                     // echo "pembilang_macet:
152 ".$pembilang_macet."<br/>";
153                     // echo "penyebut_macet:
154 ".$penyebut_macet."<br/>";
155
156                     // echo "pembilang_tdk_macet:
157 ".$pembilang_tdk_macet."<br/>";
158                     // echo "penyebut_tdk__macet:
159 ".$penyebut_tdk__macet."<br/>";
160
161                     $prob_macet = $pembilang_macet /
162 $penyebut_macet;
163                     $prob_tdk_macet = $pembilang_tdk_macet /
164 $penyebut_tdk__macet;
165
166                     echo "prob_macet: ".$prob_macet."<br/>";
167                     echo "prob_tdk_macet:
168 ".$prob_tdk_macet."<br/>";
169                     if($prob_macet >= $prob_tdk_macet){
170                         echo "hasil: Masuk Kategori Macet"
171 ."<br/>" ;
172                     }else{
173                         echo "hasil: Masuk Kategori Tidak
174 Macet" ."<br/>";
175                     }
176             }
```

Source code* 5.9 Implementasi Klasifikasi dengan Metode *Improved K-NN

4.11 Implementasi Antar Muka

Antarmuka (*interface*) di bangun dengan ujuan memudahkan pengguna dalam berinteraksi dengan sistem.

4.11.1 Tampilan Halaman Awal

Pada halaman awal terdapat judul dari sistem dan terdapat dua menu yaitu halaman pengguna dan halaman pengujian. Tampilan halaman pengujian ditunjukkan pada Gambar 5.1.

Gambar 4. 14 Tampilan Halaman Awal

1.2.2 Tampilan Halaman Pengguna

Pada halaman pengguna terdapat kolom untuk memasukkan *tweet* yang akan diuji dan tombol *submit*. Halaman pengguna ini digunakan untuk menguji sebuah *tweet* apakah masuk kategori macet atau tidak macet. Ketika menekan tombol *submit* maka akan menghasilkan *outout* berupa *tweet* asli yang akan diuji, hasil *prepocessing text* pada proses *stemming* dan hasil kategori macet atau tidak macet seperti yang ditampilkan pada Gambar 5.3. Tampilan halaman pengguna ditunjukkan pada Gambar 5.2.

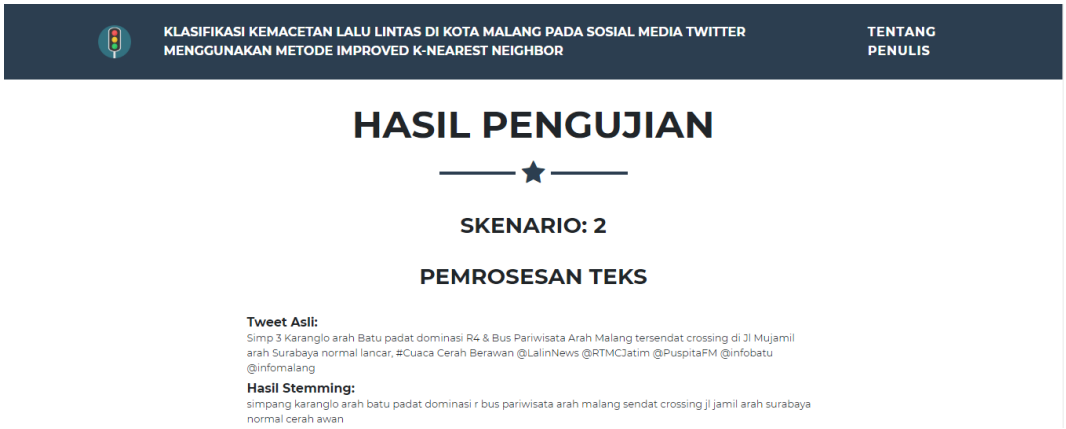
Gambar 4. 15 Tampilan Halaman Pengguna

Gambar 4. 16 Tampilan Hasil Halaman Pengguna

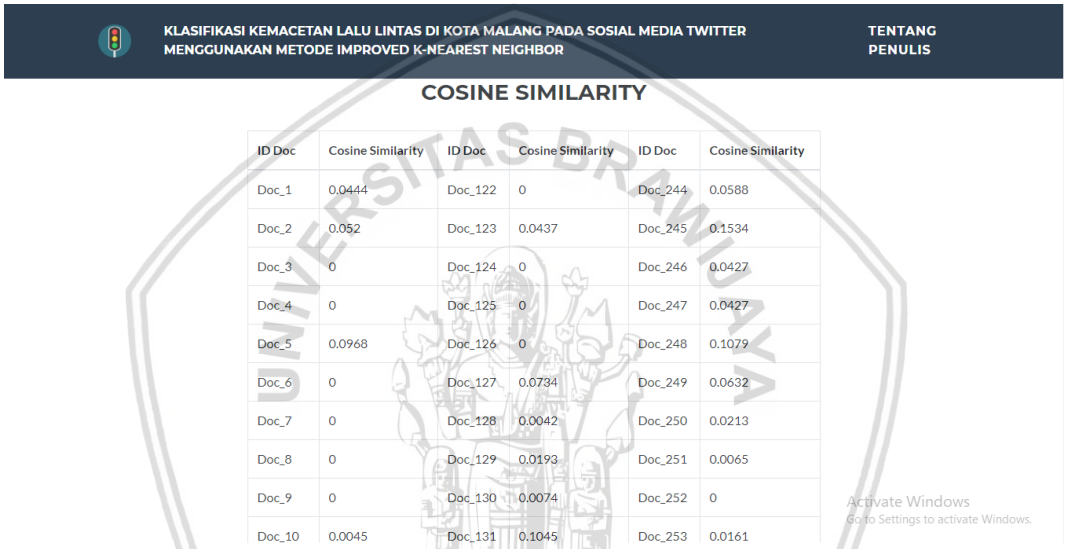
1.2.3 Tampilan Halaman Pengujian

Pada halaman pengujian terdapat kolom untuk memasukkan *tweet* yang akan diuji dan kolom untuk memasukkan skenario serta tombol *submit.* Pada kolom masukkan skenario diisi untuk melakukan pengujian pada skenario ke berapa *tweet* akan diuji. Pada sistem ini terdapat 7 skenario. Hasil pengujian akan keluar setelah menekan tombol *submit.* Hasil dari pengujian ini menampilkan proses *prepocessing text* pada proses *stemming* dan proses hasil perhitungan klasifikasi *improved KNN* yang diawali dari pembobotan, *cosine similiarity* dan *improved KNN*. Tampilan hasil halaman pengujian ditunjukkan pada Gambar 5.5. Pada Gambar 5.4, 5.5, 5.6 dan 5.7 ditunjukkan tampilan halaman pengujian.

Gambar 4. 17 Tampilan Halaman Pengujian

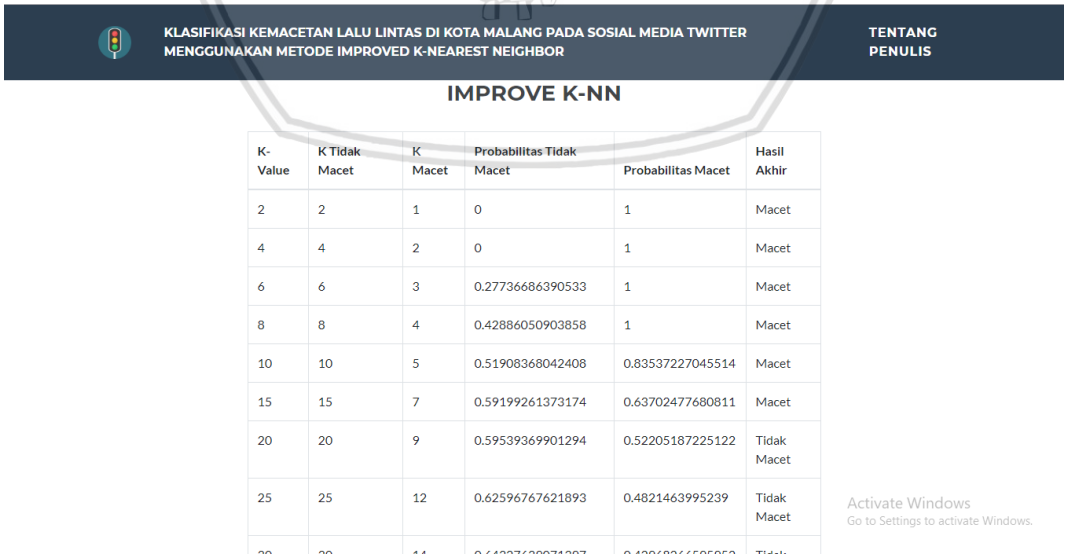


Gambar 4. 18 Tampilan Halaman Hasil Pengujian (1)



ID Doc	Cosine Similarity	ID Doc	Cosine Similarity	ID Doc	Cosine Similarity
Doc_1	0.0444	Doc_122	0	Doc_244	0.0588
Doc_2	0.052	Doc_123	0.0437	Doc_245	0.1534
Doc_3	0	Doc_124	0	Doc_246	0.0427
Doc_4	0	Doc_125	0	Doc_247	0.0427
Doc_5	0.0968	Doc_126	0	Doc_248	0.1079
Doc_6	0	Doc_127	0.0734	Doc_249	0.0632
Doc_7	0	Doc_128	0.0042	Doc_250	0.0213
Doc_8	0	Doc_129	0.0193	Doc_251	0.0065
Doc_9	0	Doc_130	0.0074	Doc_252	0
Doc_10	0.0045	Doc_131	0.1045	Doc_253	0.0161

Gambar 4. 19 Tampilan Halaman Hasil Pengujian (2)



K-Value	K Tidak Macet	K Macet	Probabilitas Tidak Macet	Probabilitas Macet	Hasil Akhir
2	2	1	0	1	Macet
4	4	2	0	1	Macet
6	6	3	0.27736686390533	1	Macet
8	8	4	0.42886050903858	1	Macet
10	10	5	0.51908368042408	0.83537227045514	Macet
15	15	7	0.59199261373174	0.63702477680811	Macet
20	20	9	0.59539369901294	0.52205187225122	Tidak Macet
25	25	12	0.62596767621893	0.4821463995239	Tidak Macet

Gambar 4. 20 Tampilan Halaman Hasil Pengujian (3)

BAB 5 PENGUJIAN DAN ANALISIS

5.1 Pengujian dan Analisis

Pengujian dilakukan untuk mengetahui pengaruh jumlah data latih dan nilai k terhadap *tweet* yang akan diuji. Pengujian dilakukan dengan meguji 150 data uji terhadap 7 skenario yang ada di mana setiap skenario terdapat perbedaan jumlah perbandingan data latih. Pada setiap skenario pengujian dilakukan dengan k *value* awal yaitu, 2, 4, 6, 8, 10, 15, 20, 25, 30, 40, 45, 50, 75 dan 100. Pada Tabel 5.1 ditunjukkan skenario pengujian yang dibangun pada penelitian ini.

Tabel 5. 1 Skenario Pengujian

Skenario	Data Latih			Data Uji		
	M	TM	Jumlah	M	TM	Jumlah
1	100	200	300	70	80	150
2	115	250	365	70	80	150
3	125	350	475	70	80	150
4	150	400	550	70	80	150
5	185	415	600	70	80	150
6	200	200	400	70	80	150
7	200	100	300	70	80	150

Keterangan:

- M : Macet
- TM : Tidak Macet

5.1.1 Skenario 1

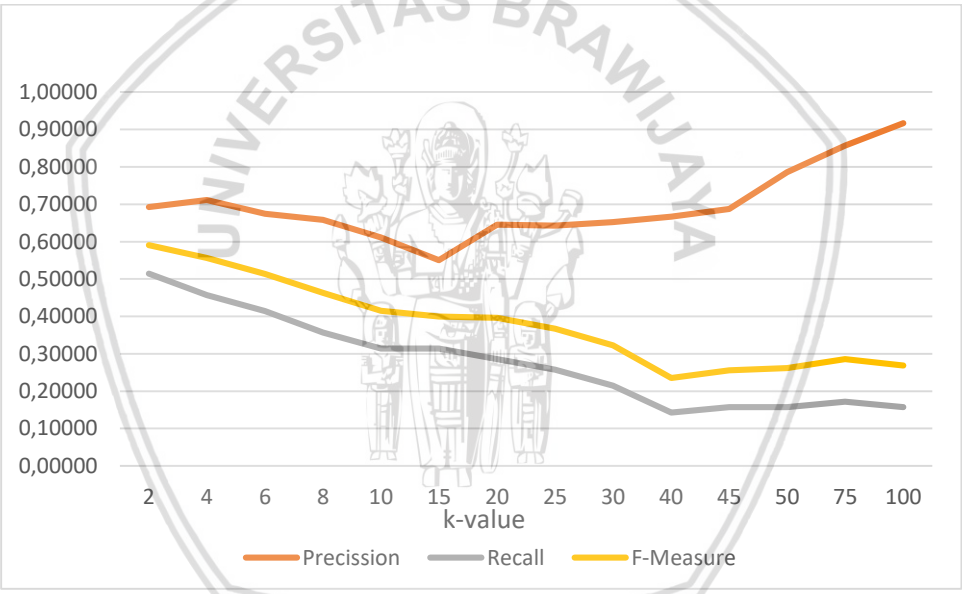
Pada pengujian skenario 1 digunakan data latih sebanyak 300 data latih dengan data tidak macet sebanyak 200 dan data macet sebanyak 100. Data uji yang digunakan sebanyak 150 data uji.

Tabel 5. 2 Pengujian Skenario 1

k-Value awal	k-Value macet	k-Value tidak macet	Precission	Recall	F-Measure	Akurasi
2	1	2	0,69231	0,51429	0,59016	66,67%
4	2	4	0,71111	0,45714	0,55652	66,00%
6	3	6	0,67442	0,41429	0,51327	63,33%
8	4	8	0,65789	0,35714	0,46296	61,33%
10	5	10	0,61111	0,31429	0,41509	58,67%
15	8	15	0,55000	0,31429	0,40000	56,00%

20	10	20	0,64516	0,28571	0,39604	59,33%
25	13	25	0,64286	0,25714	.0,36735	58,67%
30	15	30	0,65217	0,21429	0,32258	58,00%
40	20	40	0,66667	0,14286	0,23529	56,67%
45	23	45	0,68750	0,15714	0,25581	57,33%
50	25	50	0,78571	0,15714	0,26190	58,67%
75	38	75	0,85714	0,17143	0,28571	60,00%
100	50	100	0,91667	0,15714	0,26829	60,00%

Pada Tabel 5.2 menunjukan hasil pengujian dari nilai *precision, recall, f-measure* dan akurasi pada skenario 1. Dari tabel tersebut dapat diketahui bahwa pada saat nilai k-*value* awal bernilai 2 memiliki nilai *f-measure* tertinggi yaitu dengan nilai 0,59016, dan pada saat k-*value* awal bernilai 40 memiliki nilai *f-measure* terendah dengan nilai 0,23529. Gambar 5.3 merupakan grafik dari pengujian skenario 1.



Gambar 5. 1 Grafik Pengujian Skenario 1

5.1.2 Skenario 2

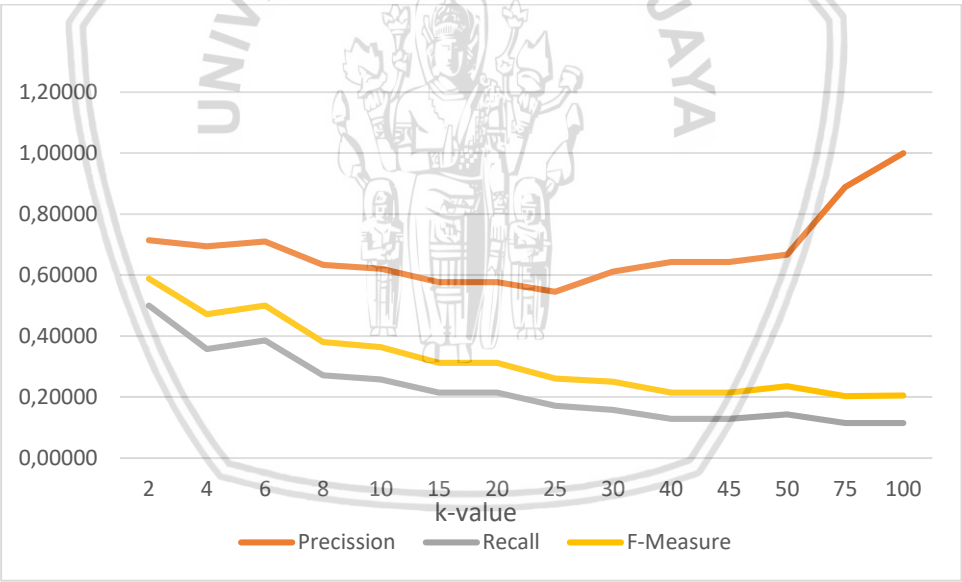
Pada pengujian skenario 2 digunakan data latih sebanyak 365 data latih dengan data tidak macet sebanyak 250 dan data macet sebanyak 115. Data uji yang digunakan sebanyak 150 data uji.

Tabel 5. 3 Pengujian Skenario 2

k-Value			Precission	Recall	F-Measure	Akurasi
Awal	Macet	Tidak Macet				
2	1	2	0,71429	0,50000	0,58824	67,33%
4	2	4	0,69444	0,35714	0,47170	62,67%
6	3	6	0,71053	0,38571	0,50000	64,00%

8	4	8	0,63333	0,27143	0,38000	58,67%
10	5	10	0,62069	0,25714	0,36364	58,00%
15	7	15	0,57692	0,21429	0,31250	56,00%
20	9	20	0,57692	0,21429	0,31250	56,00%
25	12	25	0,54545	0,17143	0,26087	54,67%
30	14	30	0,61111	0,15714	0,25000	56,00%
40	18	40	0,64286	0,12857	0,21429	56,00%
45	21	45	0,64286	0,12857	0,21429	56,00%
50	23	50	0,66667	0,14286	0,23529	56,67%
75	35	75	0,88889	0,11429	0,20253	58,00%
100	46	100	1,00000	0,11429	0,20513	58,67%

Pada Tabel 5.3 menunjukan hasil pengujian dari nilai *precision, recall, f-measure* dan akurasi pada skenario 2. Dari tabel tersebut dapat diketahui bahwa pada saat nilai k-*value* awal bernilai 2 memiliki nilai *f-measure* tertinggi yaitu dengan nilai 0,58824, dan pada saat k-*value* awal bernilai 75 dan 100 memiliki nilai *f-measure* terendah dengan nilai 0,20253. Gambar 5.2 merupakan grafik dari pengujian skenario 2.



Gambar 5. 2 Grafik Pengujian Skenario 2

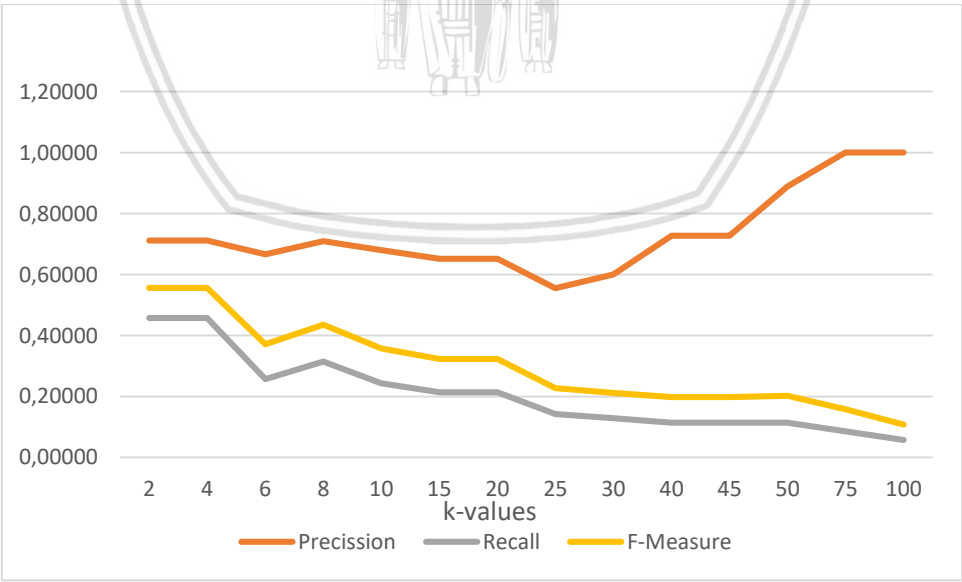
5.1.3 Skenario 3

Pada pengujian skenario 3 digunakan data latih sebanyak 475 data latih dengan data tidak macet sebanyak 350 dan data macet sebanyak 125. Data uji yang digunakan sebanyak 150 data uji.

Tabel 5. 4 Pengujian Skenario 3

k-Value			Precission	Recall	F-Measure	Akurasi
Awal	Macet	Tidak Macet				
2	1	2	0,71111	0,45714	0,55652	66,00%
4	1	4	0,71111	0,45714	0,55652	66,00%
6	2	6	0,66667	0,25714	0,37113	59,33%
8	3	8	0,70968	0,31429	0,43564	62,00%
10	4	10	0,68000	0,24286	0,35789	59,33%
15	5	15	0,65217	0,21429	0,32258	58,00%
20	7	20	0,65217	0,21429	0,32258	58,00%
25	9	25	0,55556	0,14286	0,22727	54,67%
30	11	30	0,60000	0,12857	0,21176	55,33%
40	14	40	0,72727	0,11429	0,19753	56,67%
45	16	45	0,72727	0,11429	0,19753	56,67%
50	18	50	0,88889	0,11429	0,20253	58,00%
75	27	75	1,00000	0,08571	0,15789	57,33%
100	36	100	1,00000	0,05714	0,10811	56,00%

Pada Tabel 5.4 menunjukan hasil pengujian dari nilai *precision, recall, f-measure* dan akurasi pada skenario 3. Dari tabel tersebut dapat diketahui bahwa pada saat nilai k-*value* awal bernilai 2 dan 4 memiliki nilai *f-measure* tertinggi yaitu dengan nilai 0,55652, dan pada saat k-*value* awal bernilai 100 memiliki nilai *f-measure* terendah dengan nilai 0,10811. Gambar 5.3 merupakan grafik dari pengujian skenario 3.



Gambar 5. 3 Gafik Pengujian Skenario 3

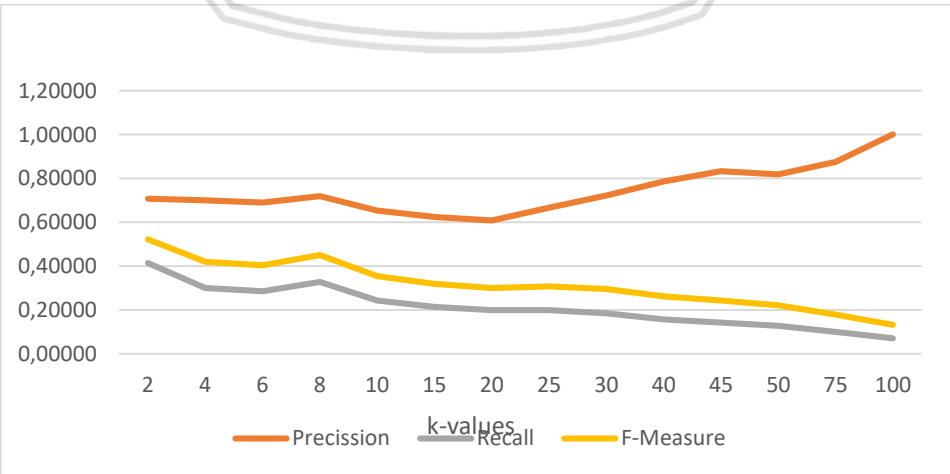
5.1.4 Skenario 4

Pada pengujian skenario 4 digunakan data latih sebanyak 550 data latih dengan data tidak macet sebanyak 400 dan data macet sebanyak 150. Data uji yang digunakan sebanyak 150 data uji.

Tabel 5. 5 Pengujian Skenario 4

k-Value			Precission	Recall	F-Measure	Akurasi
Awal	Macet	Tidak Macet				
2	1	2	0,70732	0,41429	0,52252	64,67%
4	2	4	0,70000	0,30000	0,42000	61,33%
6	2	6	0,68966	0,28571	0,40404	60,67%
8	3	8	0,71875	0,32857	0,45098	62,67%
10	4	10	0,65385	0,24286	0,35417	58,67%
15	6	15	0,62500	0,21429	0,31915	57,33%
20	8	20	0,60870	0,20000	0,30108	56,67%
25	9	25	0,66667	0,20000	0,30769	58,00%
30	11	30	0,72222	0,18571	0,29545	58,67%
40	15	40	0,78571	0,15714	0,26190	58,67%
45	17	45	0,83333	0,14286	0,24390	58,67%
50	19	50	0,81818	0,12857	0,22222	58,00%
75	28	75	0,87500	0,10000	0,17949	57,33%
100	38	100	1,00000	0,07143	0,13333	56,67%

Pada Tabel 5.5 menunjukan hasil pengujian dari nilai *precision, recall, f-measure* dan akurasi pada skenario 4. Dari tabel tersebut dapat diketahui bahwa pada saat nilai k-*value* awal bernilai 2 memiliki nilai *f-measure* tertinggi yaitu dengan nilai 0,52252, dan pada saat k-*value* awal bernilai 100 memiliki nilai *f-measure* terendah dengan nilai 0,13333. Gambar 5.4 merupakan grafik dari pengujian skenario 4.



Gambar 5. 4 Gafik Pengujian Skenario 4

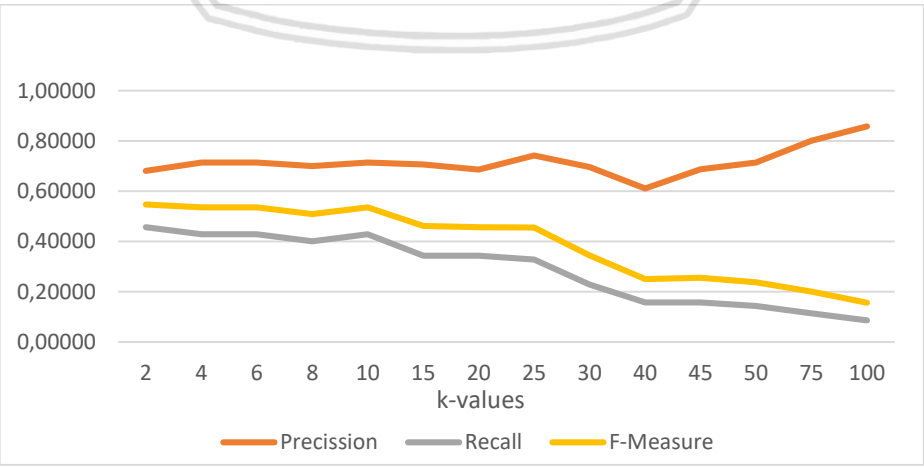
5.1.5 Skenario 5

Pada pengujian skenario 5 digunakan data latih sebanyak 600 data latih dengan data tidak macet sebanyak 400 dan data macet sebanyak 200. Data uji yang digunakan sebanyak 150 data uji.

Tabel 5. 6 Pengujian Skenario 5

k-Value			Precission	Recall	F-Measure	Akurasi
Awal	Macet	Tidak Macet				
2	1	2	0,68085	0,45714	0,54701	64,67%
4	2	4	0,71429	0,42857	0,53571	65,33%
6	3	6	0,71429	0,42857	0,53571	65,33%
8	4	8	0,70000	0,40000	0,50909	64,00%
10	4	10	0,71429	0,42857	0,53571	65,33%
15	7	15	0,70588	0,34286	0,46154	62,67%
20	9	20	0,68571	0,34286	0,45714	62,00%
25	11	25	0,74194	0,32857	0,45545	63,33%
30	13	30	0,69565	0,22857	0,34409	59,33%
40	18	40	0,61111	0,15714	0,25000	56,00%
45	20	45	0,68750	0,15714	0,25581	57,33%
50	22	50	0,71429	0,14286	0,23810	57,33%
75	33	75	0,80000	0,11429	0,20000	57,33%
100	45	100	0,85714	0,08571	0,15584	56,67%

Pada Tabel 5.6 menunjukan hasil pengujian dari nilai *precision, recall, f-measure* dan akurasi pada skenario 5. Dari tabel tersebut dapat diketahui bahwa pada saat nilai k-*value* awal bernilai 2 memiliki nilai *f-measure* tertinggi yaitu dengan nilai 0,54701, dan pada saat k-*value* awal bernilai 100 memiliki nilai *f-measure* terendah dengan nilai 0,15584. Gambar 5.5 merupakan grafik dari pengujian skenario 5.



Gambar 5. 5 Gafik Pengujian Skenario 5

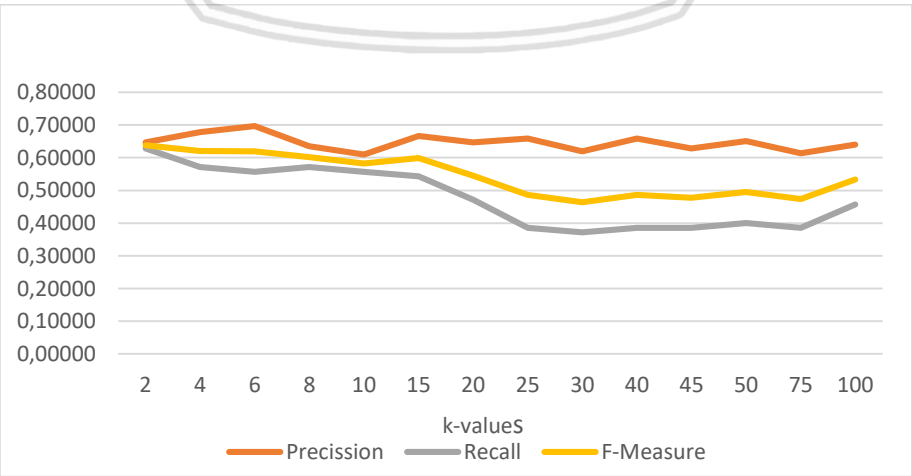
5.1.6 Skenario 6

Pada pengujian skenario 6 digunakan data latih sebanyak 400 data latih dengan data tidak macet sebanyak 200 dan data macet sebanayak 200. Data uji yang digunakan sebanyak 150 data uji.

Tabel 5. 7 Pengujian Skenario 6

k-Value			Precission	Recall	F-Measure	Akurasi
Awal	Macet	Tidak Macet				
2	2	2	0,64706	0,62857	0,63768	66,67%
4	4	4	0,67797	0,57143	0,62016	67,33%
6	6	6	0,69643	0,55714	0,61905	68,00%
8	8	8	0,63492	0,57143	0,60150	64,67%
10	10	10	0,60938	0,55714	0,58209	62,67%
15	15	15	0,66667	0,54286	0,59843	66,00%
20	20	20	0,64706	0,47143	0,54545	63,33%
25	25	25	0,65854	0,38571	0,48649	62,00%
30	30	30	0,61905	0,37143	0,46429	60,00%
40	40	40	0,65854	0,38571	0,48649	62,00%
45	45	45	0,62791	0,38571	0,47788	60,67%
50	50	50	0,65116	0,40000	0,49558	62,00%
75	75	75	0,61364	0,38571	0,47368	60,00%
100	100	100	0,64000	0,45714	0,53333	62,67%

Pada Tabel 5.7 menunjukan hasil pengujian dari nilai *precision, recall, f-measure* dan akurasi pada skenario 6. Dari tabel tersebut dapat diketahui bahwa pada saat nilai k-*value* awal bernilai memiliki nilai 15 *f-measure* tertinggi yaitu dengan nilai 0,63768, dan pada saat k-*value* awal bernilai 30 memiliki nilai *f-measure* terendah dengan nilai 0,46429. Gambar 5.6 merupakan grafik dari pengujian skenario 6.



Gambar 5. 6 Gafik Pengujian Skenario 6

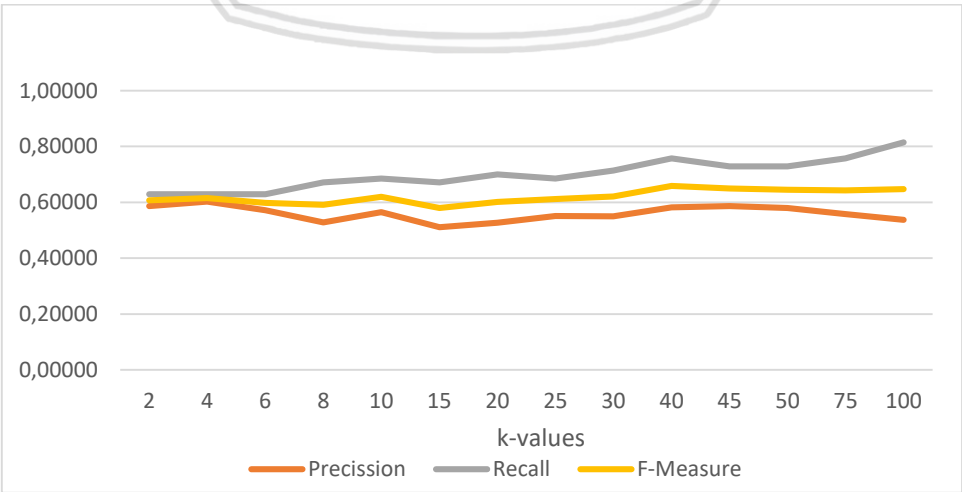
5.1.7 Skenario 7

Pada pengujian skenario 7 digunakan data latih sebanyak 300 data latih dengan data tidak macet sebanyak 100 dan data macet sebanyak 200. Data uji yang digunakan sebanyak 150 data uji.

Tabel 5. 8 Pengujian Skenario 7

k-Value			Precission	Recall	F-Measure	Akurasi
Awal	Macet	Tidak Macet				
2	1	2	0,58667	0,62857	0,60690	62,00%
4	2	4	0,60274	0,62857	0,61538	63,33%
6	3	6	0,57143	0,62857	0,59864	60,67%
8	4	8	0,52809	0,67143	0,59119	56,67%
10	5	10	0,56471	0,68571	0,61935	60,67%
15	8	15	0,51087	0,67143	0,58025	54,67%
20	10	20	0,52688	0,70000	0,60123	56,67%
25	13	25	0,55172	0,68571	0,61146	59,33%
30	15	30	0,54945	0,71429	0,62112	59,33%
40	20	40	0,58242	0,75714	0,65839	63,33%
45	23	45	0,58621	0,72857	0,64968	63,33%
50	25	50	0,57955	0,72857	0,64557	62,67%
75	38	75	0,55789	0,75714	0,64242	60,67%
100	50	100	0,53774	0,81429	0,64773	58,67%

Pada Tabel 5.8 menunjukan hasil pengujian dari nilai *precision, recall, f-measure* dan akurasi pada skenario 7. Dari tabel tersebut dapat diketahui bahwa pada saat nilai k-*value* awal bernilai 40 memiliki nilai *f-measure* tertinggi yaitu dengan nilai 0,65839, dan pada saat k-*value* awal bernilai 15 memiliki nilai *f-measure* terendah dengan nilai 0,58025. Gambar 5.7 merupakan grafik dari pengujian skenario 7.



Gambar 5. 7 Gafik Pengujian Skenario 7

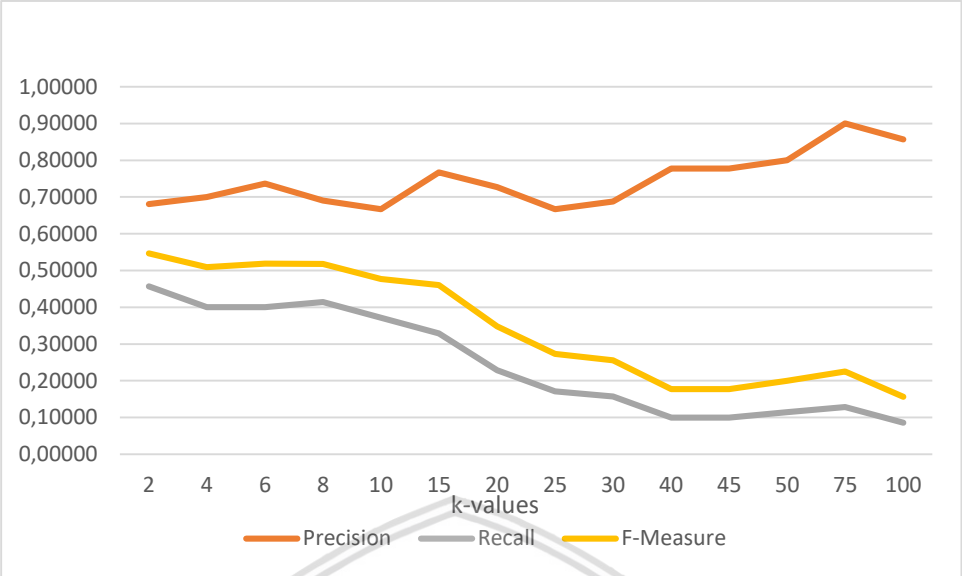
5.1.8 Perbandingan Hasil *K-Nearest Neighbor*

Berdasarkan pengujian yang telah dilakukan didapatkan akurasi terbaik pada skenario 5 yaitu dengan data latih sebanyak 600 dengan 185 data macet dan 415 data tidak macet. Oleh karena itu, skeanrio 5 akan digunakan sebagai pembanding hasil pengujian dengan metode *k-nearest neighbor*. Pada tabel 5.9 ditunjukkan nilai *precision, recall, f-measure* dan akurasi pada skenario 5 dengan menggunakan metode *k-nearest neighbor*.

Tabel 5. 9 *Precision, Recall, F-Measure* dan Akurasi Pengujian Metode *K-Nearest Neighbor*

K-Value	Precision	Recall	F-Measure	Akurasi
2	0,68085	0,45714	0,54701	64,67%
4	0,70000	0,40000	0,50909	64,00%
6	0,73684	0,40000	0,51852	65,33%
8	0,69048	0,41429	0,51786	64,00%
10	0,66667	0,37143	0,47706	62,00%
15	0,76667	0,32857	0,46000	64,00%
20	0,72727	0,22857	0,34783	60,00%
25	0,66667	0,17143	0,27273	57,33%
30	0,68750	0,15714	0,25581	57,33%
40	0,77778	0,10000	0,17722	56,67%
45	0,77778	0,10000	0,17722	56,67%
50	0,80000	0,11429	0,20000	57,33%
75	0,90000	0,12857	0,22500	58,67%
100	0,85714	0,08571	0,15584	56,67%

Berdasarkan pengujian yang dilakukan pada skenario 5 apabila dibandingkan maka dapat disimpulkan bahwa metode *improved k-nearest neighbor* memiliki hasil yang lebih baik dari metode *k-nearest neighbor*. Pada Gambar 5.8 merupakan grafik dari nilai *precision, recall* dan *f-measure* skenario 5 dengan menggunakan metode *k-nearest neighbor*. Dari Gambar 5.8 menunjukkan grafik *precision, recall, f-measure* dan Akurasi Pengujian Metode K-Nearest Neighbor. Garis merah mewakili nilai dari *precision,* garis ungu mewakili nilai dari *f-measure* dan garis hijau mewakili nilai dari *recall*. Dari garis ungu yang mewakili *f-measure* diketahui bahwa pada saat *k-values* bernilai rendah memiliki nilai *f-measure* tinggi, sedangkan saat *k-values* bernilai tinggi memiliki nilai *f-measure* rendah. Begitu juga dengan garis hijau yang mewakili *recall* diketahui bahwa pada saat *k-values* bernilai rendah memiliki nilai *recall* tinggi, sedangkan saat *k-values* bernilai tinggi memiliki nilai *recall* rendah. Dari garis merah yang mewakili nilai *precision* diketahui bahwa pada saat *k-values* bernilai rendah memiliki nilai *precision* rendah, sedangkan saat *k-values* bernilai tinggi cenderung memiliki nilai *precision* tinggi.



Gambar 5. 8 Grafik *Precision, Recall, F-Measure* dan Akurasi Pengujian Metode *K-Nearest Neighbor*

Pada Tabel 5.10, disajikan perbandingan hasil pengujian dari skenario terbaik metode *improved k-nearest neighbor* yaitu skenario 2 dengan metode *k-nearest neighbor*.

Tabel 5. 10 Perbandingan Hasil Pengujian Metode *Improved K-Nearest Neighbor* Skenario 2 Dan Metode K-Nearest Neighbor

k-values	Improved K-NN				K-NN			
	Precision	***Recall***	***F-Measure***	**Akurasi**	***Precision***	***Recall***	***F-Measure***	**Akurasi**
2	0,68085	0,45714	0,54701	64,67%	0,68085	0,45714	0,54701	64,67%
4	0,71429	0,42857	0,53571	65,33%	0,70000	0,40000	0,50909	64,00%
6	0,71429	0,42857	0,53571	65,33%	0,73684	0,40000	0,51852	65,33%
8	0,70000	0,40000	0,50909	64,00%	0,69048	0,41429	0,51786	64,00%
10	0,71429	0,42857	0,53571	65,33%	0,66667	0,37143	0,47706	62,00%
15	0,70588	0,34286	0,46154	62,67%	0,76667	0,32857	0,46000	64,00%
20	0,68571	0,34286	0,45714	62,00%	0,72727	0,22857	0,34783	60,00%
25	0,74194	0,32857	0,45545	63,33%	0,66667	0,17143	0,27273	57,33%
30	0,69565	0,22857	0,34409	59,33%	0,68750	0,15714	0,25581	57,33%
40	0,61111	0,15714	0,25000	56,00%	0,77778	0,10000	0,17722	56,67%
45	0,68750	0,15714	0,25581	57,33%	0,77778	0,10000	0,17722	56,67%
50	0,71429	0,14286	0,23810	57,33%	0,80000	0,11429	0,20000	57,33%
75	0,80000	0,11429	0,20000	57,33%	0,90000	0,12857	0,22500	58,67%
100	0,85714	0,08571	0,15584	56,67%	0,85714	0,08571	0,15584	56,67%

Rata -rata	0,71592	0,28878	0,39151	61,19%	0,74540	0,24694	0,34580	60,33%

Tabel 5.10 merupakan tabel perbandingan hasil pengujian yang menunjukkan perbandingan dari nilai *precision, recall, f-measure* dan akurasi pada skenario 2. Dari pengujian terebut dapat diketahui nilai pada metode *improved k-nearest neighbor* nilai rata-rata *f-measure* memiliki nilai lebih baik yaitu 0,39151 dibandingkan dengan nilai *f=measure* metode *k-nearest neighbor* yaitu 0,34580. Begitu juga dengan nilai akurasi, metode *improved k-nearest neighbor* memiliki nilai lebihbaik yaitu 61,19% dibandingkan dengan nilai akurasi *k-nearest neighbor* yaitu 60,33%. Hal tersebut dpaat menunjukkan bahwa metode *improvd k-nearest neighbor* memiliki hasil yang lebih baik dari metode *k-nearest neighbor.*

5.2 Analisis

Dari hasil pengujian yang telah dilakukan pada setiap skenario pengujian yang terdiri dari 7 skenario ini, dapat diketahui bahwa dengan menggunakan metode *Improved K-Nearest Neighbor* hasilnya tidak jauh berbeda dengan menggunakan metode *K-Nearest Neighbor*. Namun, metode *Improved K-Nearest Neighbor* memiliki rata-rata nilai akurasi lebih tinggi. Hal tersebut terbukti pada Tabel 5.10 dimana hasil akurasi *Improved K-Nearest Neighbor* memiliki nilai akurasi 61,19% sedangkan metode *K-Nearest Neighbor* memiliki nilai akurasi 60,33%. Untuk hasil akurasi tertinggi dari semua skenario terdapat pada *k-value* yang kecil yaitu 2, 4 dan 6.

Pada skenario 1, 2, 3, 4 dan 5 memiliki nilai *f-measure* tertinggi pada nilai k-awal bernilai 2. Skenario ini memiliki jumlah perbandingan data tidak macet lebih besar daripada data macet. Pada skenario 6 nilai *f-measure* tertinggi ada pada nilai k-awal bernilai 15. Skenario 6 memiliki jumlah perbandingan data tidak macet dan data macet berimbang. Pada skenario 7 memiliki nilai *f-measure* tertinggi ada pada nilai k-awal bernilai 40. Skenario 7 memiliki jumlah perbandingan data macet lebih besar daripada data macet. Maka, dapat disimpulkan bahwa perbandingan jumlah data yang digunakan mempengaruhi nilai *f-measure* pada nilai k-awal. Hal ini dapat dilihat pada skenario 1, 2, 3, 4 dan 5 yang memiliki jumlah perbandingan data tidak macet lebih besar daripada data macet memiliki nilai *f-measure* tertinggi pada nilai k-awal kecil yaitu 2, sedangkan pada skenario 6 dan 7 memiliki jumlah perbandingan data tidak macet dan data macet berimbang dan memiliki jumlah perbandingan data macet lebih besar daripada data macet memiliki nilai *f-measure* tertinggi pada nilai k-awal bernilai 15 dan 40. Oleh karena itu, dalam menentukan data latih harus dilakukan dengan teliti karena besar kecilnya data latih dapat mempengaruhi hasil pengujian, penentuan data latih yang tepat dapat menghasilkan tingkat akurasi yang lebih maksimal.

Pada data latih dan data uji terdapat *tweet* yang mengandung keterangan waktu berupa jam. Pada proses *prepocessing* keterangan berupa jam tersebut

tidak dihilangkan karena pada *tweet* tersebut hanya untuk menunjukkan pola kapan terjadinya macet atau tidak macet. Hal tersebut tidak mempengaruhi hasil dari klasifikasi karena pada dasarnya *tweet* tersebut sama saja dengan *tweet* yang tidak mengandung keterangan waktu berupa jam.

Pada penelitian ini memiliki hasil akurasi dan selisih dengan metode KNN terbilang rendah, dimana hasil akurasi tertinggi hanya bernilai 61,19%. Hal ini disebabkan karena adanya *term* atau kata yang muncul pada kedua kategori. Misalnya, *term* yang menunjukkan nama jalan muncul di kategori macet dan tidak macet. Hasil klasifikasi dari metode *Improved K-Nearest Neighbor* dan KNN memuliki selisih rata-rata yang terbilang rendah hanya bernilai 0,86%. Hal ini disebabkan karena adanya nilai *k-value* baru yang sama dengan nilai *k-value* awal yang ditetapkan. Dan nilai *k-value* baru tersebut terdapat pada ketegori dengan jumlah data yang lebih banyak.

BAB 6 KESIMPULAN

6.1 Kesimpulan

Dari penelitian Klasifikasi Kemacetan Lalu Lintas Di Kota Malang Pada Sosial Media Twitter Menggunakan Metode *Improved K-Nearest Neighbor* ini, beikut merupakan kesimpulan yang didapat:

1. Metode *Improved K-Nearest Neighbor* dapat digunakan dalam poses klasifikasi kemacetan lalu lintas berdasarkan pada data yang berupa *tweet*. Untuk mendapatkan hasil klasifikasi tersebut tedapat beberapa proses yang dilakukan yaitu *prepocessing text,* pembobotan (*term weighting),* normalisasi, *cosine similiarity*, mengurutkan tingkat kemiripan dan menentukan *k-value* baru.
2. Hasil pengujian dari penelitian Klasifikasi Kemacetan Lalu Lintas Di Kota Malang Pada Sosial Media Twitter Menggunakan Metode *Improved K-Nearest Neighbor*, didapatkan hasil terbaik dengan nilai akurasi 65.33%, nilai *f-measure* 0.53714, nilai *recall* 0.428571 dan nilai *precision* 0.714286. Dari hasil yang didapatkan, dapat diketahui bahwa perbandingan jumlah data latih, banyaknya data yang digunakan dan *k-values* dapat mempengaruhi hasil klasifikasi kemacetan lalu lintas berdasarkan *tweet* dari sosial media Twitter.

6.2 Saran

Dari penelitian Klasifikasi Kemacetan Lalu Lintas Di Kota Malang Pada Sosial Media Twitter Menggunakan Metode *Improved K-Nearest Neighbor* ini, beikut merupakan beberapa saran yang didapatkan dengan tujuan dapat dikembangkan lebih lanjut adalah sebagai berikut:

1. Proses pelabelan macet dan tidak macet dalam penelitian ini dilakukan dengan memberikan kuesioner kepada beberapa mahasiswa sehingga hasil yang didapatkan masih kurang akurat. Oleh karena itu, untuk mendapatkan hasil yang akurat dibutuhkan pakar yang memiliki pemahaman lebih dalam memahami karakteristik konten *tweet* dari pengguna twitter.
2. Dalam penelitian ini terdapat beberapa data *tweet* yang menggunakan bahasa kurang baku, penyingkatan kata dan terdapat *tweet* yang sama dalam waktu berbeda, maka diperlukan proses normalisasi kata untuk memberikan hasil yang lebih optimal.
3. Sistem dibangun dengan memanfaatkan metode Improved K-NN yang kurang mampu menangani jumlah data latih yang kurang seimbang secara tepat. Pengembangan sistem dengan menggunakan metode yang lain atau menggunakan metode Improved K-NN yang digabungkan dengan metode lain akan mampu memberikan hasil pengklasifikasin yang lebih optimal.

DAFTAR PUSTAKA

B, DwijaWisnu., Hetami, A., 2015. *PERANCANGAN INFORMATION RETRIEVAL (IR) UNTUK PENCARIAN IDE POKOK TEKS ARTIKEL BERBAHASA INGGRIS DENGAN PEMBOBOTAN VECTOR SPACE MODEL*. Jurnal Ilmiah Teknologi dan Informasi ASIA Vol.9, No. 1, Februari 2015.

Baoli, L., Shiwen, Y. dan Qin, L., 2003. *An Improved k-Nearest Neighbor Algorithm for Text Categorization*. [online] Tersedia di: <https://pdfs.semanticscholar.org/490a/b325ba480f6fb71cdbb5f87ff4cb70918686.pdf> [Diakses 25 Januari 2018]

BeritaSatu, 2017. *Indonesia Masuk Lima Besar Pengguna Twitter*. [online] Tersedia di: <http://www.beritasatu.com/digital-life/428591-indonesia-masuk-lima-besar-pengguna-twitter.html> [Diakses 25 Februari 2018]

Claudy, Y.I., Perdana, R.S., Fauzi, M.A., 2018. *Klasifikasi Dokumen Twitter Untuk Mengetahui Karakter Calon Karyawan Menggunakan Algoritme K-Nearest Neighbor (KNN)*. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Vol. 2, No. 8, Agustus 2018, hlm. 2761-2765.

Fauzi, M.A., Arifin, A.Z., Yuniarti, A., 2014. *Term Weighting Berbasis Indeks Buku dan Kelas untuk Perangkingan Dokumen Berbahasa Arab*. LONTAR KOMPUTER VOL. 5, NO. 2, AGUSTUS 2014.

Herdiawan, 2015. *Analisis Sentimen Terhadap TELKOM Indihome berdasarkan Opini Publik menggunakan Metode Improved K-Nearest Neighbor*. Jurnal Ilmiah Komputer dan Informatika (KOMPUTA). [online] Tersedia di: <http://elib.unikom.ac.id/download.php?id=303675> [Diakses 25 Januari 2018]

Kominfo, 2017. *Kominfo : Pengguna Internet di Indonesia 63 Juta Orang.* [online] Tersedia di: <https://www.kominfo.go.id/content/detail/3415/kominfo-pengguna-internet-di-indonesia-63-juta-orang/0/berita_satker> [Diakses 24 Januari 2018]

Kompas, 2016. *Mengintip Sejarah Pendirian Twitter yang Penuh Intrik*. [online] Tersedia di: < https://tekno.kompas.com/read/2016/03/21/18021707/Mengintip.Sejarah.Pendirian.Twitter.yang.Penuh.Intrik> [Diakses 25 Februari 2018]

Kompas, 2017. *Twitter 280 Karakter Resmi di Seluruh Dunia*. [online] Tersedia di: < https://tekno.kompas.com/read/2017/11/08/08340057/twitter-280-karakter-resmi-di-seluruh-dunia> [Diakses 25 Februari 2018]

Kompas, 2018. *Ini 10 Kota Termacet di Indonesia*. [online] Tersedia di: <https://properti.kompas.com/read/2018/02/25/182046621/ini-10-kota-termacet-di-indonesia.> [Diakses 25 Februari 2018]

Megantara, G. Kurniati, A.P., Suryani, A.A., 2010. *KLASIFIKASI TEKS DENGAN MENGGUNAKAN IMPROVED K-NEAREST NEIGHBOR ALGORITHM*. Teknik Informatika, Fakultas Informatika, Universitas Telkom, Bandung.

Nathania, D.Z., Indriarti., Bachtiar, F.A., 2018. *Klasifikasi Spam Pada Twitter Menggunakan Metode Improved KNN.* Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Vol. 2, No. 10, Oktober 2018, hlm. 3948-3956.

Nurjanah, W.W., Perdana, R.S., Fauzi, M.A., 2017. *Analisis Sentimen Terhadap Tayangan Televisi Berdasarkan Opini Masyarakat pada Media Sosial Twitter menggunakan Metode K-Nearest Neighbor dan Pembobotan Jumlah Retweet*. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Vol. 1, No. 12, Desember 2017, hlm. 1750-1757.

Puspitasari, A.A., Santoso Edy., Indriati., 2018. *Klasifikasi Dokumen Tumbuhan Obat Menggunakan Metode Improved k-Nearest Neighbor.* Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Vol. 2, No. 2, Februari 2018, hlm. 486-492.

Putri, P.A., Ridok Achmad., Indriati., 2017. *IMPLEMENTASI METODE IMPROVED K-NEAREST NEIGHBOR PADA ANALISIS SENTIMEN TWITTER BERBAHASA INDONESIA.* [online] Tersedia di: < https://www.academia.edu/11472083/IMPLEMENTASI_METODE_IMPROVED_K-NEAREST_NEIGHBOR_PADA_ANALISIS_SENTIMEN_TWITTER_BERBAHASA_INDONESIA?auto=download> [Diakses 25 Januari 2018]

Ridok, Ahmad., Latifah, Ritnani., 2015. *Klasifikasi Teks Bahasa Indonesia Pada Corpus Tak Seimbang Menggunakan NWKNN*. Konferensi Nasional Sistem & Informatika 2015.

Rodiansyah, S.F., Winarko ,E., 2013. *Klasifikasi Posting Twitter Kemacetan Lalu Lintas Kota Bandung Menggunakan Naive Bayesian Classification*. IJCSS Vol. 7, No.1, Januari 2013, hlm 13-22.

Sremanthy, J. Balamurugan, P.S., 2012. *AN EFFICIENT TEXT CLASSIFICATION USING KNN AND NAIVE BAYESIAN*. International Journal on Computer Science and Engineering (IJCSE). Coimbatore, India.

Surya Malang, 2017. *Ternyata, Pemicu Kemacetan di Kota Malang Tak Hanya Meningkatnya Kendaraan, Ini Penyebab Lainnya*. [online] Tersedia di: <http://suryamalang.tribunnews.com/2017/03/31/ternyata-pemicu-kemacetan-di-kota-malang-tak-hanya-meningkatnya-kendaraan-ini-penyebab-lainnya?page=all> [Diakses 23 Januari 2018]