



**ANALISIS SENTIMEN KEBIJAKAN PEMINDAHAN IBUKOTA
REPUBLIK INDONESIA DENGAN MENGGUNAKAN
ALGORITME TERM-BASED RANDOM SAMPLING DAN
METODE KLASIFIKASI NAÏVE BAYES**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Akhmad Sa'rony
NIM: 165150200111119



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2020



PENGESAHAN

Analisis Sentimen Kebijakan Pemindahan Ibukota Republik Indonesia dengan Menggunakan Algoritme Term-Based Random Sampling dan Metode Klasifikasi

Naïve Bayes

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Akhmad Sa'rony

NIM: 165150200111119

Skripsi ini telah diuji dan dinyatakan lulus pada

7 Januari 2020

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing 2

Putra Pandu Adikara, S.Kom., M.Kom.

NIP: 19850725 200812 1 002

Randy Cahya Wihandika, S.ST., M.Kom.

NIK: 201405 880206 1 001

Mengetahui

Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T., M.T., Ph.D.

NIP: 19710518 200312 1 001



Universitas Brawijaya
Universitas Brawijaya

Universitas Brawijaya
Universitas Brawijaya
Universitas Brawijaya
Universitas Brawijaya
Universitas Brawijaya
Universitas Brawijaya
Universitas Brawijaya
Universitas Brawijaya
Universitas Brawijaya

Universitas Brawijaya
Universitas Brawijaya
Universitas Brawijaya
Universitas Brawijaya
Universitas Brawijaya
Universitas Brawijaya
Universitas Brawijaya
Universitas Brawijaya
Universitas Brawijaya

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 referensi.

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, 10 Januari 2020



Akhmad Sa'rony

NIM: 165150200111119



PRAKATA

Puji Syukur kami panjatkan atas kehadirat Tuhan Yang Maha Esa yang telah memberikan anugrah, rahmat serta hidayah-Nya kepada penulis, sehingga skripsi ini dapat terselesaikan dengan baik. Skripsi ini disusun sebagai salah satu syarat untuk memenuhi persyaratan untuk memperoleh gelar Sarjana Komputer pada program studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang. Pada kesempatan ini, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu, baik secara langsung maupun tidak langsung dalam pembuatan skripsi ini dari awal hingga selesai. Ucapan terima kasih penulis sampaikan kepada:

1. Bapak Putra Pandu Adikara, S.Kom., M.Kom dan Bapak Randy Cahya Wihandika, S.ST., M.Kom. selaku dosen Pembimbing I dan dosen Pembimbing II yang telah sabar membimbing dan mengarahkan penulis dalam menyelesaikan skripsi ini.
2. Bapak Agus Wahyu Widodo, S.T, M.Cs selaku ketua Program Studi Teknik Informatika.
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D. selaku Ketua Jurusan Teknik Informatika Universitas Brawijaya Malang.
4. Orang Tua dan seluruh keluarga yang selama ini memberi semangat, perhatian, dan doa hingga skripsi ini terselesaikan.
5. Semua pihak dan teman-teman lainnya yang tidak dapat penulis sebutkan satu persatu.

Penulis juga menyadari sepenuhnya bahwa masih banyak terdapat kekurangan dalam penulisan tugas akhir ini, untuk itu diharapkan adanya kritik dan saran yang bersifat membangun demi meningkatkan wawasan serta besar harapan semoga laporan ini dapat bermanfaat bagi semua pihak.

Malang, 10 Januari 2020

Akhmad Sa'rony
aron97729@gmail.com



ABSTRAK

Akhmad Sa'rony, Analisis Sentimen Kebijakan Pemindahan Ibukota Republik Indonesia dengan Menggunakan Algoritme Term-Based Random Sampling dan Metode Klasifikasi Naïve Bayes

Pembimbing: Putra Pandu Adikara, S.Kom., M.Kom. dan Randy Cahya Wihandika, S.ST., M.Kom.

Kebijakan pemindahan Ibukota Republik Indonesia yang telah dicanangkan oleh Presiden Joko Widodo pada bulan Agustus lalu banyak menimbulkan pro dan kontra di masyarakat, terutama dalam lingkungan media sosial. Dalam penelitian ini dilakukan analisis sentimen terhadap kebijakan tersebut dengan menggunakan data yang didapatkan dari media sosial Twitter. Proses pengembangan sistem meliputi *data scraping*, *preprocessing*, perhitungan *raw term frequency* dan klasifikasi dengan menggunakan algoritme Naïve Bayes. Dalam *preprocessing*, proses *filtering* dilakukan dengan menggunakan metode *term-based random sampling* untuk membuat *stoplist*. Pengujian dilakukan dengan 2 metode yaitu pengujian parameter dan *confusion matrix* berjenis *multiclass*. Pengujian parameter dilakukan dengan mengubah persentase *term* data latih yang digunakan sebagai *stoplist* yaitu dari rentang 0 persen hingga 60 persen, sedangkan *confusion matrix* digunakan untuk menghitung nilai akurasi, *precision*, *recall*, dan *f-measure*. Berdasarkan *confusion matrix* hasil pengujian, sistem mendapatkan nilai *macroaverage* terbaik pada klasifikasi dengan *stoplist* 20 persen dengan nilai *macroaverage akurasi* sebesar 0,94, *macroaverage precision* sebesar 0,945, *macroaverage recall* sebesar 0,94, dan *macroaverage f-measure* sebesar 0,938.

Kata kunci: *analisis sentimen*, *term-based random sampling*, *stoplist*, *naïve bayes*, *pemindahan Ibukota*



ABSTRACT

Akhmad Sa'rony, Analysis of Sentiment Policy on Transferring the Capital City of the Republic of Indonesia with the Term-Based Random Sampling Algorithm and Naïve Bayes Classification Method

Supervisors: Putra Pandu Adikara, S.Kom., M.Kom. and Randy Cahya Wihandika, S.ST., M.Kom.

The capital city relocation policy of the Republic of Indonesia that was announced by President Joko Widodo last August caused many pros and cons in the community, especially in the social media environment. In this study, sentiment analysis of the policy is done using data obtained from Twitter. The system development process includes data scraping, preprocessing, Raw term frequency calculation and classification using the Naïve Bayes algorithm. In preprocessing, the filtering process is done using the term-based random sampling method to create a stoplist. The testing process is done by 2 methods, parameter testing and multiclass confusion matrix testing. Parameter testing is done by changing the percentage of term of the training data used as a stoplist, ranging from 0 percent to 60 percent, while the confusion matrix is used to calculate the value of accuracy, precision, recall, and f-measure. Based on the confusion matrix test results, the system gets the best macroaverage value in the classification with a stoplist of 20 percent with an accuracy macroaverage value of 0.94, precision macroaverage value of 0.945, recall macroaverage value of 0.94, and f-measure macroaverage value of 0.938.

Keywords: sentiment analysis, term-based random sampling, stoplist, naïve bayes, capital city relocation



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Pembahasan.....	4
BAB 2 TINJAUAN PUSTAKA.....	6
2.1 Kajian Pustaka	6
2.2 Landasan Teori.....	9
2.2.1 <i>Text Mining</i>	9
2.2.2 Analisis Sentimen	11
2.2.3 Media Sosial	11
2.2.4 Twitter	11
2.2.5 Pembobotan <i>Raw Term Frequency</i> (RTF).....	12
2.2.6 Metode Klasifikasi Naïve Bayes.....	13
2.2.7 <i>Term-Based Random Sampling</i>	14
2.2.8 Evaluasi.....	15
BAB 3 METODOLOGI PENELITIAN	17
3.1 Tipe Penelitian	17
3.2 Strategi Penelitian.....	17
3.3 Peralatan Pendukung.....	17

Universitas Brawijaya	Universitas Brawijaya	Universitas Brawijaya	Universitas Brawijaya	Universitas Brawijaya
awijaya	Universitas Brawijaya	Universitas Brawijaya	Universitas Brawijaya	Universitas Brawijaya
awijaya	Universitas Brawijaya	Universitas Brawijaya	Universitas Brawijaya	Universitas Brawijaya
awijaya	Universitas Brawijaya	Universitas Brawijaya	Universitas Brawijaya	Universitas Brawijaya
awijaya	Universitas Brawijaya	Universitas Brawijaya	Universitas Brawijaya	Universitas Brawijaya
awijaya	Universitas Brawijaya	Universitas Brawijaya	Universitas Brawijaya	Universitas Brawijaya
awijaya	Universitas Brawijaya	Universitas Brawijaya	Universitas Brawijaya	Universitas Brawijaya
awijaya	3.3.1 Hardware	17	Universitas Brawijaya	Universitas Brawijaya
awijaya	3.3.2 Software	17	Universitas Brawijaya	Universitas Brawijaya
awijaya	3.4 Lokasi Penelitian	18	Universitas Brawijaya	Universitas Brawijaya
awijaya	3.5 Teknik Pengumpulan Data	18	Universitas Brawijaya	Universitas Brawijaya
awijaya	3.6 Data Penelitian	18	Universitas Brawijaya	Universitas Brawijaya
awijaya	3.7 Teknik Analisis Data	18	Universitas Brawijaya	Universitas Brawijaya
awijaya	3.8 Perancangan Algoritme	19	Universitas Brawijaya	Universitas Brawijaya
awijaya	3.9 Teknik Penerapan Metode	20	Universitas Brawijaya	Universitas Brawijaya
BAB 4 PERANCANGAN	21	Universitas Brawijaya	Universitas Brawijaya	
4.1 Diagram Alir Sistem (<i>Flowchart</i>)	21	Universitas Brawijaya	Universitas Brawijaya	
4.1.1 Diagram Alir <i>Preprocessing</i>	21	Universitas Brawijaya	Universitas Brawijaya	
4.1.2 <i>Term-Based Random Sampling</i>	28	Universitas Brawijaya	Universitas Brawijaya	
4.1.3 Diagram Alir Perhitungan <i>Raw Term Frequency</i>	35	Universitas Brawijaya	Universitas Brawijaya	
4.1.4 Diagram Alir Klasifikasi Naïve Bayes	37	Universitas Brawijaya	Universitas Brawijaya	
4.2 Perhitungan Manual	39	Universitas Brawijaya	Universitas Brawijaya	
4.2.1 <i>Data Scraping</i>	40	Universitas Brawijaya	Universitas Brawijaya	
4.2.2 <i>Preprocessing</i>	40	Universitas Brawijaya	Universitas Brawijaya	
4.2.3 Perhitungan <i>Raw term frequency (RTF)</i>	46	Universitas Brawijaya	Universitas Brawijaya	
4.2.4 <i>Term-Based Random Sampling (TBRs)</i>	48	Universitas Brawijaya	Universitas Brawijaya	
4.2.5 <i>Filtering</i> dengan menggunakan <i>Stoplist</i>	56	Universitas Brawijaya	Universitas Brawijaya	
4.2.6 Klasifikasi	63	Universitas Brawijaya	Universitas Brawijaya	
4.2.7 Hasil Analisis Sentimen	78	Universitas Brawijaya	Universitas Brawijaya	
4.3 Perancangan Pengujian	78	Universitas Brawijaya	Universitas Brawijaya	
BAB 5 IMPLEMENTASI	85	Universitas Brawijaya	Universitas Brawijaya	
5.1 Implementasi Kode Program	85	Universitas Brawijaya	Universitas Brawijaya	
5.1.1 Kode Program <i>Void Main</i>	85	Universitas Brawijaya	Universitas Brawijaya	
5.1.2 Kode Program <i>Import Library</i>	86	Universitas Brawijaya	Universitas Brawijaya	
5.1.3 Kode Program <i>Buka File</i>	87	Universitas Brawijaya	Universitas Brawijaya	
5.1.4 Kode Program <i>Cleaning</i>	87	Universitas Brawijaya	Universitas Brawijaya	
5.1.5 Kode Program <i>Tokenisasi</i>	88	Universitas Brawijaya	Universitas Brawijaya	
5.1.6 Kode Program <i>Case Folding</i>	88	Universitas Brawijaya	Universitas Brawijaya	
5.1.7 Kode Program Normalisasi kata	88	Universitas Brawijaya	Universitas Brawijaya	



DAFTAR TABEL

Tabel 2.1 Tabel Kajian Pustaka.....	8
Tabel 2.2 Contoh Perhitungan Raw Term Frequency	12
Tabel 2.3 <i>Confusion Matrix</i>	15
Tabel 4.1 Contoh <i>Cleaning</i>	40
Tabel 4.2 Contoh Tokenisasi	41
Tabel 4.3 Contoh <i>Case Folding</i>	42
Tabel 4.4 Contoh Normalisasi kata	44
Tabel 4.5 Contoh <i>Type</i>	45
Tabel 4.6 Contoh Perhitungan <i>Raw term frequency</i>	46
Tabel 4.7 Contoh Data untuk Perhitungan Manual TBRS	48
Tabel 4.8 Hasil Perhitungan TF.....	48
Tabel 4.9 Urutan Kata Berdasarkan Nilai bobot Kullback-Leibler.....	51
Tabel 4.10 Contoh <i>Lexicon Non Stopword</i>	53
Tabel 4.11 Daftar 10 Persen Kata untuk <i>Stoplist</i>	53
Tabel 4.12 Daftar 20 Persen Kata untuk <i>Stoplist</i>	54
Tabel 4.13 Daftar 30 Persen Kata untuk <i>Stoplist</i>	54
Tabel 4.14 Daftar 40 Persen Kata untuk <i>Stoplist</i>	54
Tabel 4.15 Daftar 50 Persen Kata untuk <i>Stoplist</i>	55
Tabel 4.16 Daftar 60 Persen Kata untuk <i>Stoplist</i>	56
Tabel 4.17 Daftar Kata Setelah <i>Filtering</i> dengan <i>Stoplist</i> 0 Persen dari Total Kata Data Latih	57
Tabel 4.18 Daftar Kata Setelah <i>Filtering</i> dengan <i>Stoplist</i> 10 Persen dari Total Kata Data Latih	58
Tabel 4.19 Daftar Kata Setelah <i>Filtering</i> dengan <i>Stoplist</i> 20 Persen dari Total Kata Data Latih	59
Tabel 4.20 Daftar Kata Setelah <i>Filtering</i> dengan <i>Stoplist</i> 30 Persen dari Total Kata Data Latih	60
Tabel 4.21 Daftar Kata Setelah <i>Filtering</i> dengan <i>Stoplist</i> 40 Persen dari Total Kata Data Latih	61
Tabel 4.22 Daftar Kata Setelah <i>Filtering</i> dengan <i>Stoplist</i> 50 Persen dari Total Kata Data Latih	61



Tabel 4.23 Daftar Kata Setelah <i>Filtering</i> dengan <i>Stoplist</i> 60 Persen dari Total Kata Data Latih	62
Tabel 4.24 Nilai <i>Raw term frequency (RTF)</i> Term pada Tiap Data Latih di Setiap Kelas	63
Tabel 4.25 Confusion Matrix Klasifikasi dengan <i>Stoplist</i> 0 persen, 10 persen, 20 persen dan 30 persen.....	78
Tabel 4.26 Confusion Matrix Klasifikasi dengan <i>Stoplist</i> 40 persen	80
Tabel 4.27 Confusion Matrix Klasifikasi dengan <i>Stoplist</i> 50 persen dan 60 persen	82
Tabel 6.1 Cuplikan Hasil Klasifikasi Data Uji dengan <i>Stoplist</i> 0 Persen	99
Tabel 6.2 <i>Confusion Matrix</i> Hasil Klasifikasi Sistem	100
Tabel 6.3 Cuplikan Hasil Klasifikasi Data Uji dengan <i>Stoplist</i> 20 Persen.....	103
Tabel 6.4 <i>Confusion Matrix</i> Hasil Klasifikasi Sistem	104
Tabel 6.5 Cuplikan Hasil Klasifikasi Data Uji dengan <i>Stoplist</i> 30 Persen.....	106
Tabel 6.6 <i>Confusion Matrix</i> Hasil Klasifikasi Sistem	107
Tabel 6.7 Cuplikan Hasil Klasifikasi Data Uji dengan <i>Stoplist</i> 40 Persen.....	110
Tabel 6.8 <i>Confusion Matrix</i> Hasil Klasifikasi Sistem	110
Tabel 6.9 Cuplikan Hasil Klasifikasi Data Uji dengan <i>Stoplist</i> 50 Persen.....	113
Tabel 6.10 <i>Confusion Matrix</i> Hasil Klasifikasi Sistem	114
Tabel 6.11 Cuplikan Hasil Klasifikasi Data Uji dengan <i>Stoplist</i> 60 Persen.....	117
Tabel 6.12 <i>Confusion Matrix</i> Hasil Klasifikasi Sistem	118



DAFTAR GAMBAR

Gambar 3.1 Diagram Alir Proses Pelatihan Sistem	19
Gambar 3.2 Diagram Alir Proses Pengujian Sistem	20
Gambar 4.1 Diagram Alir <i>Preprocessing</i>	22
Gambar 4.2 Diagram Alir <i>Cleaning</i>	23
Gambar 4.3 Diagram Alir Pengubahan Normalisasi kata.....	26
Gambar 4.4 Diagram Alir <i>Filtering</i>	27
Gambar 4.5 Diagram Alir <i>Filtering</i> dengan <i>Stoplist</i>	27
Gambar 4.6 Diagram Alir <i>Term-Based Random Sampling</i>	28
Gambar 4.7 Diagram Alir Membuat <i>Inverted Index</i>	29
Gambar 4.8 Diagram Alir Hitung <i>Px</i>	30
Gambar 4.9 Diagram Alir Hitung Nilai Bobot Kullback-Leibler	31
Gambar 4.10 Diagram Alir <i>Random Sampling</i>	32
Gambar 4.11 Diagram Alir Mengurutkan Term Berdasarkan Nilai Kullback-Leibler	33
Gambar 4.12 Diagram Alir Membuat <i>Stoplist</i>	34
Gambar 4.13 Diagram Alir Perhitungan <i>Raw term frequency</i>	35
Gambar 4.14 Diagram Alir Perhitungan <i>Term Frequency</i>	36
Gambar 4.15 Diagram Alir Perhitungan <i>Raw term frequency</i>	37
Gambar 4.16 Diagram Alir Klasifikasi Naïve Bayes.....	37
Gambar 4.17 Diagram Alir Hitung <i>Likelihood</i>	38
Gambar 4.18 Diagram Alir Hitung <i>Posterior</i>	39
Gambar 5.1 Contoh Hasil Keluaran Kode Program 5.18	96
Gambar 5.2 Contoh Hasil Keluaran Kode Program 5.19	97
Gambar 5.3 Contoh Hasil Keluaran Kode Program 5.20.....	98
Gambar 6.1 Grafik Evaluasi Sistem dengan <i>Stoplist</i> 0 Persen dan 10 Persen	102
Gambar 6.2 Grafik Evaluasi Sistem dengan <i>Stoplist</i> 20 Persen	106
Gambar 6.3 Grafik Evaluasi Sistem dengan <i>Stoplist</i> 30 Persen	109
Gambar 6.4 Grafik Evaluasi Sistem dengan <i>Stoplist</i> 40 Persen	113
Gambar 6.5 Grafik Evaluasi Sistem dengan <i>Stoplist</i> 50 Persen	116
Gambar 6.6 Grafik Evaluasi Sistem dengan <i>Stoplist</i> 60 Persen	120



Gambar 6.7 Grafik *Macroaverage* Evaluasi Keseluruhan 120

Gambar 6.8 Persentase Sentimen Warganet Twitter Berdasarkan Klasifikasi dengan Stoplist 20 persen 121



BAB 1 PENDAHULUAN

Pada bab pendahuluan ini dijelaskan latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika pembahasan dari penelitian yang dibuat.

1.1 Latar Belakang

Berdasarkan portal berita daring BBC (2019), pada sidang bersama DPD-DPR yang diadakan 16 Agustus 2019 lalu, Presiden Joko Widodo menyatakan bahwa Ibukota Republik Indonesia akan dipindahkan ke pulau Kalimantan, tepatnya di provinsi Kalimantan Timur, yaitu sebagian daerah Penajam Paser Utara dan Kutai Kartanegara. Dalam konferensi pers, presiden juga menyebut bahwa kebijakan pemindahan Ibukota harus dilakukan karena dua masalah utama. Pertama, Jakarta dianggap sudah terlalu berat bebananya sebagai pusat pemerintahan, pusat ekonomi, pusat perdagangan, pusat keuangan dan pusat jasa. Kedua, sebanyak 150 juta warga Indonesia berada di pulau Jawa, yang artinya 54 persen penduduk Indonesia berada di Pulau Jawa. Oleh karenanya, Presiden Joko Widodo beranggapan bahwa akan sangat membebani bila Ibukota dipindahkan ke daerah lainnya di pulau Jawa.

Kebijakan terkait pemindahan Ibukota ini pun tentunya menjadi perhatian seluruh rakyat Indonesia. Di media sosial, ramai warga net mengemukakan pendapatnya terhadap kebijakan ini. Ada yang menilai positif kebijakan ini dengan berbagai alasan, salah satunya yaitu untuk pemerataan ekonomi dan sektor lainnya, agar semuanya tidak tersentralisasi di pulau Jawa saja. Namun, di sisi lain, banyak juga warga net yang beranggapan negatif dengan alasan yang bermacam-macam, salah satunya yaitu pemindahan Ibukota dianggap pemborosan. Melihat pemberitaan terkait kebijakan pemindahan Ibukota Indonesia yang begitu masif dan menjadi perbincangan seluruh rakyat Indonesia, utamanya warga net di sosial media, sehingga analisis data terkait sentimen masyarakat terhadap kebijakan ini sulit untuk dilakukan secara manual mengingat banyaknya data yang perlu dikumpulkan dan dianalisis untuk mendapatkan hasil analisis sentimen yang kredibel. Analisis data untuk mengetahui sentimen dari masyarakat terkait kebijakan ini dilakukan untuk keperluan yang bermacam-macam, contohnya untuk keperluan statistik data dalam pemberitaan, sehingga dengan disertakannya analisis data yang lebih mendalam, berita menjadi lebih faktual dan kredibel. Oleh karena itu, perlu dibuat suatu sistem untuk melakukan analisis sentimen yang bertujuan untuk memudahkan proses analisis data terkait sentimen masyarakat terhadap suatu permasalahan tertentu, baik dari segi efisiensi waktu maupun dari segi efisiensi biaya.

Terkait kebijakan yang sedang masif dibicarakan ini, penulis melakukan penelusuran jurnal tentang analisis sentimen kebijakan pemindahan Ibukota Indonesia melalui mesin pencari Google dan hingga saat ini masih belum terdapat penelitian yang membahas tentang hal tersebut, sehingga dapat disimpulkan



bahwa penelitian analisis sentimen terkait permasalahan ini masih sangat minim dilakukan hingga saat ini.

Terdapat beberapa metode analisis sentimen yang dapat digunakan. Dalam penelitian yang dilakukan oleh Vohra dan Teraiya (2014), dijelaskan bahwa terdapat 3 jenis metode analisis sentimen, yaitu metode *machine learning based*, *lexicon based* dan *hybrid*. Metode *machine learning based* merupakan metode yang sifatnya *supervised*. Beberapa contoh metode *machine learning based* diantaranya adalah Naïve Bayes, Maximum Entropy, Neural Network dan Support Vector Machine (SVM). Metode *lexicon based* merupakan metode *unsupervised* sehingga proses klasifikasinya dilakukan dengan membandingkan fitur-fitur dari teks dengan *lexicon* yang ada. Apabila suatu dokumen mengandung kata-kata yang terdapat dalam *lexicon* positif, maka sentimen dari dokumen tersebut akan cenderung mengarah ke sentimen positif, dan sebaliknya. Sedangkan metode *hybrid* adalah metode yang menggabungkan *machine learning based* dan *lexicon based*. Pada umumnya, metode *hybrid* menggunakan *lexicon* sebagai *classifier* awal, kemudian metode *machine learning based* digunakan untuk mengembangkan pembendaharaan kata dalam *lexicon* yang telah ada.

Dalam penelitian yang dilakukan oleh Pintoko dan Muslim (2018), dibuat sistem analisis sentimen terhadap jasa transportasi *online* dengan data yang didapatkan melalui media sosial Twitter dan menggunakan metode klasifikasi Naïve Bayes. Data yang digunakan yaitu data twit yang menyebut nama akun Twitter resmi Gojek dan Grab. Terdapat 2000 data yang digunakan, 1500 dari data tersebut digunakan sebagai data latih dan sisanya digunakan sebagai data uji. Hasil dari penelitian dalam jurnal tersebut didapatkan nilai *recall* sebesar 90,29 persen, *precision* sebesar 94,56 persen, *f-measure* sebesar 92,38 persen dan akurasi sebesar 86,80 persen. Melihat hasil evaluasi yang sangat baik dari jurnal tersebut, sehingga pada penelitian skripsi ini, penulis menggunakan metode Naïve Bayes untuk melakukan analisis sentimen. Selain itu, dalam penelitian skripsi ini juga akan digunakan algoritme *term-based random sampling* untuk membuat *stopword list* yang akan digunakan dalam proses *filtering* pada *preprocessing*. Dalam penelitian yang dilakukan oleh Lo et al. (2005), digunakan beberapa koleksi *lexicon* untuk membandingkan *precision* antara algoritme *term-based random sampling* dan *classical stopword list* dalam melakukan *filtering* untuk proses temu kembali informasi. Koleksi dokumen yang digunakan yaitu disk45, WT2G, WT10G dan DOTGOV. Berdasarkan hasil dari penelitian tersebut dijelaskan bahwa penggunaan *term-based random sampling* meningkatkan nilai rata-rata *precision* dibandingkan dengan penggunaan *classical stopword list*, seperti *fox stopword list* untuk semua koleksi dokumen yang digunakan. Melihat hasil penelitian dalam jurnal tersebut, penulis memutuskan untuk menggunakan algoritme *term-based random sampling* untuk membuat *stopword list* secara otomatis dalam penelitian skripsi ini, sehingga diharapkan analisis sentimen yang akan dilakukan mendapatkan nilai *precision* yang baik.

1.2 Rumusan Masalah

Berdasarkan apa yang telah dijabarkan dalam latar belakang, didapatkan rumusan masalah sebagai berikut:

1. Bagaimana penerapan metode Naïve Bayes dan *term-based random sampling* untuk keperluan analisis sentimen?
2. Bagaimakah pendapat warga net Twitter terkait kebijakan pemindahan Ibukota?
3. Berapakah hasil evaluasi metode yang digunakan untuk melakukan proses analisis sentimen?

1.3 Tujuan

Dari rumusan masalah yang didapatkan, maka diketahui tujuan dari penelitian yang dilakukan yaitu:

1. Merancang algoritme pembuatan sistem untuk menganalisis sentimen terkait kebijakan pemindahan Ibukota Indonesia dengan menggunakan metode Naïve Bayes dan *term-based random sampling*.
2. Mengetahui pendapat mayoritas warga net media sosial Twitter terkait kebijakan pemindahan Ibukota berdasarkan hasil analisis sentimen.
3. Menghitung nilai evaluasi sistem yang digunakan untuk melakukan proses analisis sentimen yang menggunakan metode Naïve Bayes dan *term-based random sampling*.

1.4 Manfaat

Manfaat dilihat dari berbagai sisi, antara lain:

1. Memberikan penjelasan detail tentang penerapan dan implementasi metode klasifikasi Naïve Bayes dan *term-based random sampling* untuk keperluan analisis sentimen.
2. Memberikan penjelasan terkait pendapat mayoritas warganet pada media sosial Twitter terkait kebijakan pemindahan Ibukota negara Indonesia.
3. Memberikan penjelasan terkait evaluasi metode yang mencerminkan tingkat liabilitas dan efektifitas metode Naïve Bayes dan *term-based random sampling* untuk keperluan analisis sentimen.

1.5 Batasan Masalah

Penelitian yang dilakukan dibatasi oleh beberapa hal, yaitu:

1. Data yang digunakan yaitu twit yang mengandung kata kunci yang berhubungan dengan topik pemindahan Ibukota, antara lain kata kunci “Ibukota pindah”, “pemindahan Ibukota” dan “ibukota dipindahkan”.

2. Parameter yang diambil untuk dilakukan analisis sentimen yaitu hanya isi dari twit saja.
3. Normalisasi kata dilakukan dengan menggunakan korpus yang dibuat dari data yang telah didapatkan untuk keperluan pengujian.
4. Kata yang dinormalisasi adalah kata ambigu atau frasa bernegasi yang sering muncul pada data twit.

1.6 Sistematika Pembahasan

Guna memudahkan dalam memahami penulisan tiap-tiap bab dalam pembuatan skripsi ini, maka akan dijabarkan secara singkat sistematika penulisan proposal yang terdiri dari:

BAB 1 PENDAHULUAN

Pendahuluan berisi latar belakang penelitian analisis sentimen terkait kebijakan pemindahan Ibukota negara Indonesia dengan menggunakan algoritme *term-based random sampling* dan metode klasifikasi Naïve Bayes , rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika pembahasan.

BAB 2 TINJAUAN PUSTAKA

Tinjauan pustaka berisi tinjauan dari beberapa studi pustaka, yaitu menjelaskan tentang teori-teori yang terkait dengan permasalahan yang diambil, sebagai acuan dalam analisa dan pemecahan masalah dari studi pustaka yang berkaitan dengan permasalahan yang dibahas dan nantinya akan memudahkan penulis dalam menyelesaikan dan memecahkan masalah.

BAB 3 METODOLOGI PENELITIAN

Metodologi penelitian berisi langkah-langkah yang digunakan dalam penelitian ini agar terstruktur dengan baik. Dengan sistematika ini proses penelitian dapat dipahami dan diikuti oleh pihak lain. Bab ini meliputi tipe penelitian, strategi penelitian, peralatan pendukung, lokasi penelitian, teknik pengumpulan data, data penelitian, teknik analisis data, perancangan algoritme, dan teknik penerapan metode.

BAB 4 PERANCANGAN

Dalam perancangan dijabarkan perancangan terkait langkah-langkah yang dilakukan dalam implementasi metode dalam melakukan penelitian. Perancangan yang dijabarkan antara lain, diagram alir sistem, perhitungan manual dan perancangan pengujian.

BAB 5 IMPLEMENTASI

Bab Implementasi berisi kode program yang dibuat berdasarkan perancangan sistem pada Bab Perancangan. Selain itu, juga akan dijelaskan secara singkat



terkait fungsi dan tujuan dari tiap kode program. Gambar hasil keluaran kode program saat dijalankan juga disertakan apabila kode program mencetak hasil keluaran saat dijalankan.

BAB 6 HASIL DAN PEMBAHASAN

Hasil berfungsi untuk melaporkan hasil pelaksanaan metode/teknik penelitian dan menyajikan data yang mendukung hasil tersebut. Penyajian data dan penjelasannya dilakukan secara terurut dan logis menggunakan teks dan ilustrasi lainnya (misalnya, tabel dan gambar). Urutan penjelasan dapat dilakukan secara kronologis berdasarkan urutan pelaksanaan metode atau berdasarkan tingkat kepentingan substansinya, dari yang lebih penting sampai ke yang prioritasnya lebih rendah.

Pembahasan berfungsi untuk menerjemahkan makna dari hasil yang diperoleh untuk menjawab pertanyaan atau masalah penelitian. Fungsi lainnya adalah untuk menjelaskan pemahaman baru yang didapatkan dari hasil penelitian, yang diharapkan berguna dalam pengembangan keilmuan.

BAB 7 PENUTUP

Penutup terdiri dari dua, yaitu kesimpulan dan saran. Kesimpulan berisi rangkuman secara singkat dari hasil pembahasan masalah. Saran berisi harapan dan kemungkinan lebih lanjut dari hasil pembahasan masalah yang diperoleh untuk menuju lebih baik.



BAB 2 TINJAUAN PUSTAKA

Dalam bab tinjauan pustaka dijelaskan kajian pustaka dan landasan pustaka yang menjadi sumber referensi dalam penelitian ini.

2.1 Kajian Pustaka

Dalam penelitian yang dilakukan oleh Fajar Ratnawati (2018), dibuat sistem analisis sentimen pada Twitter untuk mengetahui opini masyarakat tentang film dari twit berbahasa indonesia. Penelitian ini menggunakan API *search* Twitter untuk mengumpulkan data dari Twitter yang diperoleh dengan membuat program *scraping* menggunakan *library scrapy* yang sudah di sediakan oleh python. Metode yang digunakan adalah Naïve Bayes *classifier* untuk mengelompokkan data ke dalam sentimen positif dan sentimen negatif. Pada pengukuran performa klasifikasi, dihitung dengan menghitung akurasi, *precision*, *recall* dan *f-measure*. Pada penelitian ini digunakan 200 twit positif dan 200 twit negatif untuk data latih dan 100 twit untuk data uji. Pengujian ini dibagi menjadi 5 *fold*, masing-masing *fold* terdiri dari 100 twit. Akurasi tertinggi didapatkan pada saat pengujian *fold* ke-1 yaitu sebesar 90 persen. *Precision* tertinggi juga didapatkan pada saat pengujian *fold* ke-1 yaitu sebesar 92 persen. Hasil *recall* dan *f-measure* tertinggi juga didapatkan pada saat *fold* ke-1 sebesar 90 persen untuk *recall* dan 90 persen untuk *f-measure*.

Dalam penelitian yang dilakukan oleh Nurhuda et al. (2013), dibuat sistem analisis sentimen masyarakat terhadap Calon Presiden Indonesia 2014 berdasarkan opini pada Twitter. Penelitian tersebut menggunakan metode klasifikasi Naïve Bayes. Data untuk keperluan penelitian didapatkan dari twit berbahasa Indonesia yang merupakan hasil pencarian berdasarkan percakapan seseorang terhadap akun resmi tokoh politik yang akan maju sebagai calon Presiden Indonesia tahun 2014. Beberapa akun resmi yang diambil yaitu, @jokowi_do2 (Joko Widodo), @Prabowo08 (Prabowo Subianto), @Pak_JK (Jusuf Kalla), @hattarajasa (Hatta Rajasa). Penelitian ini menggunakan API *search* Twitter untuk mengumpulkan data dari Twitter yang kemudian disimpan ke dalam *database*. Penelitian ini menghasilkan akurasi sebesar 90 persen. Hasil dari penelitian terhadap pasangan calon presiden dan wakil presiden menunjukkan pasangan Prabowo Subianto–Hatta Rajasa mendapatkan 47,7 persen untuk sentimen positif, 26,4 persen sentimen negatif dan 25,9 persen sentimen netral. Sementara itu, pasangan Joko Widodo–Jusuf Kalla mendapatkan 37,6 persen sentimen positif, 34,4 persen sentimen negatif, dan 27,9 persen sentimen netral.

Pada penelitian yang dilakukan oleh Pintoko dan Muslim (2018), dibuat sistem analisis sentimen terhadap jasa transportasi *online* pada Twitter. Metode yang digunakan adalah Naïve Bayes *Classifier*. Analisis dilakukan berdasarkan twit yang melakukan *mention* kepada akun resmi Gojek dan Grab. Pengambilan data dilakukan dengan menggunakan API Twitter dan didapatkan total 2000 data. Dari jumlah data tersebut, 1500 data digunakan untuk data latih dan 500 data digunakan untuk data uji. Data tersebut berdasarkan twit yang *mention* ke akun

@gojekindonesia sebanyak 1000 data dan @grabID sebanyak 1000 data. Hasil dari penelitian ini yaitu didapatkan *recall* sebesar 90,29 persen, *precision* sebesar 94,56 persen, dan *f-measure* sebesar 92,38 persen. Didapatkan pula hasil sentimen positif sebesar 88,60 persen dan sentimen negatif 11,40 persen dengan akurasi sebesar 86,80 persen.

Dalam penelitian yang dilakukan oleh Listari et al. (2019), dibuat sistem analisis sentimen terhadap bom bunuh diri di Surabaya 13 Mei 2018 dengan menggunakan data dari Twitter. Pengambilan data dilakukan menggunakan API Twitter. Variabel penelitian yang digunakan yaitu teks yang merupakan twit dari pengguna, *score* yang merupakan nilai klasifikasi berdasarkan kamus, dan klasifikasi yang merupakan mengklasifikasi setiap twit menjadi 2 kategori yaitu positif dan negatif. Metode yang digunakan dalam penelitian ini adalah Support Vector Machine (SVM). Data yang diambil merupakan twit sebanyak 2042 twit melalui kata kunci "Surabaya, bom, dan teroris". Setelah dilakukan analisis, didapatkan sentimen negatif sebanyak 1921 twit sedangkan sentimen positif sebanyak 121 twit. Akurasi yang didapatkan menggunakan pendekatan SVM yaitu sebesar 100 persen.

Dalam penelitian yang dilakukan oleh Pravina et al. (2019), dibuat sistem analisis sentimen tentang opini maskapai penerbangan menggunakan algoritme Support Vector Machine (SVM) dengan fitur Lexicon Based. Data didapatkan dari Twitter. Dari penelitian didapatkan nilai parameter *learning rate* (*gamma*) sebesar 0,03 dan nilai C sebesar 10 sebagai nilai parameter paling optimal. Tingkat akurasi paling besar sebesar 40 persen, *precision* 40 persen, *recall* 100 persen, dan *f-measure* sebesar 57,14 persen. Hasil tersebut didapatkan dengan jumlah iterasi maksimum sebanyak 50 kali.

Pada penelitian yang dilakukan oleh Maulana Aziz Assuja dan Saniati Saniati (2016), dibuat sistem analisis sentimen dengan data dari media sosial Twitter dilakukan dengan menggunakan metode Backpropagation Neural Network (BNN). Pada penelitian tersebut digunakan total sebanyak 944 data twit, 630 data twit digunakan untuk data latih dan 314 data twit untuk data uji. Data latih terbagi menjadi 500 data twit yang berlabel netral, 254 data twit berlabel positif dan 190 data twit yang berlabel negatif. Berdasarkan hasil evaluasi sistem, didapatkan nilai akurasi terbaik sebesar 78,34 persen dan nilai *precision* terbaik sebesar 84,21 persen.

Pada penelitian yang dilakukan oleh Alvi Pranandha Syah, Adiwijaya, dan Said Al Faraby (2017), dibuat sistem analisis sentimen terhadap data ulasan produk toko *online* dengan menggunakan metode Maximum Entropy. Data yang digunakan pada penelitian dalam jurnal ini, yaitu data ulasan produk dari toko *online* Amazon yang terfokuskan pada kategori *cell phones* dan *accessories*. Proses kerja sistem secara urut yaitu pengambilan *dataset*, *data splitting*, *preprocessing*, *feature extraction* dan klasifikasi menggunakan Maximum Entropy. Sementara untuk hasil evaluasi sistem didapatkan nilai akurasi terbaik yaitu 83 persen dan nilai *f-1 score* sebesar 90.074 persen pada iterasi ke 10000.

Tabel 2.1 merupakan tabel kajian pustaka yang berisi judul, objek, metode, penulis dan tahun dari setiap jurnal penelitian.

Tabel 2.1 Tabel Kajian Pustaka

No	Judul	Objek	Metode	Penulis	Tahun
1.	Implementasi Algoritme Naïve Bayes Terhadap Analisis Sentimen Opini Film Pada Twitter	Data twit yang mengandung tagar #judul film yang diinginkan	Naïve Bayes	Ratnawati, F.Brawijaya	2018
2.	Analisis Sentimen Masyarakat terhadap Calon Presiden Indonesia 2014 berdasarkan Opini dari Twitter Menggunakan Metode Naïve Bayes Classifier	Twit percakapan seseorang terhadap akun @jokowi_do2, @Prabowo08, @Pak_JK, @hattarajasa	Naïve Bayes	Nurhuda, F., Sihwi, S.W., & Doewes, A.	2013
3.	Analisis Sentimen Jasa Transportasi Online pada Twitter Menggunakan Metode Naïve Bayes Classifier	Twit mention ke akun @gojekindonesia dan @grabID	Naïve Bayes	Pintoko, B.M., & Muslim, K.	2018
4.	Analisis Sentimen Twitter terhadap Bom Bunuh Diri di Surabaya 13 Mei 2018 menggunakan Pendekatan Support Vector Machine	Twit yang memiliki kata kunci “Surabaya, bom, versitas dan teroris”	Support Vector Machine	Listari, Ihsan, M., Paradistia, E.R., & Widodo, E.	2019
5	Analisis Sentimen	Twit yang berhubungan	Support Vector Machine	Pravina, A. M.,	2019

	Tentang Opini Maskapai Penerbangan pada Dokumen Twitter Menggunakan Algoritme Support Vector Machine (SVM)	dengan berbagai maskapai		Cholissodin, I. & Adikara, P. P.	
6	Analisis Sentimen dengan Data dari Media Sosial Twitter dilakukan dengan menggunakan Metode Backpropagation Neural Network (BNN)	Data twit dari media sosial Twitter	Backpropagation Neural Network	Assuja, M. A. & Saniati	2016
7	Analisis Sentimen terhadap Data Ulasan Produk Toko Online dengan menggunakan Metode Maximum Entropy	Data produk ulasan dari e-commerce Amazon	Maximum Entropy	Syah, A. P., Adiwijaya & Faraby, S. A.	2017

2.2 Landasan Teori

Dalam landasan pustaka dijelaskan topik-topik penting yang perlu diketahui dalam melakukan penelitian ini.

2.2.1 Text Mining

Text Mining memiliki definisi menambang data berupa teks dokumen yang bertujuan untuk mencari pola tersembunyi yang terdapat pada dokumen-dokumen teks, pola yang telah ditemukan kemudian akan digunakan untuk suatu tujuan tertentu (Indriati & Ridok, 2016). Dalam Text Mining terdapat teknik *preprocessing* yaitu suatu teknik memproses data teks sedemikian rupa hingga data teks tersebut dirasa dapat digunakan untuk proses penelitian lebih lanjut.

Dalam penelitian skripsi ini pun diimplementasikan teknik *preprocessing* pada data

twit yang ditujukan untuk melakukan suatu normalisasi kata-kata dalam data twit hingga sedemikian rupa agar data twit dapat digunakan untuk proses penelitian yang selanjutnya.

Terdapat beberapa tahapan *preprocessing* yang dilakukan dalam penelitian ini, yaitu:

Cleaning

Cleaning adalah proses menghapus karakter-karakter tanda baca dan tautan yang terdapat pada dokumen. Proses ini dilakukan untuk menghindari kesalahan interpretasi *term* oleh sistem karena adanya tanda baca yang mengikuti *term* tersebut. Sementara itu, tautan perlu dihapus dari dokumen karena tidak digunakan dalam proses analisis sentimen.

Tokenisasi

Tokenisasi merupakan proses pemisahan dokumen teks menjadi kata per kata. Tahap ini dilakukan *Cleaning* untuk mempermudah proses analisis, karena analisis sentimen pada penelitian ini menggunakan fitur *bag of words*, sehingga analisis dilakukan secara kata per kata.

Case Folding

Dalam tahap ini, data yang telah ditokenisasi berupa kata per kata kemudian akan dikonversi menjadi kata-kata dalam huruf kecil. Tahap ini dilakukan untuk menghindari terjadinya kesalahan interpretasi *term* oleh sistem karena adanya kapitalisasi pada *term*.

Normalisasi kata

Pada tahap ini dilakukan penggantian kata-kata yang bersifat ambigu, seperti misalnya kata yang disingkat, menjadi kata yang baku. Selain itu dalam tahap ini juga dilakukan proses penggantian kata yang mengandung kata tidak menjadi kata sinonimnya yang tidak mengandung kata tidak, hal ini dilakukan karena kata tidak akan hilang pada saat tahap *filtering*, sehingga apabila hal itu terjadi, akan mengubah makna yang dimaksud dan akan berpengaruh pada hasil analisis sentimen.

Filtering

Pada tahap ini pada umumnya digunakan *stopword list*, yaitu dengan membuang kata-kata pada dokumen yang dianggap kurang penting. Tahap ini dilakukan untuk mengefisiensi proses analisis dokumen pada tahap selanjutnya.

Stemming

Stemming adalah proses perubahan kata-kata berimbuhan menjadi kata dasar. Proses ini dilakukan untuk mempermudah proses analisis sentimen, sehingga proses analisis dapat dilakukan dengan lebih efisien karena kata yang berimbuhan akan diubah menjadi bentuk kata dasarnya.

- Type

Pada tahap ini dilakukan penghapusan kata-kata yang muncul lebih dari satu kali. Kata-kata yang berlebih tersebut akan dihapus sehingga kata akan muncul hanya satu kali dalam dokumen. Hasil dari proses ini disebut sebagai *term* yang kemudian digunakan untuk proses klasifikasi.

2.2.2 Analisis Sentimen

Menurut Walaa Medhat, Ahmed Hassan, dan Hoda Korashy (2014), analisis sentimen adalah suatu bidang pembelajaran komputasional terhadap pendapat, sikap, dan emosi seseorang terhadap suatu entitas. Entitas tersebut dapat berupa individu, peristiwa atau topik pembicaraan. Analisis sentimen merupakan proses memahami, mengekstrak dan mengolah data teksual secara otomatis untuk mendapatkan informasi sentimen yang terkandung dalam suatu kalimat opini. Analisis sentimen dilakukan untuk mengetahui kecenderungan opini terhadap sebuah masalah atau objek oleh seseorang, apakah cenderung beropini negatif atau positif (Rozi, et al., 2012).

2.2.3 Media Sosial

Media sosial merupakan sekelompok aplikasi berbasis internet yang dibangun atas dasar ideologi dan teknologi Web 2.0, dan memungkinkan penciptaan dan pertukaran *user-generated content* (Kaplan & Haenlein, 2010). Media sosial dapat ditemukan dalam berbagai bentuk, seperti *social network*, *weblogs*, *social blogs*, dll. Menurut Kaplan dan Haenlein (2010), ada enam jenis media sosial diantaranya: proyek kolaborasi, *blog* dan *microblogs*, komunitas konten, situs jaringan sosial, *virtual game*, dan *virtual social*.

Media sosial merupakan situs yang dapat digunakan setiap orang untuk membuat *web page* pribadi, kemudian terhubung dengan teman–teman untuk berbagi informasi dan berkomunikasi. Media sosial terbesar saat ini contohnya Facebook, Twitter, dan Instagram. Berkembang pesatnya penggunaan media sosial dikarenakan sekarang manusia dapat memiliki media mereka sendiri. Pengguna media sosial dapat mengaksesnya menggunakan jaringan internet bahkan yang aksesnya lambat sekalipun, tanpa biaya yang besar, dan dapat dilakukan sendiri.

2.2.4 Twitter

Twitter merupakan salah satu situs jejaring sosial terbesar yang sangat digemari oleh berbagai kalangan masyarakat. Selain digunakan untuk menyalurkan pikiran, Twitter juga dapat digunakan untuk mencari informasi berita, hiburan, dan lainnya. Twitter merupakan *micro-blogging* yang artinya penggunaan karakter dalam Twitter sangat terbatas sehingga dala penggunaanya

Twitter lebih ringan dibandingkan dengan media sosial lainnya. Pada awal perilisannya, Twitter hanya dapat menggunakan 140 karakter saja, tetapi saat ini sudah diperbarui menjadi 280 karakter.

Terdapat beberapa fitur yang dapat digunakan pada Twitter antara lain *hashtag*, *direct message*, *link picture*, *video* dan masih banyak lagi. *Hashtag* merupakan salah satu fitur andalan Twitter. Fitur ini dapat digunakan untuk mengkategorikan *tweet-tweet* pengguna sehingga sangat efektif untuk mencari dan menampilkan *tweet-tweet* yang terkait dengan hashtag tersebut. Fitur lainnya yang tidak kalah menarik adalah fitur *follower*. Pengguna dapat melihat berapa dan siapa saja yang orang telah mengikuti akunnya.

2.2.5 Pembobotan *Raw Term Frequency* (RTF)

Berdasarkan penelitian yang dilakukan oleh Hananto et al. (2018) *raw term frequency* dilakukan dengan menghitung jumlah kata unik pada suatu dokumen.

Nilai *raw term frequency* dihitung pada tiap *term* dokumen yang dihasilkan pada proses *preprocessing* data latih. Langkah awal yang dilakukan yaitu menghitung nilai *term frequency* dari tiap *term* pada tiap dokumen. Setelah nilai *term frequency* dari tiap *term* dihitung, kemudian nilai *raw term frequency* dari *term* tersebut dapat diketahui dengan menjumlahkan *term frequency* dari *term* tersebut pada tiap dokumen. Proses ini dilakukan untuk seluruh *term* yang dihasilkan pada proses *preprocessing* data latih. Pada Tabel 2.2 ditunjukkan contoh perhitungan *raw term frequency* dari 2 data berikut:

D1 : Usahlah bersedih, semoga Ibukota gak pindah,dan batal dilantik,Aamiin..

D2 : Tanpa Payung Hukum, Rencana Jokowi Pindah Ibukota Cacat Prosedur

Tabel 2.2 Contoh Perhitungan Raw Term Frequency

No	Term	D1	D2	Raw term frequency
1	Ibukota	1	1	2
2	pindah	1	1	2
3	usahlah	1	0	1
4	bersedih	1	0	1
5	semoga	1	0	1
6	tidak	1	0	1
7	dan	1	0	1
8	batal	1	0	1
9	dilantik	1	0	1
10	aamiin	1	0	1
11	tanpa	0	1	1

No	Term	D1	D2	Raw term frequency
12	payung	0	1	1
13	hukum	0	1	1
14	rencana	0	1	1
15	jokowi	0	1	1
16	cacat	0	1	1
17	prosedur	0	1	1

2.2.6 Metode Klasifikasi Naïve Bayes

Menurut Mark Hall (2006), metode klasifikasi Naïve Bayes merupakan salah satu metode yang banyak digunakan untuk keperluan *data mining* karena penggunaannya yang sederhana. Sementara itu, menurut Jie Cheng dan Russel Greiner (1999), Naïve Bayes mudah dipelajari dan dipahami, efisien, dan cenderung memiliki hasil akurasi yang tinggi. Naïve Bayes memanfaatkan pendekatan untuk sebuah ketidakpastian yang diukur dengan probabilitas. Secara umum teorema ini dapat dinotasikan dengan Persamaan 2.1.

$$\text{Posterior} = \text{Prior} * \text{Likelihood}$$

$$P(c|w) = P(c) * \prod_{w \in d} P(w|c)$$
(2.1)

Keterangan:

$P(c|w)$ = probabilitas *posterior term w* pada kelas c

$P(c)$ = probabilitas *prior* kelas c

$P(w|c)$ = probabilitas *likelihood term w* pada kelas c

Pada perhitungan klasifikasi, *Prior* digunakan untuk menghitung peluang kemunculan kategori pada semua dokumen. Perhitungan prior dapat dilihat pada Persamaan 2.2.

$$P(c) = \frac{N_c}{N}$$
(2.2)

Keterangan:

N_c = Banyak dokumen berkategori c pada dokumen latih

N = Jumlah dokumen latih yang digunakan

Likelihood atau $P(w|c)$ merupakan peluang sebuah kata masuk ke dalam sebuah kategori. *Likelihood* dapat dihitung dengan menggunakan formula *Multinomial Model* dengan *Laplace Smoothing*. Kelas dokumen tidak hanya



ditentukan dengan kata yang muncul tetapi juga jumlah kemunculan katanya.

Perhitungan *Likelihood* menggunakan *Multinomial Model* dengan *Laplace Smoothing* dapat dilihat pada Persamaan 2.3.a

$$\text{Un}P(w|c) = \frac{\text{count}(w,c)+1}{\text{count}(c)+|\nu|}$$

Keterangan:

$\text{count}(w, c)$ = jumlah kemunculan kata w pada kategori c

$\text{count}(c)$ = jumlah total kemunculan semua kata pada kategori c

$|\nu|$ = jumlah term unik

2.2.7 Term-Based Random Sampling

Term-based random sampling adalah sebuah algoritme untuk membuat *stopword list* secara otomatis berdasarkan tingkat kepentingan dari suatu *term* (Lo, et al., 2005). Penting tidaknya suatu *term* dapat diketahui dari seberapa informatif *term* tersebut. Untuk menilai seberapa penting suatu *term*, digunakan rumus dari teori Kullback-Leibler. Pada Persamaan 2.4 dijabarkan rumus dari teori Kullback-Leibler untuk pembobotan *term*.

$$w(t) = Px \cdot \log_2 \frac{Px}{Pc} \quad (2.4)$$

Keterangan:

$w(t)$ = bobot *term* t pada dokumen sampel

$$Px = \frac{tfx}{lx}$$

$$Pc = \frac{F}{tokenc}$$

tfx = frekuensi dari *query term* dalam dokumen sampel

lx = jumlah dari panjang kumpulan dokumen sampel

F = *term frequency* dari *query term* dalam koleksi dokumen

$tokenc$ = total *token* yang terdapat dalam seluruh koleksi dokumen

Pada sistem nantinya proses perhitungan bobot *term* Kullback-Leibler dilakukan dengan menggunakan *Simple Random Sampling*. *Simple Random Sampling* diimplementasikan untuk memilih satu *term* dari seluruh *term* yang ada secara acak, kemudian *term* yang terpilih akan dihitung nilai bobot *term* Kullback-Leiblernya. *Simple Random Sampling* dilakukan oleh sistem sebanyak Y kali. Y



merupakan parameter yang nilainya dapat diubah secara manual untuk mencari nilai Y dengan hasil klasifikasi terbaik.

Hasil dari perhitungan bobot tiap *term* menggunakan teori Kullback-Leibler kemudian akan digunakan untuk memperoleh *stopword list* dengan mengambil kumpulan *term* dengan nilai bobot terendah sebanyak X *term*. X merupakan parameter yang berarti nilainya dapat diubah untuk keperluan pengujian sehingga didapatkan nilai yang menghasilkan nilai hasil evaluasi sistem terbaik.

2.2.8 Evaluasi

Pada penelitian ini, digunakan *confusion matrix* jenis *multiclass* untuk melakukan evaluasi. *Confusion matrix* merupakan sebuah alat yang digunakan untuk menganalisis seberapa baik *classifier* dapat mengenali *tuple* dari berbagai kelas berbeda (Han, et al., 2012). Terdapat 4 istilah yang menggambarkan hasil klasifikasi menggunakan *Confusion Matrix* yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Istilah-istilah tersebut menyatakan jumlah data uji yang benar diklasifikasikan dan jumlah data uji yang salah diklasifikasikan. Tabel 2.3 merupakan contoh *Confusion Matrix* dengan jenis klasifikasi *multiclass*.

Tabel 2.3 Confusion Matrix

Confusion Matrix		Prediksi		
		A	B	C
Aktual	A	TP _A	E _{AB}	E _{AC}
	B	E _{BA}	TP _B	E _{BC}
	C	E _{CA}	E _{CB}	TP _C

Pada *confusion matrix*, TP (*True Positive*) merupakan data positif (relevan) yang terdeteksi benar sebagai data positif, sedangkan TN (*True Negative*) yang merupakan data negatif (tidak relevan) yang terdeteksi benar sebagai data negatif. FP (*False Positive*) merupakan data negatif (tidak relevan) namun terdeteksi sebagai data positif, dan FN (*False Negative*) yang merupakan data positif (relevan) yang terdeteksi sebagai data negatif. Persamaan 2.5, Persamaan 2.6 dan Persamaan 2.7 berturut-turut merupakan rumus yang digunakan untuk menghitung FP, FN dan TN pada *confusion matrix* jenis *multiclass*.

$$FP = \sum_{i=1}^l E_{li} - TP_i \quad (2.5)$$

Berdasarkan nilai TP (*True Positive*), TN (*True Negative*), FP (*False Positive*) dan FN (*False Negative*) dapat diperoleh hasil evaluasi berupa:

a. Accuracy yang merupakan persentase jumlah data yang diklasifikasikan dengan benar. Nilai Accuracy dapat diperoleh dengan Persamaan 2.8.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.8)$$

b. *Precision* merupakan tingkat ketepatan antara informasi yang diminta oleh *user* dengan jawaban yang diberikan oleh sistem. Nilai *Precision* dapat diperoleh dengan Persamaan 2.9.

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

c. *Recall* merupakan tingkat keberhasilan sistem dalam mengklasifikasikan data positif (relevan). Nilai *Recall* dapat diperoleh dengan Persamaan 2.10.

$$Recall = \frac{TP}{TP + FN} \quad (2.10)$$

d. *F-measure* merupakan parameter pengukuran yang digunakan untuk mengukur bobot rata-rata dari *precision* dan *recall*. Nilai *f-measure* dapat diperoleh dengan Persamaan 2.11.

$$F - \text{measure} = 2 \times \frac{(\text{precision} \times \text{recall})}{(\text{precision} + \text{recall})} \quad (2.11)$$

BAB 3 METODOLOGI PENELITIAN

Pada metodologi penelitian ini dijelaskan tahapan yang dilakukan peneliti dalam membuat sistem analisis sentimen untuk topik permasalahan kebijakan pemindahan Ibukota negara Indonesia.

3.1 Tipe Penelitian

Penelitian analisis sentimen yang dilakukan bertipe non-implementatif.

Penelitian tipe implementatif yaitu penelitian yang fokus kepada pengamatan terhadap situasi ataupun fenomena yang terjadi di masyarakat. Selain itu, dalam penelitian ini juga dilakukan analisis terhadap suatu fenomena yang terjadi untuk diambil suatu kesimpulan, dalam hal ini terkait sentimen dari warganet media sosial Twitter terkait kebijakan pemindahan Ibukota negara Indonesia.

Pendekatan penelitian yang digunakan yaitu pendekatan analitik. Pendekatan ini dilakukan dengan menganalisis hubungan antara objek-objek penelitian dengan kasus yang diteliti. Hasil akhir yang menjadi keluaran dari penelitian ini didasarkan pada rumusan masalah yang terdapat pada Bab Pendahuluan.

3.2 Strategi Penelitian

Strategi penelitian yang diterapkan dalam penelitian analisis sentimen ini menggunakan strategi yang bersifat kualitatif, yaitu strategi penelitian studi kasus. Strategi ini menekankan pada objek penelitian. Objek penelitian akan diteliti dan dipahami sebagai suatu kasus atau permasalahan. Fokus dalam penelitian ini yaitu analisis sentimen terkait kebijakan pemindahan Ibukota Republik Indonesia.

3.3 Peralatan Pendukung

Dalam melakukan penelitian ini, diperlukan peralatan pendukung berupa perangkat keras (*hardware*) dan perangkat lunak (*software*). Terdapat beberapa standar yang ditetapkan agar penelitian dapat dijalankan dengan baik.

3.3.1 Hardware

Hardware yang digunakan pada pembuatan sistem analisis sentimen ini menggunakan laptop HP ProBook 440 G0 dengan spesifikasi:

- Processor Intel® Core™ i5-3340M CPU @ 2.70GHz

- RAM 4 GB

- Sistem operasi Windows 10

- Harddisk 500 GB

3.3.2 Software

Software yang digunakan dalam pembuatan aplikasi analisis sentimen ini adalah Jupyter Notebook yang dibuka menggunakan Google Chrome untuk mengolah data dalam Bahasa Python. *Library* yang digunakan dalam system

antara lain, *math*, *nltk*, *numpy*, *collections*, *re*, *random* dan *operator*. Selain itu, Notepad++ juga digunakan untuk menyimpan data yang telah didapatkan.

3.4 Lokasi Penelitian

Penelitian analisis sentimen terkait kebijakan pemindahan Ibukota Indonesia dilaksanakan di Laboratorium Komputasi Cerdas Fakultas Ilmu Komputer Universitas Brawijaya. Penelitian yang dilakukan di laboratorium tersebut bertujuan untuk menerapkan algoritme *term-based random sampling* dan metode Naïve Bayes untuk keperluan analisis sentimen terkait kebijakan pemindahan Ibukota Indonesia.

3.5 Teknik Pengumpulan Data

Metode pengumpulan yang digunakan yaitu metode *non-participant observation*. Metode ini merupakan metode observasi tanpa melibatkan peneliti secara langsung dalam proses maupun kegiatan yang sedang diteliti atau diamati. Objek dalam penelitian ini adalah data yang diambil dari twit pengguna media sosial Twitter yang mengandung kata kunci terkait kebijakan pemindahan Ibukota Republik Indonesia. Kumpulan data twit didapatkan dengan menggunakan fitur *advanced search* dari media sosial Twitter dan menggunakan *library* Tweepy dari bahasa pemrograman Python 3. Total terdapat 450 data yang digunakan dalam penelitian ini. Dari data tersebut, 300 data digunakan sebagai data latih, 50 data untuk data uji dan 100 data untuk keperluan sampel analisis sentimen. Dari data-data tersebut terdapat data latih dan data uji yang merupakan data berlabel. Untuk melabeli data-data tersebut, penulis dibantu oleh 2 orang mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya yang bernama Salsabila Insani dan Vinesia Yolanda agar label pada data tersebut tidak bersifat subjektif.

3.6 Data Penelitian

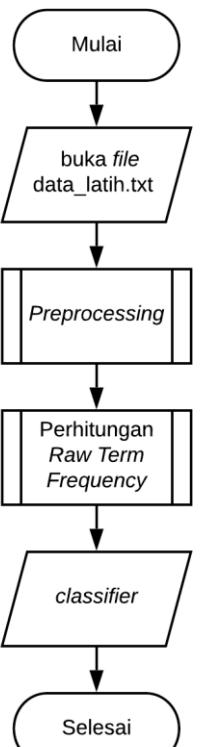
Data penelitian yang digunakan yaitu data twit yang mengandung kata kunci terkait pemindahan Ibukota, yaitu kata kunci “pemindahan Ibukota”, “Ibukota pindah” dan “ibukota dipindahkan”. Terdapat total 450 data yang digunakan dalam penelitian ini. 300 dari 450 data digunakan sebagai data latih dengan persebaran 150 data latih positif dan 150 data latih negatif. Selain itu, 100 data tanpa label digunakan untuk keperluan analisis sentimen untuk menarik kesimpulan terkait pro dan kontra warganet Twitter terkait kebijakan pemindahan Ibukota. Sisa data sebanyak 50 data berlabel digunakan untuk keperluan pengujian.

3.7 Teknik Analisis Data

Teknik analisis data dilakukan dengan menggunakan mekanisme pengujian terhadap hasil klasifikasi sistem terhadap data uji penelitian. Pengujian dilakukan terhadap metode yang digunakan oleh sistem. Parameter pengujian yang dihitung antara lain akurasi, *precision*, *recall*, dan *f-measure*.

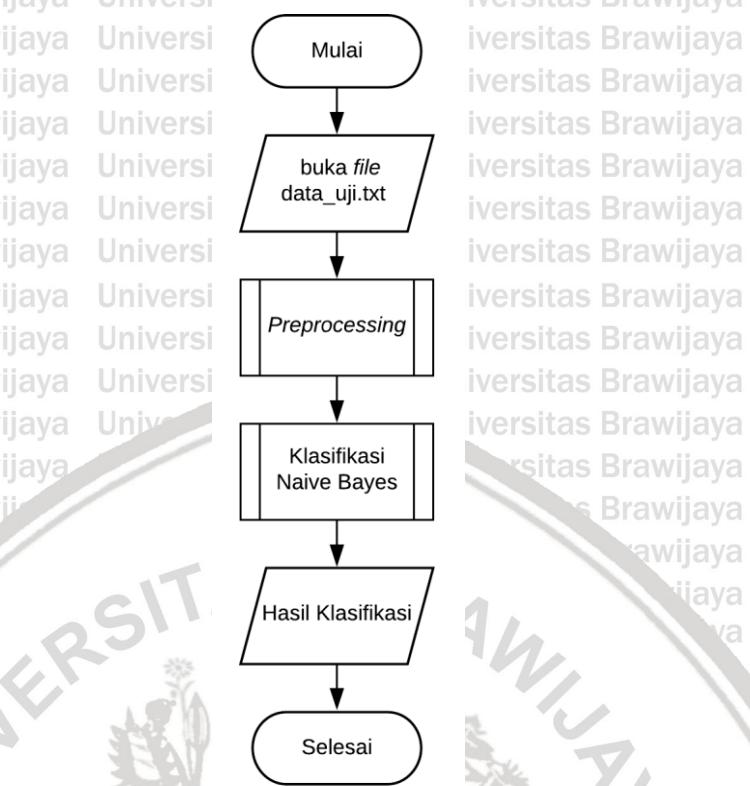
3.8 Perancangan Algoritme

Pada tahap ini dilakukan penjabaran langkah-langkah implementasi metode secara garis besar dari awal hingga terbentuk sistem analisis sentimen. Pada penelitian ini, proses pembuatan sistem diawali dengan proses *preprocessing* data latih, kemudian dilakukan proses perhitungan *raw term frequency* dari *term* data latih. Setelah nilai *raw term frequency* dari tiap *term* data latih selesai dihitung, kemudian dilakukan implementasi metode klasifikasi Naïve Bayes untuk mengklasifikasikan data uji ke dalam salah satu kelas sentimen. Pada Gambar 3.1 ditunjukkan diagram alir proses pelatihan sistem.



Gambar 3.1 Diagram Alir Proses Pelatihan Sistem

Setelah melakukan proses pelatihan, selanjutnya sistem melakukan proses pengujian. Proses pengujian dilakukan dengan melakukan klasifikasi data uji dengan menggunakan metode klasifikasi Naïve Bayes berdasarkan classifier yang didapatkan dari hasil pelatihan. Hasil akhir dari proses pengujian yaitu berupa hasil klasifikasi dari kumpulan data uji yang ada. Pada Gambar 3.2 ditunjukkan diagram alir dari proses pengujian sistem.



Gambar 3.2 Diagram Alir Proses Pengujian Sistem

3.9 Teknik Penerapan Metode

Dalam penerapan metode dilakukan tahap implementasi. Implementasi dilakukan berdasarkan perancangan sistem yang telah dilakukan sebelumnya. Implementasi sistem meliputi:

1. Mengumpulkan data yang telah didapat dari proses *data scraping*. Setelah itu, data yang telah didapatkan dibagi menjadi data latih, data uji dan data untuk analisis sentimen.
2. Melakukan *preprocessing* data, termasuk implementasi algoritme *term-based random sampling* untuk memperoleh *stoplist* yang digunakan pada proses *filtering*.
3. Melakukan perhitungan *raw term frequency* dari tiap *term*.
4. Menerapkan metode Naïve Bayes untuk klasifikasi data uji.

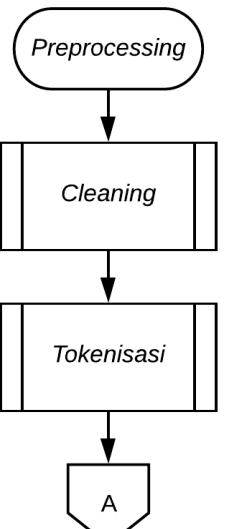
BAB 4 PERANCANGAN

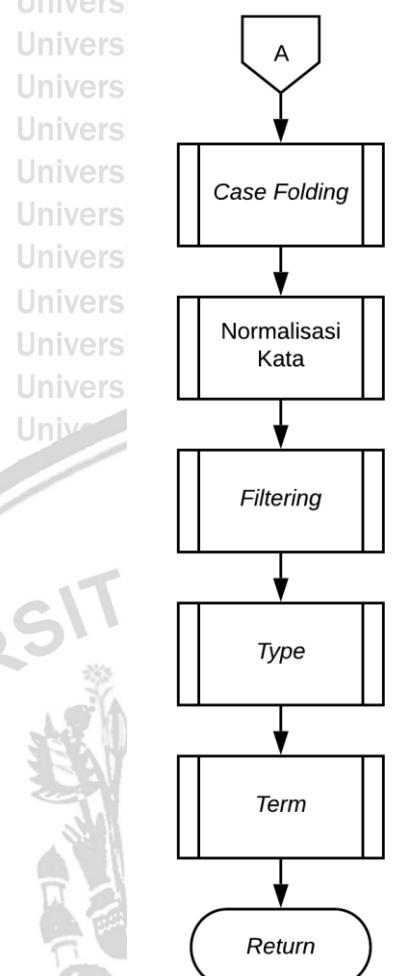
4.1 Diagram Alir Sistem (*Flowchart*)

Diagram alir sistem (*flowchart*) merupakan uraian dari tahapan-tahapan yang dilakukan untuk menyelesaikan masalah. Klasifikasi teks pada sistem analisis sentimen ini dimulai dengan melakukan masukkan data latih dan data uji yang digunakan dalam penelitian, kemudian dilakukan tahapan *preprocessing* untuk mempersiapkan data agar data dapat diolah untuk proses selanjutnya. Dalam tahapan *preprocessing* dilakukan mekanisme pembuatan *stopword list* secara otomatis dari data latih dengan menggunakan metode *term-based random sampling*. Setelah melakukan *preprocessing* tahapan selanjutnya adalah Perhitungan *raw term frequency*. Nilai *raw term frequency* didapatkan dengan menjumlahkan nilai *term frequency* dari suatu term pada semua dokumen. Setelah nilai *raw term frequency* didapatkan, kemudian dilakukan tahapan terakhir yaitu klasifikasi data uji dengan menggunakan metode klasifikasi Naïve Bayes. Pada subbab dibawah ini akan ditunjukkan diagram alir yang menjelaskan secara rinci terkait tahapan-tahapan yang dilakukan dalam penelitian ini.

4.1.1 Diagram Alir Preprocessing

Proses mengolah data mentah menjadi data yang siap diolah untuk proses selanjutnya merupakan pengertian dari *preprocessing* data. Terdapat beberapa tahapan yang akan dilakukan dalam proses *preprocessing* yakni: *case folding*, *tokenizing*, *filtering*, dan tahapan yang terakhir adalah *stemming*. Gambar 4.1 merupakan diagram alir dari proses *preprocessing*.



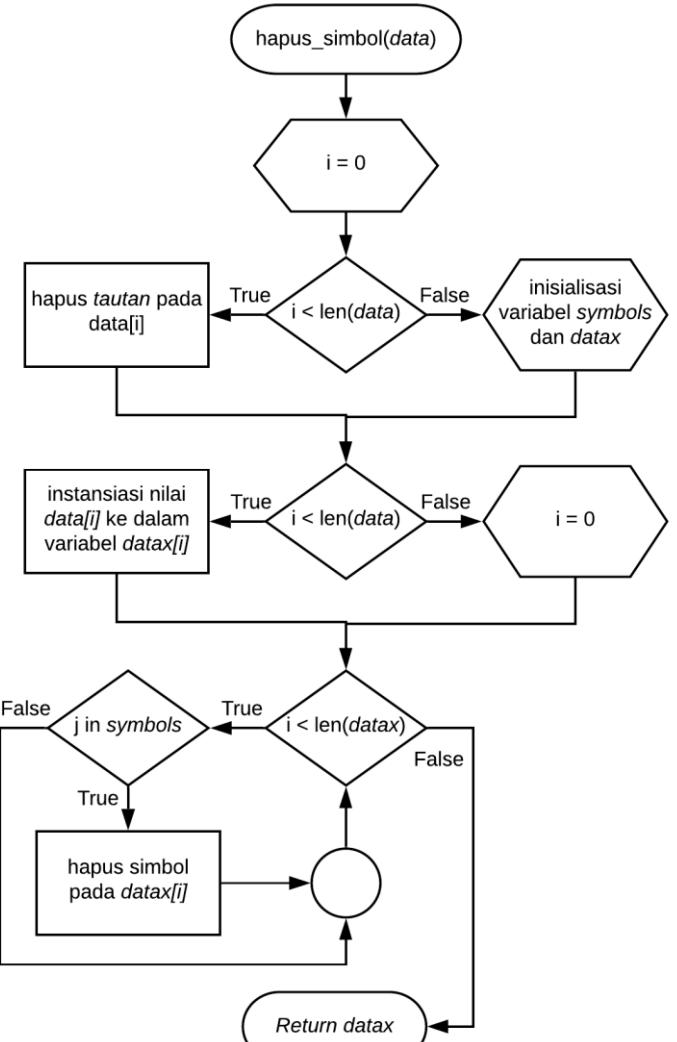


Gambar 4.1 Diagram Alir Preprocessing

Perlu diketahui bahwa proses *tokenisasi*, *case folding*, *filtering*, *stemming* dan *type* dilakukan oleh sistem dengan menggunakan bantuan *library* maupun fungsi pada bahasa pemrograman Python. Oleh karena itu, kesemua proses tersebut tidak akan dijelaskan secara lebih lanjut dalam bentuk diagram alir. Proses *preprocessing* yang dilakukan oleh sistem tanpa bantuan *library* maupun fungsi pada bahasa pemrograman Python yaitu proses normalisasi kata. Rincian proses normalisasi kata dijelaskan pada subbab berikutnya.

4.1.1.1 Diagram Alir Cleaning

Pada tahap *cleaning* dilakukan proses penghapusan *link*, karakter tanda baca atau simbol, dan juga angka. Proses penghapusan tersebut dilakukan untuk menghilangkan entitas yang dianggap tidak penting dan dianggap mengganggu proses klasifikasi data untuk keperluan analisis sentimen. Gambar 4.2 menunjukkan diagram alir tahap *cleaning*. Fungsi *len()* pada diagram alir merupakan fungsi untuk mendapatkan nilai dari panjang *array*.



Gambar 4.2 Diagram Alir Cleaning

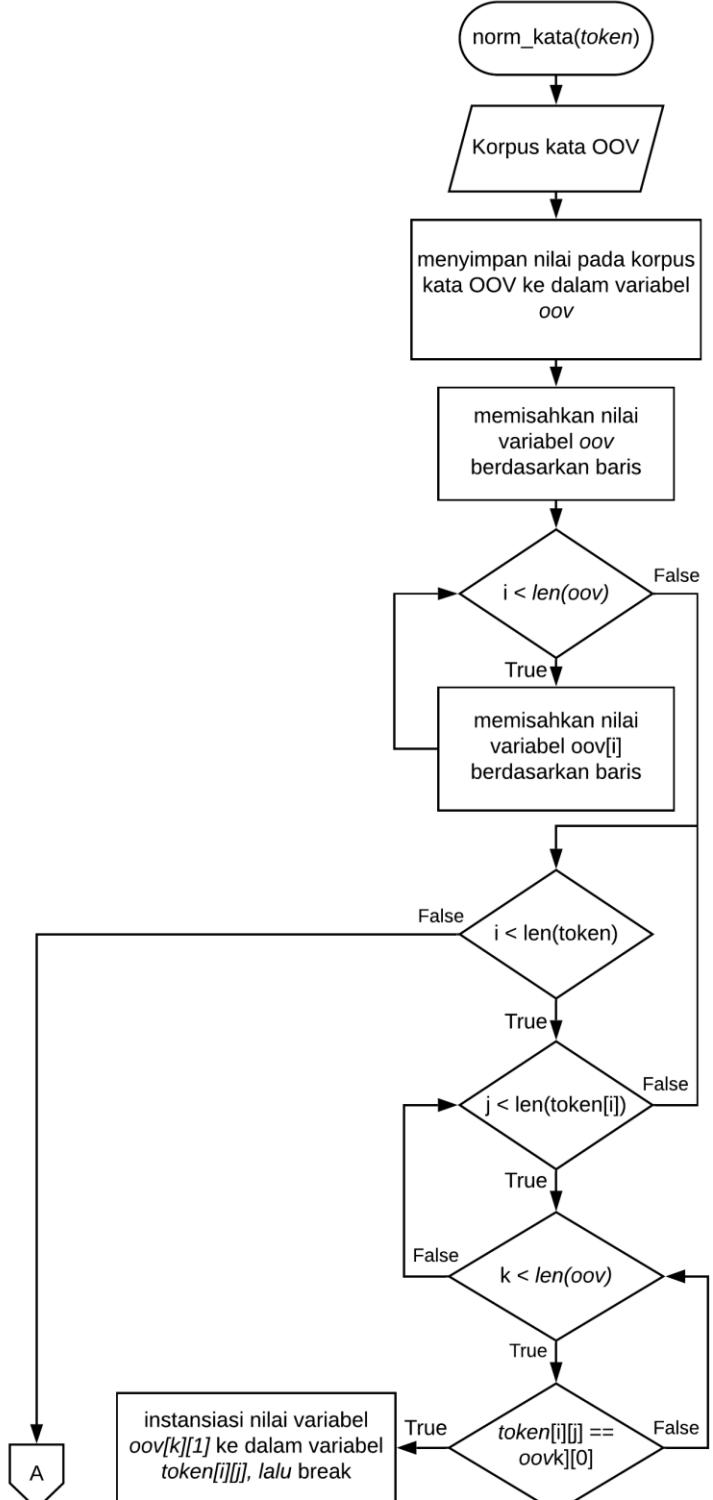
4.1.1.2 Diagram Alir Normalisasi kata

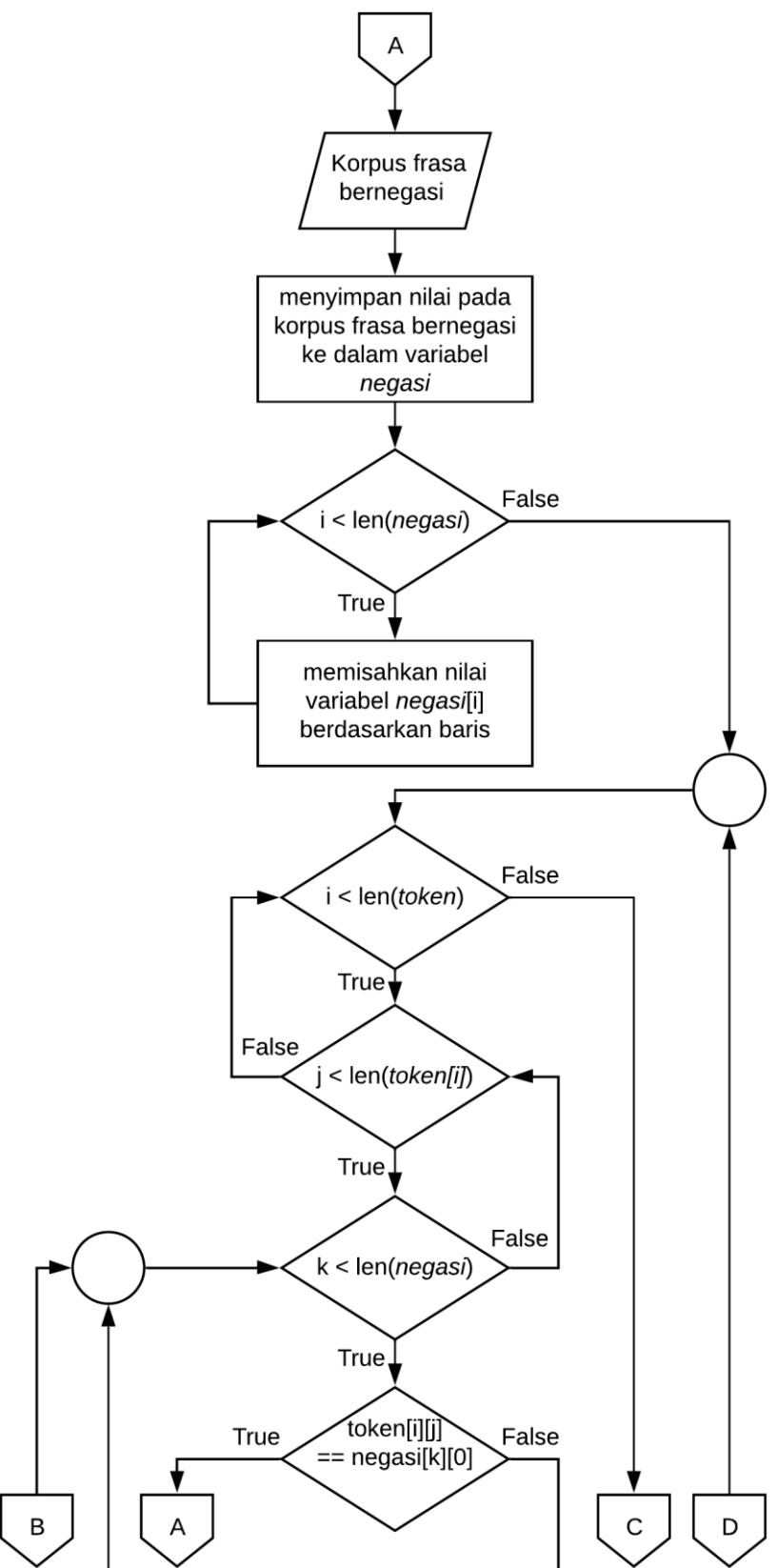
Proses Normalisasi kata merupakan proses untuk mengubah kata-kata ambigu atau tidak baku menjadi kata baku. Hal ini perlu dilakukan karena pada media sosial Twitter pengguna hanya dapat membuat tweet hanya dalam 280 karakter saja, sehingga seringkali pengguna menggunakan kata singkatan untuk mempersingkat kata. Namun, hal ini justru mempersulit proses analisis sentimen karena rawan terjadi *misjudge* oleh sistem terhadap makna kata tersebut, sehingga perlu dilakukan normalisasi kata agar sistem tetap dapat memahami makna dari suatu kata. Proses normalisasi kata dibagi menjadi 2 bagian, yaitu proses perubahan kata ambigu dan proses perubahan frasa bernegasi atau frasa yang mengandung kata "tidak". Proses perubahan kata ambigu yaitu mengubah kata-kata yang tidak baku dan kata yang disingkat menjadi ke bentuk bakuanya, sedangkan proses perubahan frasa bernegasi yaitu mengubah frasa yang mengandung kata negasi ke kata sinonimnya yang tidak mengandung kata "tidak" atau kata negasi lainnya. Proses perubahan frasa bernegasi dilakukan karena kata

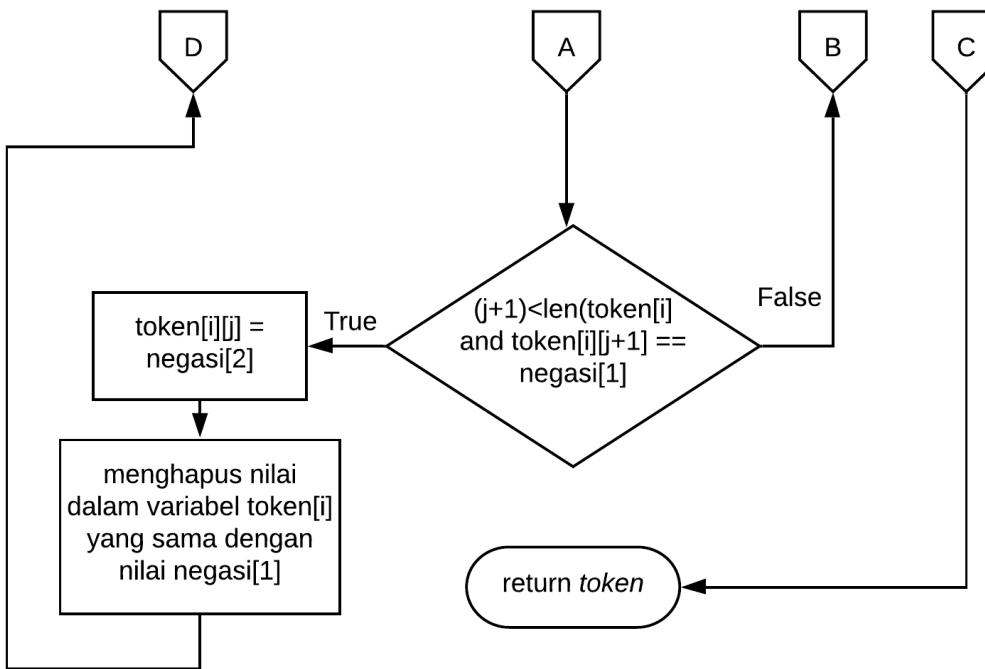


negasi seperti contohnya, kata “tidak” dan “bukan” yang akan dihapus pada proses *Filtering*, sehingga perubahan frasa bernegasi harus dilakukan untuk menghindari terjadinya perubahan makna dokumen pada proses *preprocessing*.

Gambar 4.3 menunjukkan diagram alir dari proses normalisasi kata yang dilakukan oleh sistem. Istilah OOV pada diagram alir merupakan singkatan dari *Out-of-Vocabulary*, yang berarti kata-kata yang bukan kata baku.



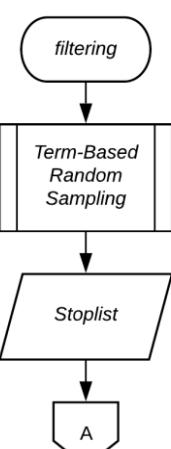


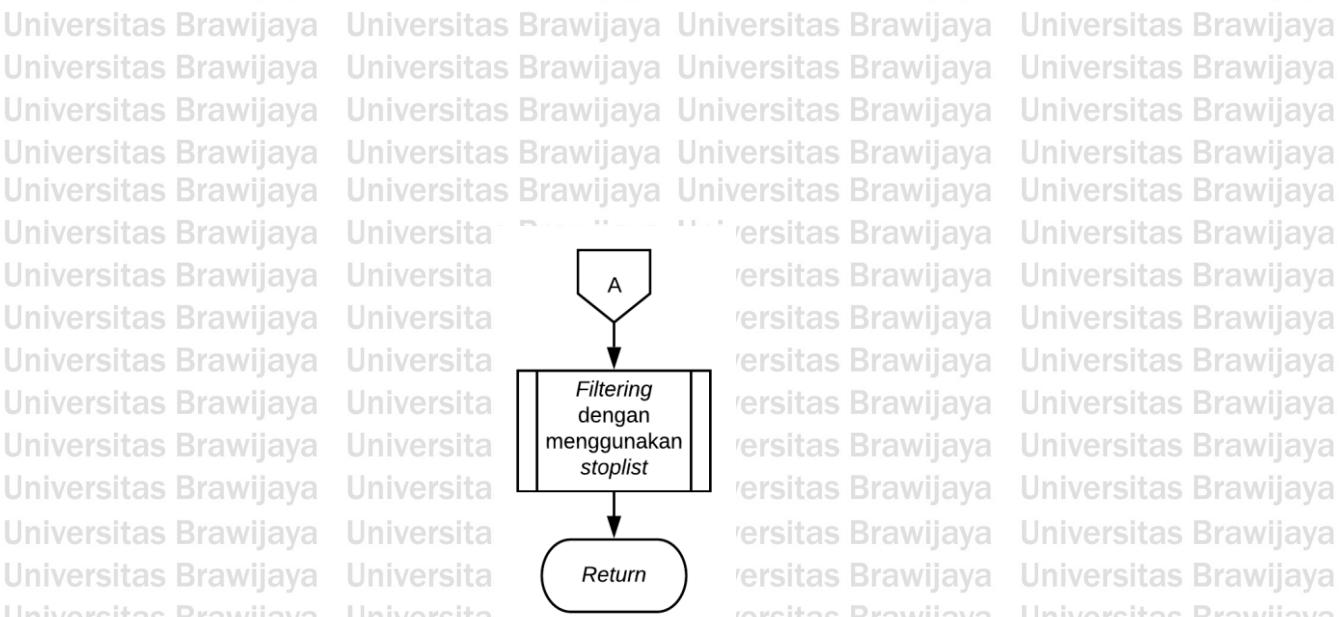


Gambar 4.3 Diagram Alir Pengubahan Normalisasi kata

4.1.1.3 Diagram Alir *Filtering*

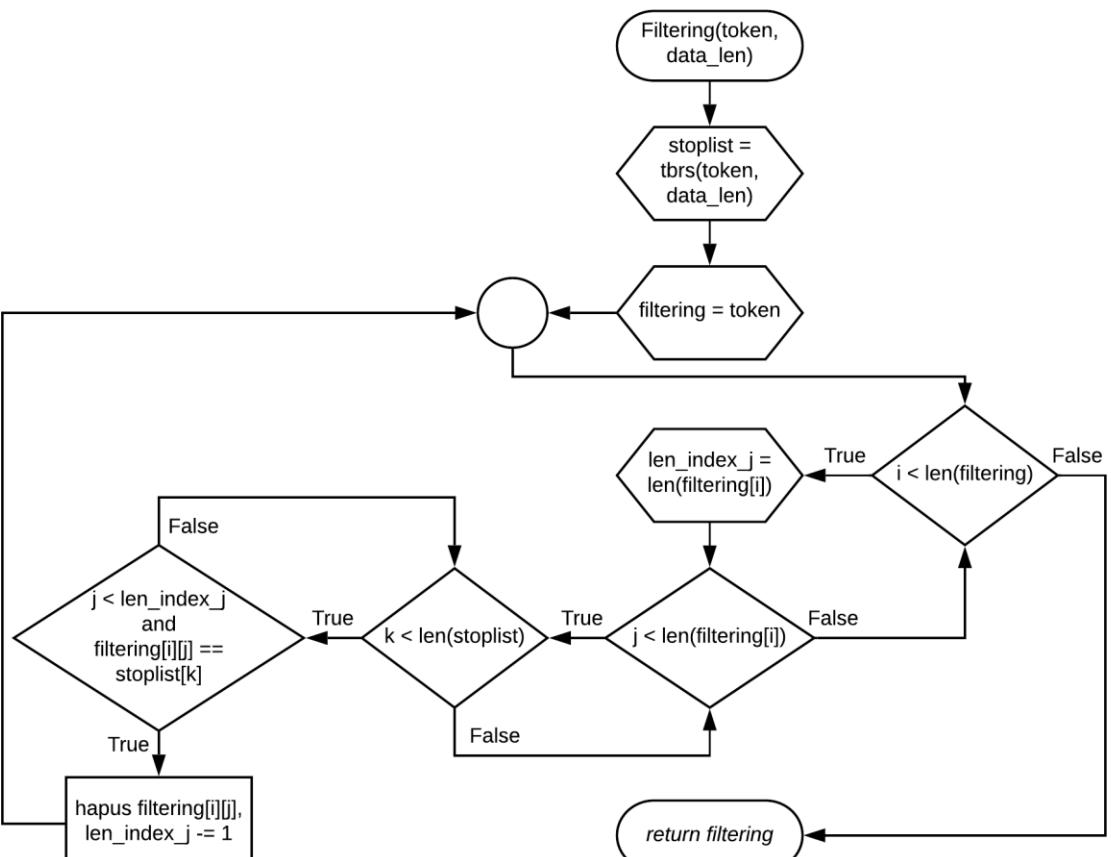
Pada tahap ini dilakukan proses penghapusan kata yang dianggap tidak penting. Untuk melakukan *Filtering* diperlukan daftar kata yang perlu dihapus dalam dokumen yang disebut dengan *stoplist*. Pada penelitian ini, *stoplist* dibuat secara otomatis berdasarkan data latih yang dimasukkan ke dalam sistem. Pembuatan *stoplist* dilakukan dengan menggunakan metode *term-based random sampling*. Dalam implementasinya, metode *term-based random sampling* akan menggunakan nilai Kullback-Leibler sebagai acuan pembobotan untuk melakukan pemeringkatan daftar *term*. *Term* sebanyak x yang memiliki nilai bobot Kullback-Leibler terkecil akan dijadikan sebagai *stoplist*. Gambar 4.4 menunjukkan diagram alir dari tahap *Filtering*.





Gambar 4.4 Diagram Alir *Filtering*

Setelah dilakukan proses implementasi metode *term-based random sampling* didapatkan *stoplist* yang akan digunakan dalam proses *Filtering*. Pada proses *Filtering* akan dilakukan proses penghapusan *term-term* dalam dokumen yang termasuk ke dalam daftar *stoplist*. Gambar 4.5 menunjukkan diagram alir proses *Filtering* menggunakan *stoplist*.

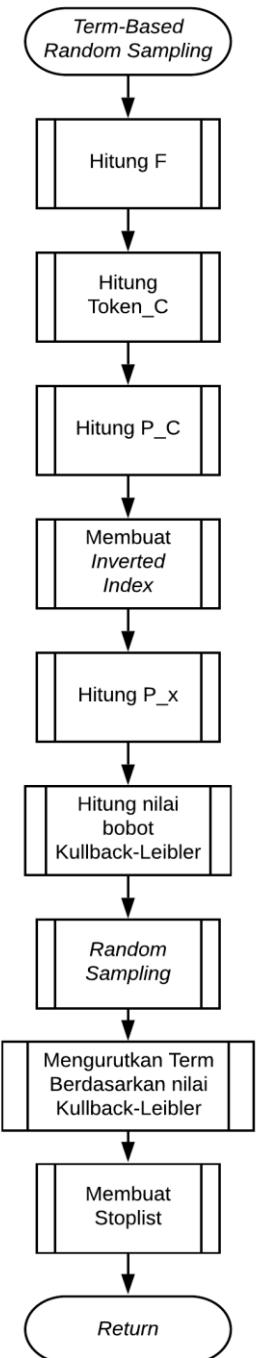


Gambar 4.5 Diagram Alir *Filtering dengan Stoplist*



4.1.2 Term-Based Random Sampling

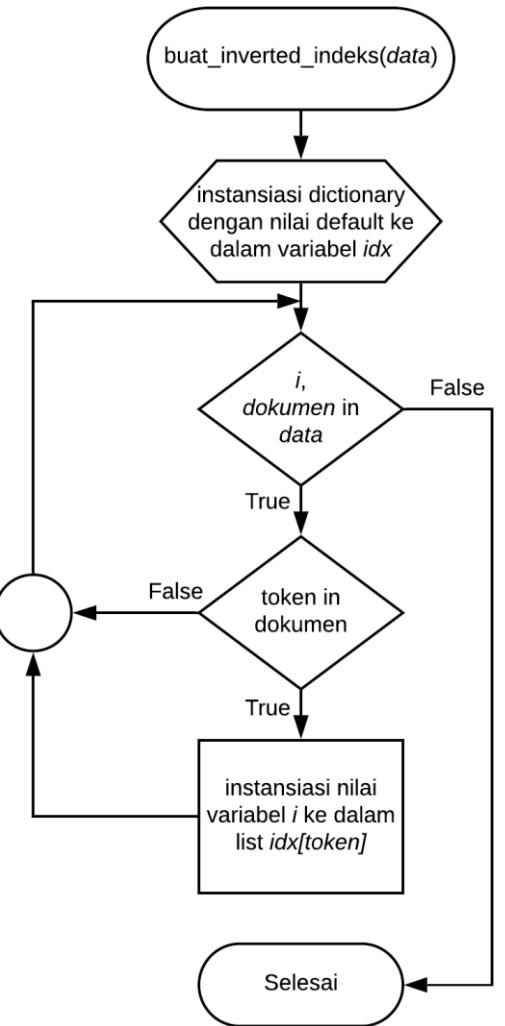
Algoritme *term-based random sampling* diimplementasikan untuk membuat *stoplist* yang kemudian digunakan pada proses *filtering*. *Stoplist* dibuat berdasarkan *term* yang didapatkan dari data latih. Gambar 4.6 menunjukkan Diagram Alir *term-based random sampling*.



Gambar 4.6 Diagram Alir *Term-Based Random Sampling*

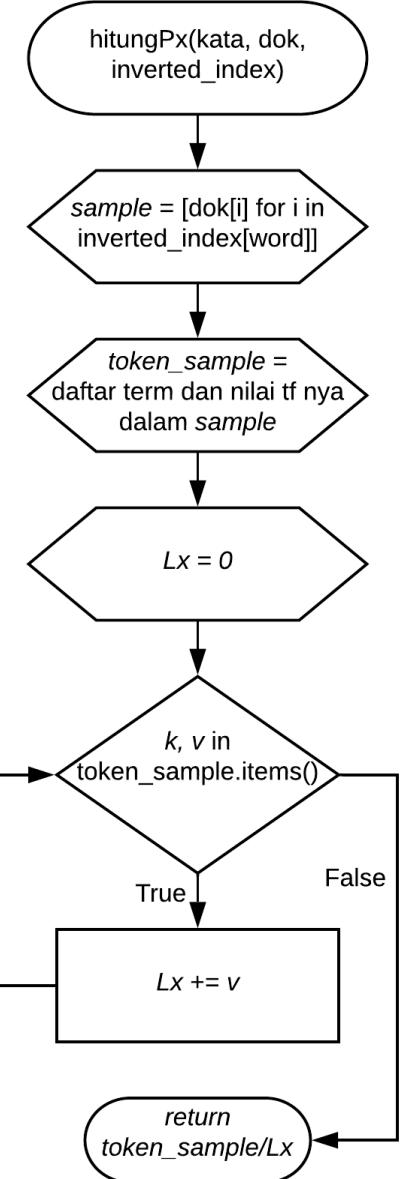


Sebelum menghitung P_x dibuat variabel yang menyimpan nilai dari *inverted index*. Pembentukan *inverted index* bertujuan untuk mengurutkan *term* berdasarkan kemunculannya dalam koleksi dokumen. Selain itu, dalam variabel yang menyimpan nilai *inverted index* juga akan disimpan *index* kemunculan tiap *term* pada koleksi dokumen. Gambar 4.7 menampilkan diagram alir untuk pembuatan *inverted index*.



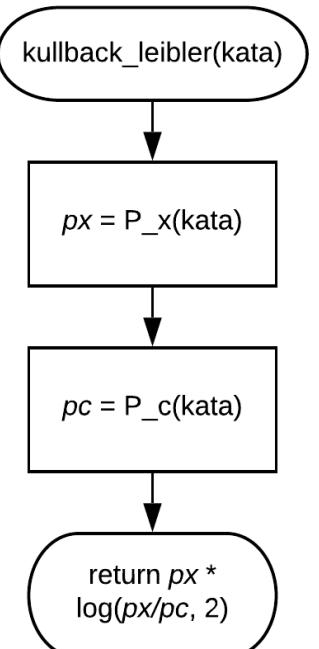
Gambar 4.7 Diagram Alir Membuat *Inverted Index*

Untuk menghitung nilai bobot Kullback-Leibler, perlu diketahui *token* dari data latih dan indeks kemunculannya dalam koleksi dokumen data latih. Oleh karena itu, *Inverted Index* dibuat untuk menyimpan nilai-nilai tersebut. Variabel yang menyimpan nilai *inverted index* akan digunakan pada proses perhitungan bobot Kullback-Leibler.



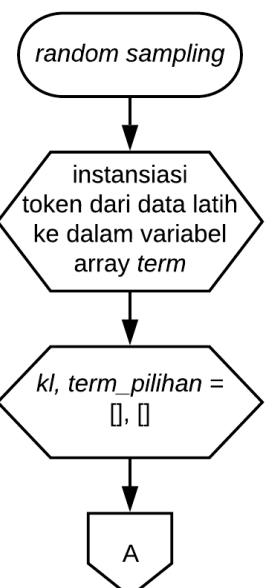
Gambar 4.8 Diagram Alir Hitung Px

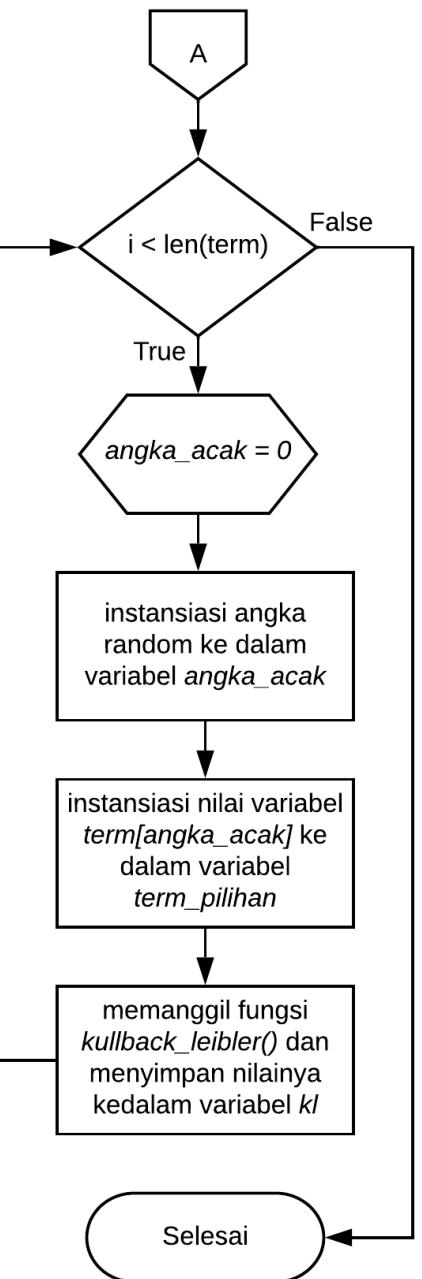
Untuk menghitung P_x dibutuhkan nilai dari Tfx dan Lx . Tfx adalah *term frequency* dari suatu *term* pada sampel dokumen, sedangkan Lx adalah jumlah total *term frequency* dari seluruh *term* pada sampel dokumen. P_x dihitung dengan cara membagi Tfx dengan Lx . Gambar 4.8 menunjukkan diagram alir untuk menghitung P_x .



Gambar 4.9 Diagram Alir Hitung Nilai Bobot Kullback-Leibler

Setelah dibuat fungsi untuk menghitung P_x dan P_c , kemudian dilakukan perhitungan nilai bobot Kullback-Leibler dari tiap term. Pada fungsi, perhitungan nilai bobot Kullback-Leibler dilakukan dengan memanggil fungsi P_x dan fungsi P_c . Setelah didapatkan nilai P_x dan P_c kemudian fungsi akan mengembalikan nilai bobot Kullback-Leibler yang dihitung berdasarkan Persamaan 2.4. Gambar 4.9 menunjukkan diagram alir untuk menghitung nilai bobot Kullback-Leibler.

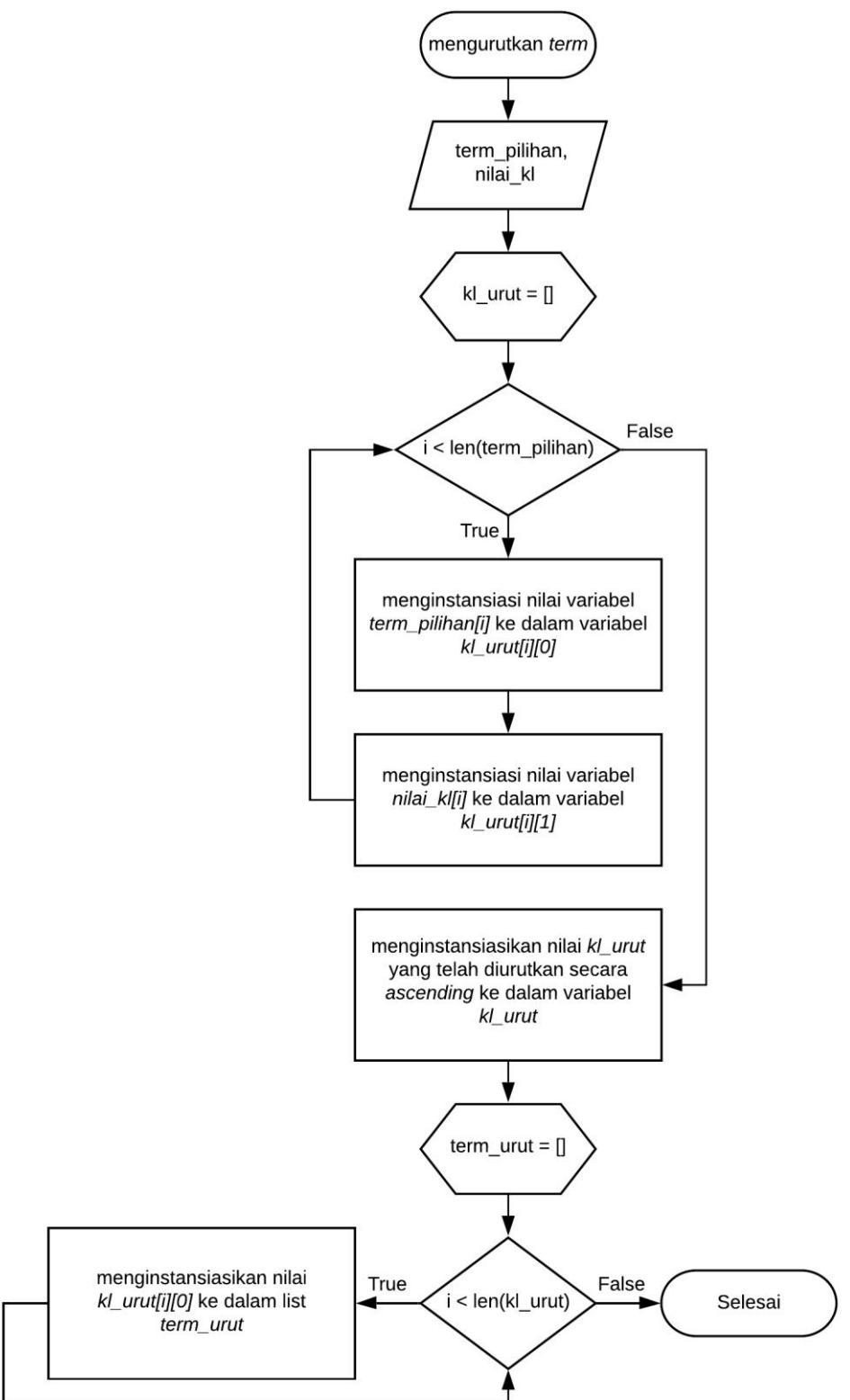




Gambar 4.10 Diagram Alir Random Sampling

Setelah dilakukan perhitungan nilai bobot Kullback-Leibler dari tiap *term*, kemudian dilakukan proses *Random Sampling*. Proses *Random Sampling* dilakukan untuk memilih *term* sebanyak *X* secara acak dari daftar term yang telah didapat dari data latih. *X* merupakan parameter yang nilainya dapat diubah-ubah.

Pada Gambar 4.10 ditunjukkan diagram alir dari proses *Random Sampling*.

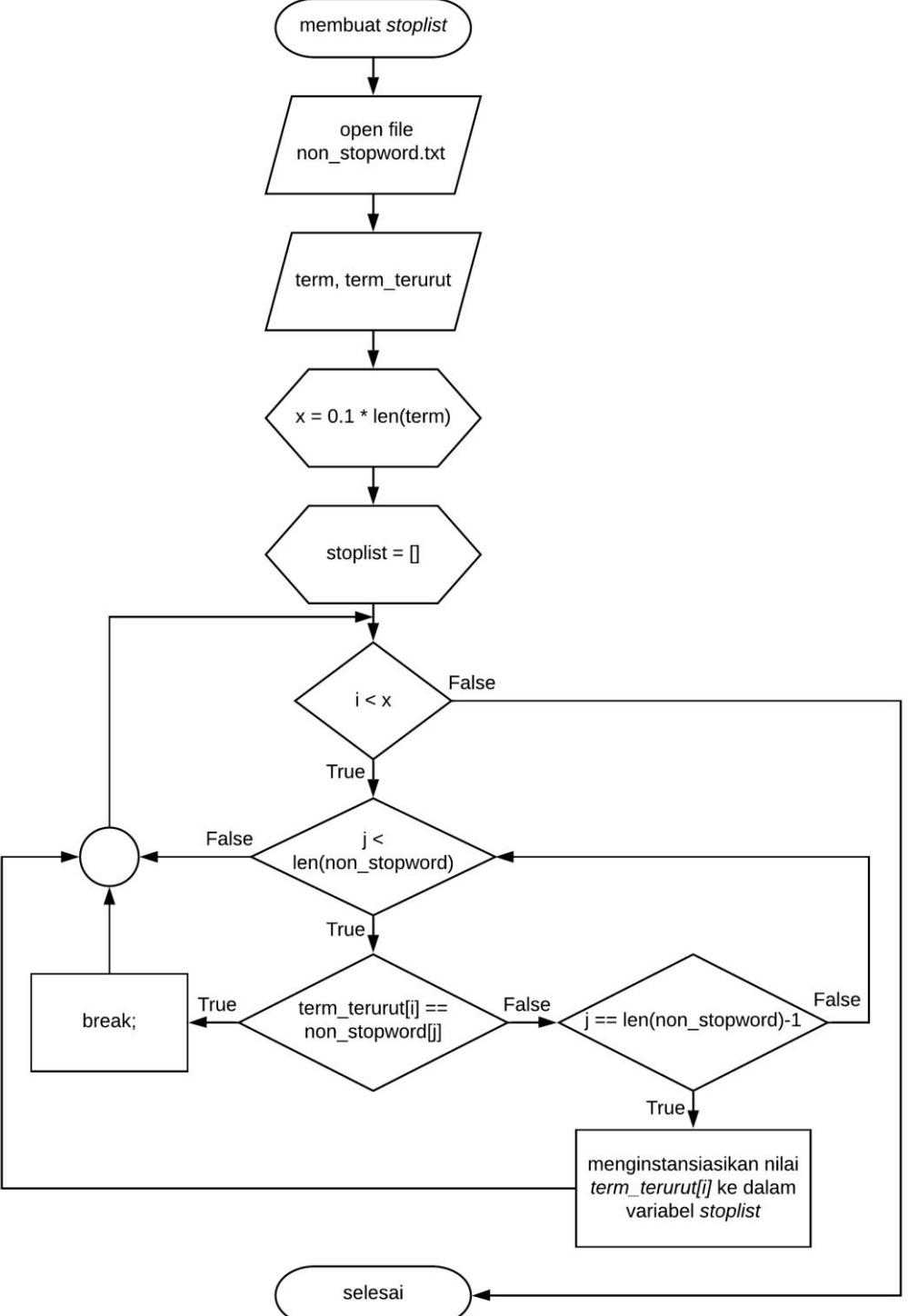


Gambar 4.11 Diagram Alir Mengurutkan Term Berdasarkan Nilai Kullback-Leibler

Setelah dihitung nilai bobot Kullback-Leibler, kemudian dilakukan pengurutan term berdasarkan nilai bobot Kullback-Leiblernya. Pengurutan ini dilakukan secara



ascending dan bertujuan untuk mempermudah proses pemilihan stoplist. Gambar 4.11 menunjukkan diagram alir dari proses mengurutkan *term* berdasarkan nilai Kullback-Leibler.

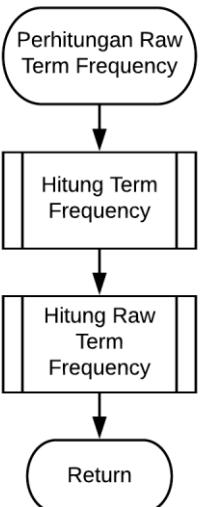


Gambar 4.12 Diagram Alir Membuat Stoplist

Setelah daftar *term* diurutkan berdasarkan nilai bobot Kullback-Leiblernya, kemudian dilakukan pemilihan kata yang akan dijadikan sebagai *stopword*. Pemilihan ini didasarkan pada nilai bobot Kullback-Leibler. Kumpulan kata sebanyak X dengan nilai Kullback-Leibler terkecil akan dipilih untuk masuk ke dalam *stoplist*. Namun, sebelum itu perlu dilakukan pengecekan apakah kata tersebut termasuk dalam kata yang terdapat dalam *lexicon non stopword* atau tidak. Apabila kata tersebut termasuk dalam kata yang terdapat dalam *lexicon non stopword*, maka meskipun kata tersebut memiliki nilai bobot Kullback-Leibler yang kecil, namun kata tersebut tidak boleh dijadikan *stopword*. Pada Gambar 4.12 ditunjukkan diagram alir proses pembuatan *stoplist*.

4.1.3 Diagram Alir Perhitungan Raw Term Frequency

Setelah melewati tahap *preprocessing* teks, didapatkan daftar *term*. Setelah daftar *term* didapatkan, kemudian sistem melakukan proses selanjutnya yaitu melakukan perhitungan nilai *Raw term frequency* dari tiap *term*. Nilai *Raw term frequency* didapatkan dengan menghitung *tf* dari *term* pada semua dokumen kemudian dijumlahkan. Gambar 4.13 menunjukkan diagram alir *Raw term frequency*.

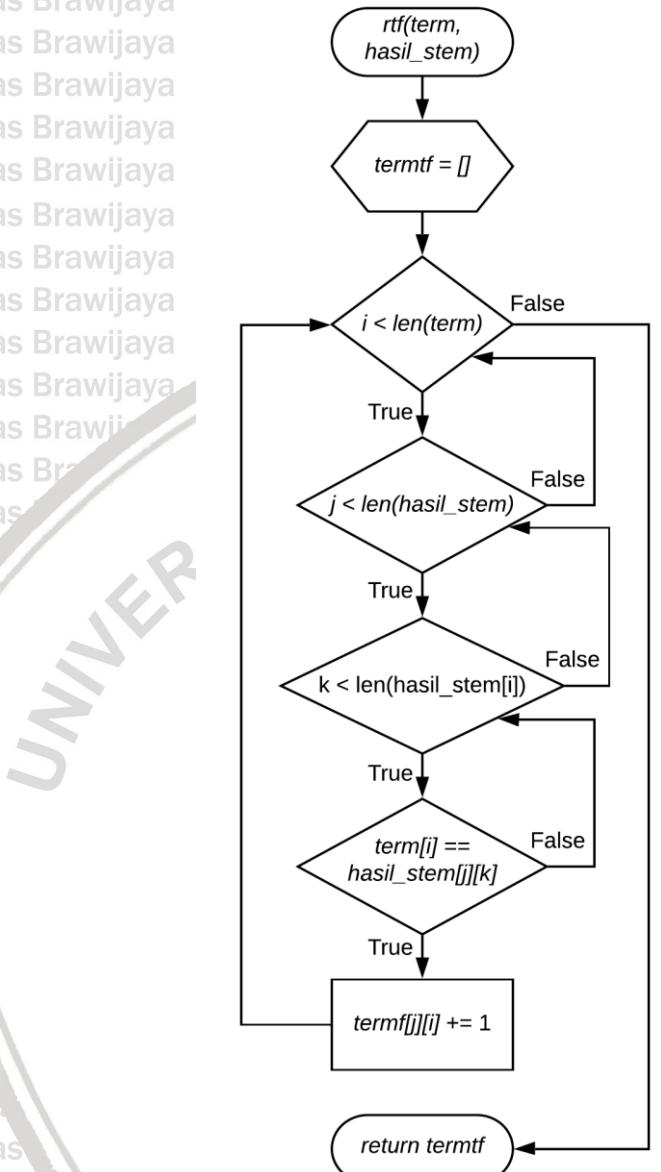


Gambar 4.13 Diagram Alir Perhitungan *Raw term frequency*

4.1.3.1 Diagram Alir Hitung Term Frequency

Dalam penelitian ini, analisis sentimen dilakukan dengan fitur *bag of words*. Perhitungan *Raw term frequency* dilakukan untuk membantu proses klasifikasi data teks. Untuk menghitung nilai *Raw term frequency*, sebelumnya perlu dilakukan perhitungan *Term Frequency* dari setiap *term* terhadap setiap dokumen.

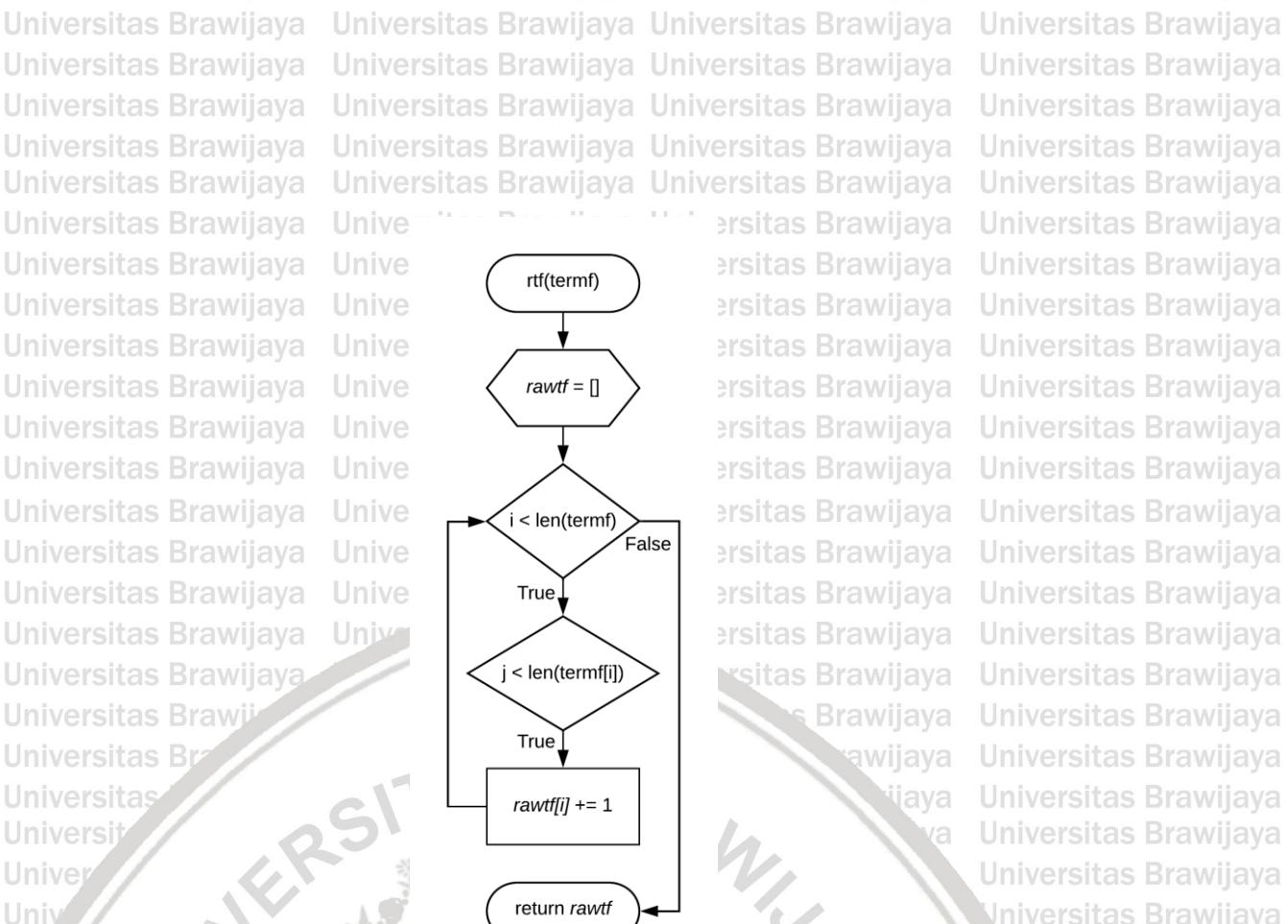
Gambar 4.14 merupakan diagram alir dari proses perhitungan *Term Frequency*.



Gambar 4.14 Diagram Alir Perhitungan *Term Frequency*

4.1.3.2 Diagram Alir Hitung *Raw Term Frequency*

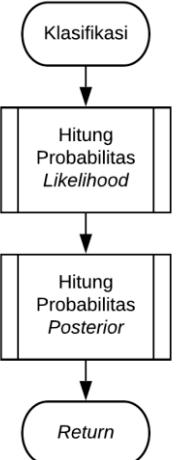
Setelah didapatkan nilai *term frequency*, kemudian dilakukan proses perhitungan *raw term frequency*. *Raw term frequency* dihitung dengan cara menjumlahkan semua nilai *term frequency* dari suatu *term* dari semua dokumen. Pada Gambar 4.15 ditunjukkan diagram alir perhitungan *raw term frequency*.



Gambar 4.15 Diagram Alir Perhitungan *Raw term frequency*

4.1.4 Diagram Alir Klasifikasi Naïve Bayes

Setelah dilakukan perhitungan *raw term frequency*, tahap selanjutnya adalah klasifikasi data teks dengan menggunakan metode klasifikasi Naïve Bayes. Klasifikasi Naïve Bayes dilakukan dengan cara menghitung probabilitas *likelihood* dan *prior*. Dalam penelitian ini, digunakan model *multinomial* untuk menghitung probabilitas *likelihood*. Gambar 4.16 menunjukkan diagram alir dari perhitungan probabilitas *likelihood*.

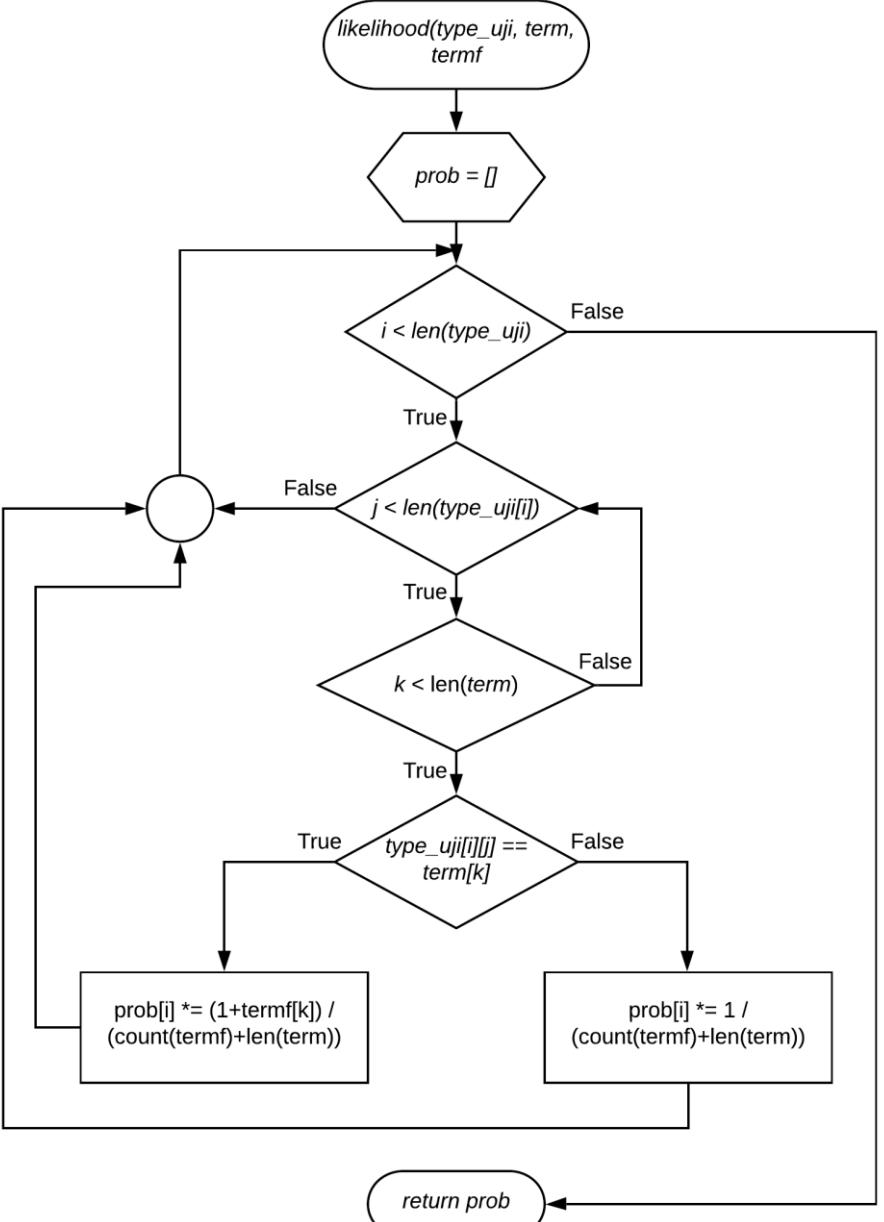


Gambar 4.16 Diagram Alir Klasifikasi Naïve Bayes



4.1.4.1 Diagram Alir Hitung *Likelihood*

Pada penelitian ini digunakan model *multinomial* untuk menghitung probabilitas *likelihood*, sehingga untuk dibutuhkan parameter berupa nilai hasil *preprocessing* hingga tahap *type* dari data uji, *term* dari data latih dan nilai *Term Frequency* dari tiap *term* yang dihasilkan dari data latih. Pada Gambar 4.17 ditunjukkan diagram alir dari perhitungan probabilitas *Likelihood* dari data uji.



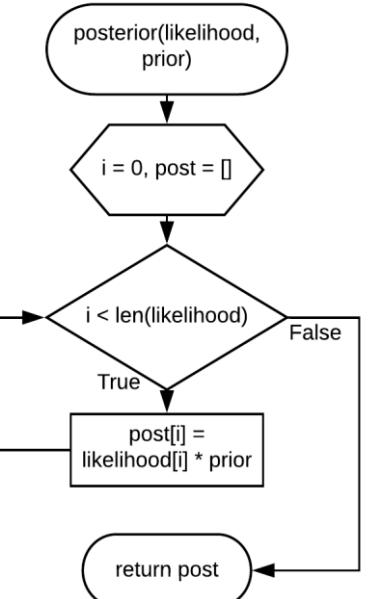
Gambar 4.17 Diagram Alir Hitung *Likelihood*

4.1.4.2 Diagram Alir Hitung Posterior

Setelah dilakukan perhitungan probabilitas *likelihood*, selanjutnya dilakukan perhitungan probabilitas *posterior*. Probabilitas *posterior* dapat dilakukan dengan cara mengalikan probabilitas *likelihood* dan probabilitas *prior*. Dalam penelitian



ini, perhitungan probabilitas *prior* tidak dijelaskan secara rinci, karena dalam kode program, perhitungan probabilitas *prior* dapat dilakukan dengan melakukan perhitungan sederhana. Pada Gambar 4.18 ditunjukkan diagram alir perhitungan probabilitas *Posterior*.



Gambar 4.18 Diagram Alir Hitung *Posterior*

4.2 Perhitungan Manual

Untuk melakukan perhitungan manual digunakan 4 data latih dan 2 data uji. 4 data latih terbagi menjadi 2 data latih berlabel positif dan 2 data latih berlabel negatif. Berikut data yang digunakan dalam perhitungan manual:

Data Latih:

D1: Saya sebagai warga negara yang mencintai indonesia, mengijinkan Ibukota dipindahkan ke pulau Kalimantan pak. Lebih cepat lebih baik! (positif)

D2 : Usahlah bersedih, semoga Ibukota gak pindah,dan batal dilantik,Aamiin.. (negatif)

D3 : Tanpa Payung Hukum, Rencana Jokowi Pindah Ibukota Cacat Prosedur (negatif)

D4 : Pendapat saya pribadi, Ibukota dipindahkan ada dampak positifny jd. Saya setuju! (positif)

Data Uji:

D1 : Sy pribadi stuju Ibukota dipindahkan, jakarta uda padet bgt. (positif)

D2 : pokonya ga setuju klo Ibukota dipindahin! Semoga aja gak jadi pindah. (negatif)

4.2.1 Data Scraping

Data Scraping merupakan proses pengambilan data untuk keperluan penelitian. Pada penelitian ini, proses *Data Scraping* dilakukan dengan cara mencari data/twit yang mengandung kata kunci yang terkait dengan topik permasalahan pemindahan Ibukota Indonesia, seperti "Ibukota pindah", "pemindahan Ibukota" dan lain-lain pada situs search.twitter.com dan juga melalui *library* Tweepy pada bahasa pemrograman Python.

4.2.2 Preprocessing

Preprocessing merupakan proses untuk mengekstraksi informasi dari data yang tidak terstruktur. Proses-proses dalam *Preprocessing* sebagai berikut:

- Cleaning

Merupakan proses penghapusan link, karakter tanda baca dan angka numerik yang terdapat dalam dokumen. Proses penghapusan ini bertujuan untuk menghindari kesalahan interpretasi kata karena adanya karakter-karakter yang tidak perlu. Tabel 4.1 menunjukkan contoh proses *Cleaning*.

Tabel 4.1 Contoh Cleaning

No	Twit	Hasil Cleaning
1	Saya sebagai warga negara yang mencintai indonesia, mengijinkan Ibukota dipindahkan ke pulau Kalimantan pak. Lebih cepat lebih baik!	Saya sebagai warga negara yang mencintai indonesia mengijinkan Ibukota dipindahkan ke pulau Kalimantan pak Lebih cepat lebih baik
2	Usahlah bersedih, semoga Ibukota gak pindah,dan batal dilantik,Aamiin..	Usahlah bersedih semoga Ibukota gak pindah dan batal dilantik Aamiin
3	Tanpa Payung Hukum, Rencana Jokowi Pindah Ibukota Cacat Prosedur	Tanpa Payung Hukum Rencana Jokowi Pindah Ibukota Cacat Prosedur
4	Pendapat saya pribadi, Ibukota dipindahkan ada dampak positifny yg. Saya setuju!	Pendapat saya pribadi Ibukota dipindahkan ada dampak positifny yg Saya setuju

Tokenisasi

Merupakan proses pemisahan dokumen teks menjadi kata per kata.

Dalam tokenisasi juga dilakukan penghapusan karakter-karakter yang dianggap tidak penting dan berpotensi mengganggu proses penelitian, seperti tanda baca, *link website*, dll. Berikut adalah contoh data yang telah melalui proses tokenisasi yang ditampilkan pada Tabel 4.2.

Tabel 4.2 Contoh Tokenisasi

No	Hasil Cleaning	Hasil Tokenisasi
1	Saya sebagai warga negara yang mencintai indonesia mengijinkan Ibukota dipindahkan ke pulau Kalimantan pak Lebih cepat lebih baik	Saya sebagai warga negara yang mencintai indonesia mengijinkan Ibukota dipindahkan ke pulau Kalimantan pak Lebih cepat lebih baik
2	Usahlah bersedih semoga Ibukota gak pindah dan batal dilantik Aamiin	Usahlah bersedih semoga Ibukota gak pindah dan batal dilantik Aamiin
3	Tanpa Payung Hukum Rencana Jokowi Pindah Ibukota Cacat Prosedur	Tanpa Payung Hukum Rencana Jokowi Pindah Ibukota Cacat

No	Hasil Cleaning	Hasil Tokenisasi
4	Pendapat saya pribadi Ibukota dipindahkan ada dampak positifny yg Saya setuju	Pendapat saya pribadi Ibukota dipindahkan ada dampak positifny yg Saya setuju

Case Folding

Dalam tahap ini, data yang telah ditokenisasi berupa kata per kata kemudian akan dikonversi menjadi kata-kata dalam huruf kecil. Tahap ini dilakukan untuk menghindari kesalahan interpretasi term karena adanya kapitalisasi. Berikut adalah contoh data yang telah melalui tahap Case Folding yang ditampilkan pada Tabel 4.3.

Tabel 4.3 Contoh Case Folding

No	Hasil Tokenisasi	Hasil Case Folding
1	Saya sebagai warga negara yang mencintai indonesia mengijinkan Ibukota dipindahkan ke pulau Kalimantan pak Lebih cepat lebih baik	saya sebagai warga negara yang mencintai indonesia mengijinkan Ibukota dipindahkan ke pulau kalimantan pak lebih cepat lebih baik
2	Usahlah bersedih semoga	usahlah bersedih semoga

No	Hasil Tokenisasi	Hasil Case Folding
1	Ibukota gak pindah dan batal dilantik Aamiin	Ibukota gak pindah dan batal dilantik aamiin
2	Tanpa Payung Hukum Rencana Jokowi Pindah Ibukota Cacat Prosedur	tanpa payung hukum rencana jokowi pindah Ibukota cacat prosedur
3	Pendapat saya pribadi Ibukota dipindahkan ada dampak positifny jg Saya setuju	pendapat saya pribadi Ibukota dipindahkan ada dampak positifny jg saya setuju

Normalisasi kata

Pada tahap ini dilakukan penggantian kata-kata yang bersifat ambigu, seperti misalnya kata yang disingkat, menjadi kata yang baku. Selain itu dalam tahap ini juga dilakukan proses penggantian kata yang mengandung kata tidak menjadi kata sinonimnya yang tidak mengandung kata tidak, hal ini dilakukan karena kata tidak akan hilang pada saat tahap *Filtering*, sehingga apabila hal itu terjadi akan mengubah makna yang dimaksud dan akan berpengaruh pada hasil analisis sentimen. Proses Normalisasi kata dilakukan berdasarkan korpus yang berisi kumpulan kata ambigu beserta bentuk bakunya dan kumpulan frasa bernegasi beserta sinonimnya. Korpus yang digunakan dalam penelitian ini untuk melakukan proses Normalisasi kata akan disertakan dalam Lampiran skripsi. Berikut adalah contoh data yang telah melalui tahap normalisasi kata yang ditampilkan pada Tabel 4.4.

**Tabel 4.4 Contoh Normalisasi kata**

No	Hasil Case Folding	Hasil Normalisasi kata
1	saya sebagai warga negara yang mencintai indonesia mengijinkan Ibukota dipindahkan ke pulau kalimantan pak lebih cepat lebih baik	saya sebagai warga negara yang mencintai indonesia mengizinkan Ibukota dipindahkan ke pulau kalimantan pak lebih cepat lebih baik
2	usahlah bersedih semoga Ibukota gak pindah dan batal dilantik aamiin	usahlah bersedih semoga Ibukota tidak pindah dan batal dilantik aamiin
3	tanpa payung hukum rencana jokowi pindah Ibukota cacat prosedur	tanpa payung hukum rencana jokowi pindah Ibukota cacat prosedur
4	pendapat saya pribadi Ibukota dipindahkan ada	pendapat saya pribadi Ibukota dipindahkan ada

No	Hasil Case Folding	Hasil Normalisasi kata
1	dampak positifny jg saya setuju	dampak positifnya juga saya setuju

Type

Pada tahap ini dilakukan penghapusan kata-kata yang muncul lebih dari satu kali. Kata-kata yang berlebih tersebut akan dihapus sehingga kata akan muncul hanya satu kali dalam dokumen. Berikut adalah contoh data yang telah melalui tahap type yang ditampilkan pada Tabel 4.5.

Tabel 4.5 Contoh Type

No	Hasil Normalisasi Kata	Hasil Type
1	saya sebagai warga negara yang mencintai indonesia mengizinkan Ibukota dipindahkan ke pulau kalimantan pak lebih cepat lebih baik	saya sebagai warga negara yang mencintai indonesia mengizinkan Ibukota dipindahkan ke pulau kalimantan pak lebih cepat baik
2	usahalah bersedih semoga Ibukota tidak pindah dan batal dilantik aamiin	usahalah bersedih semoga tidak pindah dan batal dilantik aamiin
3	tanpa	tanpa

No	Hasil Normalisasi Kata	Hasil Type
1	payung hukum rencana jokowi pindah Ibukota cacat prosedur	rencana jokowi cacat prosedur pendapat pribadi ada dampak positifnya juga setuju
4	pendapat saya pribadi Ibukota dipindahkan ada dampak positifnya juga saya setuju	

4.2.3 Perhitungan *Raw term frequency* (RTF)

Pada tahap ini dilakukan perhitungan kemunculan *term* pada data-data twit per label/kelas. Karena pada penelitian ini, fitur yang digunakan adalah *bag of words* maka hasil perhitungan *Raw term frequency* ini nantinya akan digunakan sebagai *classifier* untuk proses klasifikasi data uji. Semakin besar nilai *Raw term frequency* dari sebuah term, maka akan semakin memperbesar peluang sebuah data terkласifikasi ke dalam kelas yang berisi *term* tersebut.

Untuk melakukan klasifikasi perlu dilakukan perhitungan *Term Frequency* (TF) dari tiap *term* untuk setiap dokumen. Nilai TF yang didapatkan kemudian digunakan untuk menghitung nilai *Raw term frequency* dari tiap-tiap *term* itu sendiri. Tabel 4.6 menunjukkan contoh perhitungan *Raw term frequency* dari data latih yang digunakan untuk perhitungan manual.

Tabel 4.6 Contoh Perhitungan *Raw term frequency*

No	Term	D1	D2	D3	D4	Raw term frequency
1	saya	1	0	0	2	3
2	sebagai	1	0	0	0	1
3	warga	1	0	0	0	1
4	negara	1	0	0	0	1
5	yang	1	0	0	0	1
6	mencintai	1	0	0	0	1
7	indonesia	1	0	0	0	1

No	Term	D1	D2	D3	D4	Raw term frequency
8	mengizinkan	1	0	0	0	1
9	Ibukota	1	1	1	1	4
10	dipindahkan	1	0	0	1	2
11	ke	1	0	0	0	1
12	pulau	1	0	0	0	1
13	kalimantan	1	0	0	0	1
14	pak	1	0	0	0	1
15	lebih	2	0	0	0	2
16	cepat	1	0	0	0	1
17	baik	1	0	0	0	1
18	usahalah	0	1	0	0	2
19	bersedih	0	1	0	0	1
20	semoga	0	1	0	0	1
21	tidak	0	1	0	0	1
22	pindah	0	1	1	0	2
23	dan	0	1	0	0	1
24	batal	0	1	0	0	1
25	dilantik	0	1	0	0	1
26	aamiin	0	1	0	0	1
27	tanpa	0	0	1	0	1
28	payung	0	0	1	0	1
29	hukum	0	0	1	0	1
30	rencana	0	0	1	0	1
31	jokowi	0	0	1	0	1
32	cacat	0	0	1	0	1
33	prosedur	0	0	1	0	1
34	pendapat	0	0	0	1	1
35	pribadi	0	0	0	1	1
36	ada	0	0	0	1	1
37	dampak	0	0	0	1	1
38	positifnya	0	0	0	1	1
39	juga	0	0	0	1	1
40	setuju	0	0	0	1	1
Total		18	10	9	11	48

Term yang didapatkan pada Tabel 4.6 merupakan hasil dari sistem. Data-data contoh juga masih terdapat banyak sekali kata-kata yang tidak baku yang sulit untuk dinormalisasi oleh sistem sehingga dalam mencari term dari data twit sistem masih mengalami kendala dalam hal ini. Namun, hal ini masih dapat ditoleransi karena sebagian besar kata tidak baku dalam data twit berhasil diubah oleh sistem menjadi term dalam bentuk baku.

4.2.4 Term-Based Random Sampling (TBRS)

Tahap perhitungan *Term-Based Random Sampling* untuk mendapatkan *stoplist* dilakukan setelah proses Normalisasi kata pada *Preprocessing*. *Stoplist* yang dihasilkan akan digunakan pada tahap *Filtering*, yaitu penghapusan kata-kata yang tidak perlu pada dokumen. Untuk melakukan proses perhitungan manual *Term-Based Random Sampling* maka akan digunakan contoh data yang ditampilkan pada Tabel 4.7.

Tabel 4.7 Contoh Data untuk Perhitungan Manual TBRS

No	Data
1	Saya sebagai warga negara yang mencintai indonesia, mengijinkan Ibukota dipindahkan ke pulau Kalimantan pak. Lebih cepat lebih baik!
2	Sudah tepat klo Ibukota harus pindah. Penurunan permukaan tanah di Jakarta jadi persoalan serius. Di wilayah utara Jakarta sendiri, air laut sudah masuk ke wilayah permukiman dan mengurangi batas wilayah.

Dari data pada Tabel 4.7 kemudian diproses pada *Preprocessing* hingga tahap Normalisasi kata. Setelah *term* hasil Normalisasi kata diperoleh, kemudian dilakukan perhitungan TF pada setiap *term* dari tiap dokumen. Tabel 4.8 menunjukkan hasil perhitungan TF dari data pada Tabel 4.7.

Tabel 4.8 Hasil Perhitungan TF

No	Term	D1	D2	D3	D4	Total Term Frequency
1	saya	1	0	0	2	3
2	sebagai	1	0	0	0	1
3	warga	1	0	0	0	1
4	negara	1	0	0	0	1
5	yang	1	0	0	0	1
6	mencintai	1	0	0	0	1
7	indonesia	1	0	0	0	1
8	mengizinkan	1	0	0	0	1
9	Ibukota	1	1	1	1	4
10	dipindahkan	1	0	0	1	2
11	ke	1	0	0	0	1
12	pulau	1	0	0	0	1
13	kalimantan	1	0	0	0	1
14	pak	1	0	0	0	1
15	lebih	2	0	0	0	2
16	cepat	1	0	0	0	1
17	baik	1	0	0	0	1
18	usahlah	0	1	0	0	2
19	bersedih	0	1	0	0	1
20	semoga	0	1	0	0	1



No	Term	D1	D2	D3	D4	Total Term Frequency
21	tidak	0	1	0	0	1
22	pindah	0	1	1	0	2
23	dan	0	1	0	0	1
24	batal	0	1	0	0	1
25	dilantik	0	1	0	0	1
26	aamiin	0	1	0	0	1
27	tanpa	0	0	1	0	1
28	payung	0	0	1	0	1
29	hukum	0	0	1	0	1
30	rencana	0	0	1	0	1
31	jokowi	0	0	1	0	1
32	cacat	0	0	1	0	1
33	prosedur	0	0	1	0	1
34	pendapat	0	0	0	1	1
35	pribadi	0	0	0	1	1
36	ada	0	0	0	1	1
37	dampak	0	0	0	1	1
38	positifnya	0	0	0	1	1
39	juga	0	0	0	1	1
40	setuju	0	0	0	1	1
Total		18	10	9	11	48

Setelah didapatkan nilai perhitungan TF, kemudian dilakukan perhitungan nilai bobot Kullback-Leibler untuk setiap kata. Perhitungan ini dilakukan berdasarkan rumus pada Persamaan 2.4. Berikut adalah contoh perhitungan bobot Kullback-Leibler:

Misalkan, kata pertama yang akan dihitung nilai bobot Kullback-Leiblernya adalah kata “saya”. Kata “saya” muncul pada dokumen 1 dan dokumen 4, sehingga kedua dokumen tersebut menjadi dokumen sampel untuk kata “saya”.

$$Px(\text{saya}) = \frac{tf_x}{lx} = \frac{1+2}{18+11} = \frac{3}{29}$$

$$Pc(\text{saya}) = \frac{F}{tokenc} = \frac{3}{48}$$

Setelah didapatkan nilai Px dan Pc dari kata “saya”, kemudian dihitung nilai bobot Kullback-Leiblernya.

$$W(\text{saya}) = Px \cdot \log_2 \frac{Px}{Pc} = \frac{3}{29} \cdot \log_2 \frac{3/29}{3/48} = 0,1034 \times 0,727 = 0,0752$$



Setelah itu, dilakukan perhitungan bobot Kullback-Leibler untuk kata “sebagai”.

Kata “sebagai” muncul pada dokumen 1, sehingga dokumen 1 menjadi dokumen sampel untuk kata “sebagai”. Berikut adalah proses perhitungan manual bobot Kullback-Leibler untuk kata “sebagai”:

$$Px(\text{sebagai}) = \frac{tfx}{lx} = \frac{1}{18}$$
$$Pc(\text{sebagai}) = \frac{F}{tokenc} = \frac{1}{48}$$
$$W(\text{sebagai}) = Px \cdot \log_2 \frac{Px}{Pc} = \frac{1}{18} \cdot \log_2 \frac{1/18}{1/48} = 0,555 \times 1,415 = 0,78613$$

Kata “warga”, “negara”, “yang”, “mencintai”, “indonesia” dan “mengizinkan” memiliki nilai bobot Kullback-Leibler yang sama dengan kata “sebagai” karena memiliki nilai TF yang sama dan sama-sama hanya muncul pada dokumen 1.

Setelah itu, perhitungan langsung dilanjutkan untuk kata “ibukota”. Kata “ibukota” muncul pada semua dokumen, sehingga semua dokumen menjadi dokumen sampel untuk kata “ibukota”. Berikut adalah perhitungan nilai bobot Kullback-Leibler untuk kata “ibukota”:

$$Px(\text{ibukota}) = \frac{tfx}{lx} = \frac{1+1+1+1}{18+10+9+11} = \frac{4}{48} = \frac{1}{12}$$
$$Pc(\text{ibukota}) = \frac{F}{tokenc} = \frac{4}{48} = \frac{1}{12}$$
$$W(\text{ibukota}) = Px \cdot \log_2 \frac{Px}{Pc} = \frac{1}{12} \cdot \log_2 \frac{1/12}{1/12} = 0,083 \times 0 = 0$$

Dan seterusnya, perhitungan bobot Kullback-Leibler dilakukan hingga kata terakhir. Perlu diketahui, bahwa nilai bobot Kullback-Leibler merepresentasikan tingkat kepentingan dari suatu kata. Semakin penting suatu kata, maka akan semakin besar pula nilai bobot Kullback-Leiblernya. Nilai bobot Kullback-Leibler juga dipengaruhi oleh kemunculan kata pada dokumen. Semakin sering suatu kata muncul dalam banyak dokumen, maka nilai bobot Kullback-Leiblernya akan semakin kecil. Hal ini dapat dilihat dari contoh perhitungan manualisasi yang telah dilakukan diatas. Nilai bobot Kullback-Leibler untuk kata “sebagai” memiliki nilai yang lebih besar daripada nilai bobot Kullback-Leibler untuk kata “ibukota”, karena kata “ibukota” muncul sebanyak total 4 kali yaitu sekali pada masing-masing dokumen 1, 2, 3 dan 4, sedangkan kata “sebagai” hanya muncul sekali pada dokumen 1.



Setelah didapatkan nilai bobot Kullback-Leibler dari seluruh kata, kemudian seluruh kata diurutkan berdasarkan nilai bobot Kullback-Leiblernya. Hal ini dilakukan untuk keperluan pemilihan kata yang masuk ke dalam *stoplist*. Tabel 4.9 menunjukkan daftar kata yang telah diurutkan secara *ascending* berdasarkan nilai bobot Kullback-Leiblernya.

Tabel 4.9 Urutan Kata Berdasarkan Nilai bobot Kullback-Leibler

No	Token	Bobot KL
1	Ibukota	0
2	dipindahkan	0,050137
3	saya	0,075205
4	warga	0,078613
5	negara	0,078613
6	yang	0,078613
7	mencintai	0,078613
8	indonesia	0,078613
9	mengizinkan	0,078613
10	ke	0,078613
11	pulau	0,078613
12	kalimantan	0,078613
13	pak	0,078613
14	cepat	0,078613
15	baik	0,078613
16	sebagai	0,078613
17	pindah	0,140741
18	lebih	0,157226
19	pendapat	0,19323
20	pribadi	0,19323
21	ada	0,19323
22	dampak	0,19323
23	positifnya	0,19323



No	Token	Bobot KL
24	juga	0,19323
25	setuju	0,19323
26	usahalah	0,226303
27	bersedih	0,226303
28	semoga	0,226303
29	tidak	0,226303
30	dan	0,226303
31	batal	0,226303
32	dilantik	0,226303
33	aamiin	0,226303
34	tanpa	0,268337
35	payung	0,268337
36	hukum	0,268337
37	rencana	0,268337
38	jokowi	0,268337
39	cacat	0,268337
40	prosedur	0,268337

Setelah didapatkan daftar kata terurut secara *ascending* berdasarkan nilai bobot Kullback-Leiblernya, kemudian dilakukan pemilihan kata yang masuk ke dalam *stoplist*. Daftar kata yang masuk ke dalam *stoplist* adalah yang memiliki nilai bobot Kullback-Leibler terendah sebanyak X persen dari total jumlah kata. X merupakan parameter sehingga nilainya dapat diubah-ubah. Untuk keperluan pengujian akan diambil 0 persen, 10 persen, 20 persen, 30 persen, 40 persen, 50 persen dan 60 persen dari seluruh daftar kata, namun terdapat kata-kata khusus yang dianggap penting dalam proses analisis sentimen, namun kemunculannya cukup sering karena kata tersebut merepresentasikan topik dari permasalahan yang diangkat dan akan dianalisis sentimennya. Untuk topik permasalahan kebijakan pemindahan Ibukota, pada Tabel 4.11 adalah contoh *lexicon* yang berisi kata yang termasuk dalam *non stopword*.

Tabel 4.10 Contoh Lexicon Non Stopword

No	Lexicon Non Stopword
1	Ibukota
2	pindah
3	kalimantan
4	jakarta
5	negara
6	indonesia
7	bijak
8	pemindahan
9	dipindahkan
10	baik
11	mengizinkan
12	mencintai
13	dampak
14	pulau
15	positifnya
16	setuju
17	jokowi
18	hukum
19	rencana
20	kebijakan

Setelah pengurutan kata berdasarkan nilai bobot Kullback-Leibler dilakukan dan *lexicon non stopword* telah dibuat, maka selanjutnya dilakukan pemilihan kata yang termasuk ke dalam *stoplist*. Untuk *stoplist* yang dibuat dari 0 persen daftar token data latih hasilnya adalah 0 *stopword*, ini berarti tidak dilakukan *Filtering*. Selanjutnya, untuk daftar 10 persen kata untuk *stoplist* ditunjukkan pada Tabel 4.11.

Tabel 4.11 Daftar 10 Persen Kata untuk Stoplist

No	Kata	Bobot Kullback-Leibler
1	saya	0,075205
2	warga	0,078613
3	yang	0,078613
4	ke	0,078613

Selanjutnya dibuat *stoplist* dari 20 persen daftar kata pada data latih. Ini berarti, akan diambil 8 kata dengan nilai bobot Kullback-Leibler terendah dari daftar kata tersebut untuk dijadikan *stoplist*. Tabel 4.12 menunjukkan hasil pemilihan *stoplist* sebanyak 20 persen kata dari data latih.

Tabel 4.12 Daftar 20 Persen Kata untuk Stoplist

No	Kata	Bobot Kullback-Leibler
1	saya	0,075205
2	warga	0,078613
3	yang	0,078613
4	ke	0,078613
5	pak	0,078613
6	cepat	0,078613
7	sebagai	0,078613
8	lebih	0,157226

Setelah itu, dibuat *stoplist* dari 30 persen daftar kata pada data latih. Ini berarti, akan diambil 12 kata dengan nilai bobot Kullback-Leibler terendah dari daftar kata tersebut untuk dijadikan *stoplist*. Tabel 4.13 menunjukkan hasil pemilihan *stoplist* sebanyak 30 persen kata dari data latih.

Tabel 4.13 Daftar 30 Persen Kata untuk Stoplist

No	Kata	Bobot Kullback-Leibler
1	saya	0,075205
2	warga	0,078613
3	yang	0,078613
4	ke	0,078613
5	pak	0,078613
6	cepat	0,078613
7	sebagai	0,078613
8	lebih	0,157226
9	pendapat	0,19323
10	pribadi	0,19323
11	ada	0,19323
12	juga	0,19323

Selanjutnya dibuat *stoplist* dari 40 persen daftar kata pada data latih. Ini berarti, akan diambil 16 kata dengan nilai bobot Kullback-Leibler terendah dari daftar kata tersebut untuk dijadikan *stoplist*. Tabel 4.14 menunjukkan hasil pemilihan *stoplist* sebanyak 40 persen kata dari data latih.

Tabel 4.14 Daftar 40 Persen Kata untuk Stoplist

No	Kata	Bobot Kullback-Leibler
1	saya	0,075205
2	warga	0,078613

No	Kata	Bobot Kullback-Leibler
3	yang	0,078613
4	ke	0,078613
5	pak	0,078613
6	cepat	0,078613
7	sebagai	0,078613
8	lebih	0,157226
9	pendapat	0,19323
10	pribadi	0,19323
11	ada	0,19323
12	juga	0,19323
13	usahalah	0,226303
14	bersedih	0,226303
15	semoga	0,226303
16	tidak	0,226303

Setelah itu, dibuat *stoplist* dari 50 persen daftar kata pada data latih. Ini berarti, akan diambil 20 kata dengan nilai bobot Kullback-Leibler terendah dari daftar kata tersebut untuk dijadikan *stoplist*. Tabel 4.15 menunjukkan hasil pemilihan *stoplist* sebanyak 50 persen kata dari data latih.

Tabel 4.15 Daftar 50 Persen Kata untuk *Stoplist*

No	Kata	Bobot Kullback-Leibler
1	saya	0,075205
2	warga	0,078613
3	yang	0,078613
4	ke	0,078613
5	pak	0,078613
6	cepat	0,078613
7	sebagai	0,078613
8	lebih	0,157226
9	pendapat	0,19323
10	pribadi	0,19323
11	ada	0,19323
12	juga	0,19323
13	usahalah	0,226303
14	bersedih	0,226303
15	semoga	0,226303
16	tidak	0,226303
17	dan	0,226303
18	batal	0,226303

No	Kata	Bobot Kullback-Leibler
19	dilantik	0,226303
20	aamiin	0,226303

Terakhir, dibuat *stoplist* dari 60 persen daftar kata pada data latih. Ini berarti, akan diambil 24 kata dengan nilai bobot Kullback-Leibler terendah dari daftar kata tersebut untuk dijadikan *stoplist*. Tabel 4.16 menunjukkan hasil pemilihan *stoplist* sebanyak 60 persen kata dari data latih.

Tabel 4.16 Daftar 60 Persen Kata untuk *Stoplist*

No	Kata	Bobot Kullback-Leibler
1	saya	0,075205
2	warga	0,078613
3	yang	0,078613
4	ke	0,078613
5	pak	0,078613
6	cepat	0,078613
7	sebagai	0,078613
8	lebih	0,157226
9	pendapat	0,19323
10	pribadi	0,19323
11	ada	0,19323
12	juga	0,19323
13	usaha	0,226303
14	bersedih	0,226303
15	semoga	0,226303
16	tidak	0,226303
17	dan	0,226303
18	batal	0,226303
19	dilantik	0,226303
20	aamiin	0,226303
21	tanpa	0,268337
22	payung	0,268337
23	cacat	0,268337
24	prosedur	0,268337

4.2.5 Filtering dengan menggunakan *Stoplist*

Setelah didapatkan *stoplist*, maka proses *Filtering* terhadap data latih dapat dilakukan. Sebelumnya, proses pemilihan kata untuk dijadikan *stoplist* dilakukan

sebanyak 7 kali, yaitu *stoplist* dari 0 persen, 10 persen, 20 persen, 30 persen, 40 persen, 50 persen dan 60 persen daftar token data latih.

Untuk *stoplist* dari 0 persen daftar kata data latih akan dihasilkan sebanyak 0 *stopword*. Hal ini berarti sama dengan tidak dilakukan *Filtering* pada data latih.

Oleh karena itu, didapatkan keluaran seperti yang ditampilkan pada Tabel 4.17.

Tabel 4.17 Daftar Kata Setelah *Filtering* dengan Stoplist 0 Persen dari Total Kata

Data Latih

No	Term pada Data Latih Positif	RTF	Term pada Data Latih Negatif	RTF
1	Ibukota	2	Ibukota	2
2	dipindahkan	2	pindah	2
3	sebagai	1	usahlah	1
4	warga	1	bersedih	1
5	negara	1	semoga	1
6	yang	1	tidak	1
7	mencintai	1	dan	1
8	indonesia	1	batal	1
9	mengizinkan	1	dilantik	1
10	ke	1	aamiin	1
11	pulau	1	tanpa	1
12	kalimantan	1	payung	1
13	pak	1	hukum	1
14	cepat	1	rencana	1
15	baik	1	jokowi	1
16	saya	3	cacat	1
17	lebih	2	prosedur	1
18	pendapat	1	-	-
19	pribadi	1	-	-
20	ada	1	-	-
21	dampak	1	-	-
22	positifnya	1	-	-

No	Term pada Data Latih Positif	RTF	Term pada Data Latih Negatif	RTF
23	juga	1	-	-
24	setuju	1	-	-
	Total	29	Total	19

Setelah itu, dilakukan proses *Filtering* terhadap data latih dengan *stoplist* 10 persen dari total kata data latih. Hasil dari *Filtering* dengan menggunakan *stoplist* 10 persen dari total kata data latih ditampilkan pada Tabel 4.18.

Tabel 4.18 Daftar Kata Setelah *Filtering* dengan Stoplist 10 Persen dari Total Kata Data Latih

No	Term pada Data Latih Positif	RTF	Term pada Data Latih Negatif	RTF
1	Ibukota	2	Ibukota	2
2	dipindahkan	2	pindah	2
3	negara	1	usahlah	1
4	mencintai	1	bersedih	1
5	indonesia	1	semoga	1
6	mengizinkan	1	tidak	1
7	pulau	1	dan	1
8	kalimantan	1	batal	1
9	pak	1	dilantik	1
10	cepat	1	aamiin	1
11	baik	1	tanpa	1
12	saya	3	payung	1
13	lebih	2	hukumrawijaya	1
14	pendapat	1	rencana	1
15	pribadi	1	jokowi	1
16	ada	1	cacat	1
17	dampak	1	prosedur	1
18	positifnya	1	-	-

No	Term pada Data Latih Positif	RTF	Term pada Data Latih Negatif	RTF
19	juga	1	-	-
20	setuju	1	-	-
	Total	25	Total	19

Setelah itu, dilakukan proses *Filtering* terhadap data latih dengan *stoplist* 20 persen dari total kata data latih. Hasil dari *Filtering* dengan menggunakan *stoplist* 20 persen dari total kata data latih ditampilkan pada Tabel 4.19.

Tabel 4.19 Daftar Kata Setelah *Filtering* dengan Stoplist 20 Persen dari Total Kata Data Latih

No	Term pada Data Latih Positif	RTF	Term pada Data Latih Negatif	RTF
1	Ibukota	2	Ibukota	2
2	dipindahkan	2	pindah	2
3	negara	1	usahlah	1
4	mencintai	1	bersedih	1
5	indonesia	1	semoga	1
6	mengizinkan	1	tidak	1
7	pulau	1	dan	1
8	kalimantan	1	batal	1
9	baik	1	dilantik	1
10	pendapat	1	aamiin	1
11	pribadi	1	tanpa	1
12	ada	1	payung	1
13	dampak	1	hukumrawijaya	1
14	positifnya	1	rencana	1
15	juga	1	jokowi	1
16	setuju	1	cacat	1
17	-	-	prosedur	1
	Total	18	Total	19

Setelah itu, dilakukan proses *Filtering* terhadap data latih dengan *stoplist* 30 persen dari total kata data latih. Hasil dari *Filtering* dengan menggunakan *stoplist* 30 persen dari total kata data latih ditampilkan pada Tabel 4.20.

Tabel 4.20 Daftar Kata Setelah *Filtering* dengan Stoplist 30 Persen dari Total Kata Data Latih

No	Term pada Data Latih Positif	RTF	Term pada Data Latih Negatif	RTF
1	Ibukota	2	Ibukota	2
2	dipindahkan	2	pindah	2
3	negara	1	usahlah	1
4	mencintai	1	bersedih	1
5	indonesia	1	semoga	1
6	mengizinkan	1	tidak	1
7	pulau	1	dan	1
8	kalimantan	1	batal	1
9	baik	1	dilantik	1
10	dampak	1	aamiin	1
11	positifnya	1	tanpa	1
12	setuju	1	payung	1
13	-	-	hukum	1
14	-	-	rencana	1
15	-	-	jokowi	1
16	-	-	cacat	1
17	-	-	prosedur	1
Total		14	Total	19

Setelah itu, dilakukan proses *Filtering* terhadap data latih dengan *stoplist* 40 persen dari total kata data latih. Hasil dari *Filtering* dengan menggunakan *stoplist* 40 persen dari total kata data latih ditampilkan pada Tabel 4.21.

Tabel 4.21 Daftar Kata Setelah *Filtering* dengan Stoplist 40 Persen dari Total Kata Data Latih

No	Term pada Data Latih Positif	RTF	Term pada Data Latih Negatif	RTF
1	Ibukota	2	Ibukota	2
2	dipindahkan	2	pindah	2
3	negara	1	dan	1
4	mencintai	1	batal	1
5	indonesia	1	dilantik	1
6	mengizinkan	1	aamiin	1
7	pulau	1	tanpa	1
8	kalimantan	1	payung	1
9	baik	1	hukum	1
10	dampak	1	rencana	1
11	positifnya	1	jokowi	1
12	setuju	1	cacat	1
13	-	-	prosedur	1
Total		14	Total	15

Setelah itu, dilakukan proses *Filtering* terhadap data latih dengan *stoplist* 50 persen dari total kata data latih. Hasil dari *Filtering* dengan menggunakan *stoplist* 50 persen dari total kata data latih ditampilkan pada Tabel 4.22.

Tabel 4.22 Daftar Kata Setelah *Filtering* dengan Stoplist 50 Persen dari Total Kata Data Latih

No	Term pada Data Latih Positif	RTF	Term pada Data Latih Negatif	RTF
1	Ibukota	2	Ibukota	2
2	dipindahkan	2	pindah	2
3	negara	1	tanpa	1
4	mencintai	1	payung	1
5	indonesia	1	hukum	1
6	mengizinkan	1	rencana	1

No	Term pada Data Latih Positif	RTF	Term pada Data Latih Negatif	RTF
7	pulau	1	jokowi	1
8	kalimantan	1	cacat	1
9	baik	1	prosedur	1
10	dampak	1	-	-
11	positifnya	1	-	-
12	setuju	1	-	-
Total		14	Total	11

Setelah itu, dilakukan proses *Filtering* terhadap data latih dengan *stoplist* 60 persen dari total kata data latih. Hasil dari *Filtering* dengan menggunakan *stoplist* 60 persen dari total kata data latih ditampilkan pada Tabel 4.23.

Tabel 4.23 Daftar Kata Setelah *Filtering* dengan Stoplist 60 Persen dari Total Kata Data Latih

No	Term pada Data Latih Positif	RTF	Term pada Data Latih Negatif	RTF
1	Ibukota	2	Ibukota	2
2	dipindahkan	2	pindah	2
3	negara	1	hukum	1
4	mencintai	1	rencana	1
5	indonesia	1	jokowi	1
6	mengizinkan	1	-	-
7	pulau	1	-	-
8	kalimantan	1	-	-
9	baik	1	-	-
10	dampak	1	-	-
11	positifnya	1	-	-
12	setuju	1	-	-
Total		14	Total	7

4.2.6 Klasifikasi

Setelah didapatkan nilai *term frequency* (TF) pada tiap label, kemudian akan dilakukan tahap klasifikasi data uji berdasarkan *classifier* yang telah didapatkan.

Metode klasifikasi yang digunakan yaitu metode Naïve Bayes. Perhitungan klasifikasi menggunakan rumus pada Persamaan 2.1 dengan nilai *likelihood* didapatkan dengan menggunakan *Multinomial Model* dengan rumus pada Persamaan 2.3.

Setelah didapatkan nilai probabilitas *likelihood* dari tiap kata pada suatu data, kemudian dihitung nilai probabilitas *posteriornya* dengan mengalikan semua nilai probabilitas *likelihood* tiap kata dan *prior*. Setelah itu, akan didapatkan nilai probabilitas *posterior* dari data tersebut terhadap tiap kelas. Untuk menentukan suatu data termasuk ke dalam kelas apa, maka dipilih hasil perhitungan probabilitas *posterior* terbesar.

Manualisasi perhitungan dilakukan dengan tujuan untuk menyamakan hasil perhitungan klasifikasi secara manual dengan hasil perhitungan sistem. Apabila hasil perhitungan sistem sama dengan hasil perhitungan manual, maka kode program sistem dianggap telah benar, dan sebaliknya apabila berbeda maka perlu dilakukan perbaikan terhadap kode program sistem, hingga hasil keduanya sama.

Pada Tabel 4.24 diuraikan data nilai *Raw term frequency* dari setiap *term* dalam setiap kelas. Nilai *Raw term frequency* akan digunakan sebagai fitur. Fitur tersebut akan digunakan untuk melakukan proses klasifikasi data.

Tabel 4.24 Nilai Raw term frequency (RTF) Term pada Tiap Data Latih di Setiap Kelas

Data ke-	Term	RTF pada Data Latih Kelas Positif	RTF pada Data Latih Kelas Negatif
1	saya	3	0
	pribadi	1	0
	setuju	1	0
	ibukota	2	2
	dipindahkan	2	0
	jakarta	0	0
	sudah	0	0
	padat	0	0
	sekali	0	0
2	pokoknya	0	0
	tidak	0	1
	setuju	1	0
	kalau	0	0
	ibukota	2	2
	pindah	0	2
	semoga	0	1



Data ke-	Term	RTF pada Data Latih Kelas Positif	RTF pada Data Latih Kelas Negatif
saja	Universitas Brawijaya	0	0
tidak	Universitas Brawijaya	0	1
jadi	Universitas Brawijaya	0	0
Pindah	Universitas Brawijaya	0	2

4.2.6.1 Klasifikasi dengan Stoplist 0 Persen

Dari Tabel 4.24 kemudian dilakukan perhitungan probabilitas *Posterior* dengan *stoplist* 0 persen untuk setiap data pada tiap kelas sentimen. Perhitungan dilakukan dengan menggunakan rumus Naïve Bayes *Multinomial Model* pada Persamaan 2.3. Proses perhitungannya adalah seperti berikut:

$$P(\text{positif}|D1) = P(\text{positif}) * \prod_{w \in d} P(w|\text{positif})$$

$$\begin{aligned} &= \frac{1}{2} \times P(\text{saya}|\text{positif}) \times P(\text{pribadi}|\text{positif}) \times P(\text{setuju}|\text{positif}) \\ &\quad \times P(\text{ibukota}|\text{positif}) \times P(\text{dipindahkan}|\text{positif}) \times P(\text{jakarta}|\text{positif}) \\ &\quad \times P(\text{sudah}|\text{positif}) \times P(\text{padat}|\text{positif}) \times P(\text{sekali}|\text{positif}) \\ &= \frac{1}{2} \times \frac{(3+1)}{(29+40)} \times \frac{(1+1)}{(29+40)} \times \frac{(1+1)}{(29+40)} \times \frac{(2+1)}{(29+40)} \times \frac{(2+1)}{(29+40)} \\ &\quad \times \frac{(0+1)}{(29+40)} \times \frac{(0+1)}{(29+40)} \times \frac{(0+1)}{(29+40)} \times \frac{(0+1)}{(29+40)} \\ &= \frac{1}{2} \times \frac{(4)}{(69)} \times \frac{(2)}{(69)} \times \frac{(2)}{(69)} \times \frac{(3)}{(69)} \times \frac{(3)}{(69)} \times \frac{(1)}{(69)} \times \frac{(1)}{(69)} \times \frac{(1)}{(69)} \\ &\quad \times \frac{(1)}{(69)} = 2,03091e^{-15} \end{aligned}$$

$$P(\text{negatif}|D1) = P(\text{negatif}) * \prod_{w \in d} P(w|\text{negatif})$$

$$\begin{aligned} &= \frac{1}{2} \times P(\text{saya}|\text{negatif}) \times P(\text{pribadi}|\text{negatif}) \times P(\text{setuju}|\text{negatif}) \\ &\quad \times P(\text{ibukota}|\text{negatif}) \times P(\text{dipindahkan}|\text{negatif}) \times P(\text{jakarta}|\text{negatif}) \\ &\quad \times P(\text{sudah}|\text{negatif}) \times P(\text{padat}|\text{negatif}) \times P(\text{sekali}|\text{negatif}) \\ &= \frac{1}{2} \times \frac{(0+1)}{(19+40)} \times \frac{(0+1)}{(19+40)} \times \frac{(0+1)}{(19+40)} \times \frac{(2+1)}{(19+40)} \times \frac{(0+1)}{(19+40)} \\ &\quad \times \frac{(0+1)}{(19+40)} \times \frac{(0+1)}{(19+40)} \times \frac{(0+1)}{(19+40)} \times \frac{(0+1)}{(19+40)} \end{aligned}$$



$$1,7315e^{-16}$$

Berdasarkan hasil perhitungan probabilitas *Posterior* dengan *stoplist* 0 persen diatas, maka dapat disimpulkan bahwa, secara perhitungan manual D1 termasuk ke dalam kelas **positif**.

$$\begin{aligned} P(\text{positif}|D2) &= P(\text{positif}) * \prod_{w \in d} P(w|\text{positif}) \\ &= \frac{1}{2} x P(\text{pokoknya}|\text{positif}) x P(\text{tidak}|\text{positif}) x P(\text{setuju}|\text{positif}) \\ &\quad x P(\text{kalau}|\text{positif}) x P(\text{ibukota}|\text{positif}) x P(\text{pindah}|\text{positif}) \\ &\quad x P(\text{semoga}|\text{positif}) x P(\text{saja}|\text{positif}) x P(\text{tidak}|\text{positif}) \\ &\quad x P(\text{jadi}|\text{positif}) x P(\text{pindah}|\text{positif}) \\ &= \frac{1}{2} x \frac{(0+1)}{(29+40)} x \frac{(0+1)}{(29+40)} x \frac{(1+1)}{(29+40)} x \frac{(0+1)}{(29+40)} x \frac{(2+1)}{(29+40)} \\ &\quad x \frac{(0+1)}{(29+40)} x \frac{(0+1)}{(29+40)} x \frac{(0+1)}{(29+40)} x \frac{(0+1)}{(29+40)} x \frac{(0+1)}{(29+40)} x \frac{(0+1)}{(29+40)} \\ &= \frac{1}{2} x \frac{(1)}{(69)} x \frac{(1)}{(69)} x \frac{(2)}{(69)} x \frac{(1)}{(69)} x \frac{(3)}{(69)} x \frac{(1)}{(69)} x \frac{(1)}{(69)} x \frac{(1)}{(69)} x \frac{(1)}{(69)} \\ &\quad x \frac{(1)}{(69)} x \frac{(1)}{(69)} = 3,55477e^{-20} \end{aligned}$$

$$P(\text{negatif}|D2) = P(\text{negatif}) * \prod_{w \in d} P(w|\text{negatif})$$

$$\begin{aligned} &= \frac{1}{2} x P(\text{pokoknya}|\text{negatif}) x P(\text{tidak}|\text{negatif}) x P(\text{setuju}|\text{negatif}) \\ &\quad x P(\text{kalau}|\text{negatif}) x P(\text{ibukota}|\text{negatif}) x P(\text{pindah}|\text{negatif}) \\ &\quad x P(\text{semoga}|\text{negatif}) x P(\text{saja}|\text{negatif}) x P(\text{tidak}|\text{negatif}) \\ &\quad x P(\text{jadi}|\text{negatif}) x P(\text{pindah}|\text{negatif}) \\ &= \frac{1}{2} x \frac{(0+1)}{(19+40)} x \frac{(1+1)}{(19+40)} x \frac{(0+1)}{(19+40)} x \frac{(0+1)}{(19+40)} x \frac{(2+1)}{(19+40)} \\ &\quad x \frac{(2+1)}{(19+40)} x \frac{(1+1)}{(19+40)} x \frac{(0+1)}{(19+40)} x \frac{(1+1)}{(19+40)} x \frac{(0+1)}{(19+40)} x \frac{(2+1)}{(19+40)} \\ &= \frac{1}{2} x \frac{(1)}{(59)} x \frac{(2)}{(59)} x \frac{(1)}{(59)} x \frac{(1)}{(59)} x \frac{(3)}{(59)} x \frac{(3)}{(59)} x \frac{(2)}{(59)} x \frac{(1)}{(59)} x \frac{(2)}{(59)} \\ &\quad x \frac{(1)}{(59)} x \frac{(3)}{(59)} = 7,16278e^{-18} \end{aligned}$$



Berdasarkan hasil perhitungan probabilitas *Posterior* dengan *stoplist* 0 persen diatas, maka dapat disimpulkan bahwa, secara perhitungan manual D2 termasuk ke dalam kelas **Negatif**.

4.2.6.2 Klasifikasi dengan *Stoplist* 10 Persen

Dari Tabel 4.24 kemudian dilakukan perhitungan probabilitas *Posterior* dengan *stoplist* 10 persen untuk setiap data pada tiap kelas sentimen. Perhitungan dilakukan dengan menggunakan rumus *Naïve Bayes multinomial model* pada Persamaan 2.3. Proses perhitungannya adalah seperti berikut:

$$\begin{aligned} P(\text{positif}|D1) &= P(\text{positif}) * \prod_{w \in d} P(w|\text{positif}) \\ &= \frac{1}{2} \times P(\text{saya}|\text{positif}) \times P(\text{pribadi}|\text{positif}) \times P(\text{setuju}|\text{positif}) \\ &\quad \times P(\text{ibukota}|\text{positif}) \times P(\text{dipindahkan}|\text{positif}) \times P(\text{jakarta}|\text{positif}) \\ &\quad \times P(\text{sudah}|\text{positif}) \times P(\text{padat}|\text{positif}) \times P(\text{sekali}|\text{positif}) \\ &= \frac{1}{2} \times \frac{(0+1)}{(25+36)} \times \frac{(1+1)}{(25+36)} \times \frac{(1+1)}{(25+36)} \times \frac{(2+1)}{(25+36)} \times \frac{(2+1)}{(25+36)} \\ &\quad \times \frac{(0+1)}{(25+36)} \times \frac{(0+1)}{(25+36)} \times \frac{(0+1)}{(25+36)} \times \frac{(0+1)}{(25+36)} \\ &= \frac{1}{2} \times \frac{(4)}{(61)} \times \frac{(2)}{(61)} \times \frac{(2)}{(61)} \times \frac{(3)}{(61)} \times \frac{(3)}{(61)} \times \frac{(1)}{(61)} \times \frac{(1)}{(61)} \times \frac{(1)}{(61)} \\ &\quad \times \frac{(1)}{(61)} = 1,53923e^{-15} \end{aligned}$$

$$P(\text{negatif}|D1) = P(\text{negatif}) * \prod_{w \in d} P(w|\text{negatif})$$

$$\begin{aligned} &= \frac{1}{2} \times P(\text{saya}|\text{negatif}) \times P(\text{pribadi}|\text{negatif}) \times P(\text{setuju}|\text{negatif}) \\ &\quad \times P(\text{ibukota}|\text{negatif}) \times P(\text{dipindahkan}|\text{negatif}) \times P(\text{jakarta}|\text{negatif}) \\ &\quad \times P(\text{sudah}|\text{negatif}) \times P(\text{padat}|\text{negatif}) \times P(\text{sekali}|\text{negatif}) \\ &= \frac{1}{2} \times \frac{(0+1)}{(19+36)} \times \frac{(0+1)}{(19+36)} \times \frac{(0+1)}{(19+36)} \times \frac{(2+1)}{(19+36)} \times \frac{(0+1)}{(19+36)} \\ &\quad \times \frac{(0+1)}{(19+36)} \times \frac{(0+1)}{(19+36)} \times \frac{(0+1)}{(19+36)} \times \frac{(0+1)}{(19+36)} \\ &= \frac{1}{2} \times \frac{(1)}{(55)} \times \frac{(1)}{(55)} \times \frac{(1)}{(55)} \times \frac{(3)}{(55)} \times \frac{(1)}{(55)} \times \frac{(1)}{(55)} \times \frac{(1)}{(55)} \times \frac{(1)}{(55)} \\ &= 3,25707e^{-16} \end{aligned}$$



Berdasarkan hasil perhitungan probabilitas *Posterior* dengan *stoplist* 10 persen diatas, maka dapat disimpulkan bahwa, secara perhitungan manual D1 termasuk ke dalam kelas **positif**.

$$\begin{aligned} P(\text{positif}|D2) &= P(\text{positif}) * \prod_{w \in d} P(w|\text{positif}) \\ &= \frac{1}{2} x P(\text{pokoknya}|\text{positif}) x P(\text{tidak}|\text{positif}) x P(\text{setuju}|\text{positif}) \\ &\quad x P(\text{kalau}|\text{positif}) x P(\text{ibukota}|\text{positif}) x P(\text{pindah}|\text{positif}) \\ &\quad x P(\text{semoga}|\text{positif}) x P(\text{saja}|\text{positif}) x P(\text{tidak}|\text{positif}) \\ &\quad x P(\text{jadi}|\text{positif}) x P(\text{pindah}|\text{positif}) \\ &= \frac{1}{2} x \frac{(0+1)}{(25+36)} x \frac{(0+1)}{(25+36)} x \frac{(1+1)}{(25+36)} x \frac{(0+1)}{(25+36)} x \frac{(2+1)}{(25+36)} \\ &\quad x \frac{(0+1)}{(25+36)} x \frac{(0+1)}{(25+36)} x \frac{(0+1)}{(25+36)} x \frac{(0+1)}{(25+36)} x \frac{(0+1)}{(25+36)} x \frac{(0+1)}{(25+36)} \\ &= \frac{1}{2} x \frac{(1)}{(61)} x \frac{(1)}{(61)} x \frac{(2)}{(61)} x \frac{(1)}{(61)} x \frac{(3)}{(61)} x \frac{(1)}{(61)} x \frac{(1)}{(61)} x \frac{(1)}{(61)} x \frac{(1)}{(61)} \\ &\quad x \frac{(1)}{(61)} x \frac{(1)}{(61)} = 2,56539e^{-16} \end{aligned}$$

$$P(\text{negatif}|D2) = P(\text{negatif}) * \prod_{w \in d} P(w|\text{negatif})$$

$$\begin{aligned} &= \frac{1}{2} x P(\text{pokoknya}|\text{negatif}) x P(\text{tidak}|\text{negatif}) x P(\text{setuju}|\text{negatif}) \\ &\quad x P(\text{kalau}|\text{negatif}) x P(\text{ibukota}|\text{negatif}) x P(\text{pindah}|\text{negatif}) \\ &\quad x P(\text{semoga}|\text{negatif}) x P(\text{saja}|\text{negatif}) x P(\text{tidak}|\text{negatif}) \\ &\quad x P(\text{jadi}|\text{negatif}) x P(\text{pindah}|\text{negatif}) \\ &= \frac{1}{2} x \frac{(0+1)}{(19+36)} x \frac{(1+1)}{(19+36)} x \frac{(0+1)}{(19+36)} x \frac{(0+1)}{(19+36)} x \frac{(2+1)}{(19+36)} \\ &\quad x \frac{(2+1)}{(19+36)} x \frac{(1+1)}{(19+36)} x \frac{(0+1)}{(19+36)} x \frac{(1+1)}{(19+36)} x \frac{(0+1)}{(19+36)} x \frac{(2+1)}{(19+36)} \\ &= \frac{1}{2} x \frac{(1)}{(55)} x \frac{(2)}{(55)} x \frac{(1)}{(55)} x \frac{(1)}{(55)} x \frac{(3)}{(55)} x \frac{(3)}{(55)} x \frac{(2)}{(55)} x \frac{(1)}{(55)} x \frac{(2)}{(55)} \\ &\quad x \frac{(1)}{(55)} x \frac{(3)}{(55)} = 7,81697e^{-15} \end{aligned}$$



Berdasarkan hasil perhitungan probabilitas *Posterior* dengan *stoplist* 10 persen diatas, maka dapat disimpulkan bahwa, secara perhitungan manual D2 termasuk ke dalam kelas **Negatif**.

4.2.6.3 Klasifikasi dengan *Stoplist* 20 Persen

Dari Tabel 4.24 kemudian dilakukan perhitungan probabilitas *Posterior* dengan *stoplist* 20 persen untuk setiap data pada tiap kelas sentimen. Perhitungan dilakukan dengan menggunakan rumus Naïve Bayes *Multinomial Model* pada Persamaan 2.3. Proses perhitungannya adalah seperti berikut:

$$P(\text{positif}|D1) = P(\text{positif}) * \prod_{w \in d} P(w|\text{positif})$$

$$= \frac{1}{2} \times P(\text{saya}|\text{positif}) \times P(\text{priyadi}|\text{positif}) \times P(\text{setuju}|\text{positif})$$

$$\times P(\text{ibukota}|\text{positif}) \times P(\text{dipindahkan}|\text{positif}) \times P(\text{jakarta}|\text{positif})$$

$$\times P(\text{sudah}|\text{positif}) \times P(\text{padat}|\text{positif}) \times P(\text{sekali}|\text{positif})$$

$$= \frac{1}{2} \times \frac{(0+1)}{(18+32)} \times \frac{(1+1)}{(18+32)} \times \frac{(1+1)}{(18+32)} \times \frac{(2+1)}{(18+32)} \times \frac{(2+1)}{(18+32)}$$

$$\times \frac{(0+1)}{(18+32)} \times \frac{(0+1)}{(18+32)} \times \frac{(0+1)}{(18+32)} \times \frac{(0+1)}{(18+32)}$$

$$= \frac{1}{2} \times \frac{(4)}{(50)} \times \frac{(2)}{(50)} \times \frac{(2)}{(50)} \times \frac{(3)}{(50)} \times \frac{(3)}{(50)} \times \frac{(1)}{(50)} \times \frac{(1)}{(50)} \times \frac{(1)}{(50)}$$

$$\times \frac{(1)}{(50)} = 9,216e^{-15}$$

$$P(\text{negatif}|D1) = P(\text{negatif}) * \prod_{w \in d} P(w|\text{negatif})$$

$$= \frac{1}{2} \times P(\text{saya}|\text{negatif}) \times P(\text{priyadi}|\text{negatif}) \times P(\text{setuju}|\text{negatif})$$

$$\times P(\text{ibukota}|\text{negatif}) \times P(\text{dipindahkan}|\text{negatif}) \times P(\text{jakarta}|\text{negatif})$$

$$\times P(\text{sudah}|\text{negatif}) \times P(\text{padat}|\text{negatif}) \times P(\text{sekali}|\text{negatif})$$

$$= \frac{1}{2} \times \frac{(0+1)}{(19+32)} \times \frac{(0+1)}{(19+32)} \times \frac{(0+1)}{(19+32)} \times \frac{(2+1)}{(19+32)} \times \frac{(0+1)}{(19+32)}$$

$$\times \frac{(0+1)}{(19+32)} \times \frac{(0+1)}{(19+32)} \times \frac{(0+1)}{(19+32)} \times \frac{(0+1)}{(19+32)}$$

$$= \frac{1}{2} \times \frac{(1)}{(51)} \times \frac{(1)}{(51)} \times \frac{(1)}{(51)} \times \frac{(3)}{(51)} \times \frac{(1)}{(51)} \times \frac{(1)}{(51)} \times \frac{(1)}{(51)} \times \frac{(1)}{(51)}$$

$$= 6,42628e^{-16}$$



Berdasarkan hasil perhitungan probabilitas *Posterior* dengan *stoplist* 20 persen diatas, maka dapat disimpulkan bahwa, secara perhitungan manual D1 termasuk ke dalam kelas **positif**.

$$\begin{aligned} P(\text{positif}|D2) &= P(\text{positif}) * \prod_{w \in d} P(w|\text{positif}) \\ &= \frac{1}{2} x P(\text{pokoknya}|\text{positif}) x P(\text{tidak}|\text{positif}) x P(\text{setuju}|\text{positif}) \\ &\quad x P(\text{kalau}|\text{positif}) x P(\text{ibukota}|\text{positif}) x P(\text{pindah}|\text{positif}) \\ &\quad x P(\text{semoga}|\text{positif}) x P(\text{saja}|\text{positif}) x P(\text{tidak}|\text{positif}) \\ &\quad x P(\text{jadi}|\text{positif}) x P(\text{pindah}|\text{positif}) \\ &= \frac{1}{2} x \frac{(0+1)}{(18+32)} x \frac{(0+1)}{(18+32)} x \frac{(1+1)}{(18+32)} x \frac{(0+1)}{(18+32)} x \frac{(2+1)}{(18+32)} \\ &\quad x \frac{(0+1)}{(18+32)} x \frac{(0+1)}{(18+32)} x \frac{(0+1)}{(18+32)} x \frac{(0+1)}{(18+32)} x \frac{(0+1)}{(18+32)} x \frac{(0+1)}{(18+32)} \\ &= \frac{1}{2} x \frac{(1)}{(50)} x \frac{(1)}{(50)} x \frac{(2)}{(50)} x \frac{(1)}{(50)} x \frac{(3)}{(50)} x \frac{(1)}{(50)} x \frac{(1)}{(50)} x \frac{(1)}{(50)} x \frac{(1)}{(50)} \\ &\quad x \frac{(1)}{(50)} x \frac{(1)}{(50)} = 1,536e^{-15} \end{aligned}$$

$$P(\text{negatif}|D2) = P(\text{negatif}) * \prod_{w \in d} P(w|\text{negatif})$$

$$\begin{aligned} &= \frac{1}{2} x P(\text{pokoknya}|\text{negatif}) x P(\text{tidak}|\text{negatif}) x P(\text{setuju}|\text{negatif}) \\ &\quad x P(\text{kalau}|\text{negatif}) x P(\text{ibukota}|\text{negatif}) x P(\text{pindah}|\text{negatif}) \\ &\quad x P(\text{semoga}|\text{negatif}) x P(\text{saja}|\text{negatif}) x P(\text{tidak}|\text{negatif}) \\ &\quad x P(\text{jadi}|\text{negatif}) x P(\text{pindah}|\text{negatif}) \\ &= \frac{1}{2} x \frac{(0+1)}{(19+32)} x \frac{(1+1)}{(19+32)} x \frac{(0+1)}{(19+32)} x \frac{(0+1)}{(19+32)} x \frac{(2+1)}{(19+32)} \\ &\quad x \frac{(2+1)}{(19+32)} x \frac{(1+1)}{(19+32)} x \frac{(0+1)}{(19+32)} x \frac{(1+1)}{(19+32)} x \frac{(0+1)}{(19+32)} x \frac{(2+1)}{(19+32)} \\ &= \frac{1}{2} x \frac{(1)}{(51)} x \frac{(2)}{(51)} x \frac{(1)}{(51)} x \frac{(1)}{(51)} x \frac{(3)}{(51)} x \frac{(3)}{(51)} x \frac{(2)}{(51)} x \frac{(1)}{(51)} x \frac{(2)}{(51)} \\ &\quad x \frac{(1)}{(51)} x \frac{(3)}{(51)} = 1,54231e^{-14} \end{aligned}$$



Berdasarkan hasil perhitungan probabilitas *Posterior* dengan *stoplist* 20 persen diatas, maka dapat disimpulkan bahwa, secara perhitungan manual D2 termasuk ke dalam kelas **Negatif**.

4.2.6.4 Klasifikasi dengan *Stoplist* 30 Persen

Dari Tabel 4.24 kemudian dilakukan perhitungan probabilitas *Posterior* dengan *stoplist* 30 persen untuk setiap data pada tiap kelas sentimen. Perhitungan dilakukan dengan menggunakan rumus Naïve Bayes *Multinomial Model* pada Persamaan 2.3. Proses perhitungannya adalah seperti berikut:

$$P(\text{positif}|D1) = P(\text{positif}) * \prod_{w \in d} P(w|\text{positif})$$

$$= \frac{1}{2} \times P(\text{saya}|\text{positif}) \times P(\text{priyadi}|\text{positif}) \times P(\text{setuju}|\text{positif})$$

$$\times P(\text{ibukota}|\text{positif}) \times P(\text{dipindahkan}|\text{positif}) \times P(\text{jakarta}|\text{positif})$$

$$\times P(\text{sudah}|\text{positif}) \times P(\text{padat}|\text{positif}) \times P(\text{sekali}|\text{positif})$$

$$= \frac{1}{2} \times \frac{(0+1)}{(14+28)} \times \frac{(0+1)}{(14+28)} \times \frac{(1+1)}{(14+28)} \times \frac{(2+1)}{(14+28)} \times \frac{(2+1)}{(14+28)}$$

$$\times \frac{(0+1)}{(14+28)} \times \frac{(0+1)}{(14+28)} \times \frac{(0+1)}{(14+28)} \times \frac{(0+1)}{(14+28)}$$

$$= \frac{1}{2} \times \frac{(4)}{(42)} \times \frac{(1)}{(42)} \times \frac{(2)}{(42)} \times \frac{(3)}{(42)} \times \frac{(3)}{(42)} \times \frac{(1)}{(42)} \times \frac{(1)}{(42)} \times \frac{(1)}{(42)}$$

$$\times \frac{(1)}{(42)} = 2,21309e^{-14}$$

$$P(\text{negatif}|D1) = P(\text{negatif}) * \prod_{w \in d} P(w|\text{negatif})$$

$$= \frac{1}{2} \times P(\text{saya}|\text{negatif}) \times P(\text{priyadi}|\text{negatif}) \times P(\text{setuju}|\text{negatif})$$

$$\times P(\text{ibukota}|\text{negatif}) \times P(\text{dipindahkan}|\text{negatif}) \times P(\text{jakarta}|\text{negatif})$$

$$\times P(\text{sudah}|\text{negatif}) \times P(\text{padat}|\text{negatif}) \times P(\text{sekali}|\text{negatif})$$

$$= \frac{1}{2} \times \frac{(0+1)}{(19+28)} \times \frac{(0+1)}{(19+28)} \times \frac{(0+1)}{(19+28)} \times \frac{(2+1)}{(19+28)} \times \frac{(0+1)}{(19+28)}$$

$$\times \frac{(0+1)}{(19+28)} \times \frac{(0+1)}{(19+28)} \times \frac{(0+1)}{(19+28)} \times \frac{(0+1)}{(19+28)}$$

$$= \frac{1}{2} \times \frac{(1)}{(47)} \times \frac{(1)}{(47)} \times \frac{(1)}{(47)} \times \frac{(3)}{(47)} \times \frac{(1)}{(47)} \times \frac{(1)}{(47)} \times \frac{(1)}{(47)} \times \frac{(1)}{(47)}$$

$$= 1,34033e^{-15}$$



Berdasarkan hasil perhitungan probabilitas *Posterior* dengan *stoplist* 30 persen diatas, maka dapat disimpulkan bahwa, secara perhitungan manual D1 termasuk ke dalam kelas **positif**.

$$\begin{aligned} P(\text{positif}|D2) &= P(\text{positif}) * \prod_{w \in d} P(w|\text{positif}) \\ &= \frac{1}{2} x P(\text{pokoknya}|\text{positif}) x P(\text{tidak}|\text{positif}) x P(\text{setuju}|\text{positif}) \\ &\quad x P(\text{kalau}|\text{positif}) x P(\text{ibukota}|\text{positif}) x P(\text{pindah}|\text{positif}) \\ &\quad x P(\text{semoga}|\text{positif}) x P(\text{saja}|\text{positif}) x P(\text{tidak}|\text{positif}) \\ &\quad x P(\text{jadi}|\text{positif}) x P(\text{pindah}|\text{positif}) \\ &= \frac{1}{2} x \frac{(0+1)}{(14+28)} x \frac{(0+1)}{(14+28)} x \frac{(1+1)}{(14+28)} x \frac{(0+1)}{(14+28)} x \frac{(2+1)}{(14+28)} \\ &\quad x \frac{(0+1)}{(14+28)} x \frac{(0+1)}{(14+28)} x \frac{(0+1)}{(14+28)} x \frac{(0+1)}{(14+28)} x \frac{(0+1)}{(14+28)} x \frac{(0+1)}{(14+28)} \\ &= \frac{1}{2} x \frac{(1)}{(42)} x \frac{(1)}{(42)} x \frac{(2)}{(42)} x \frac{(1)}{(42)} x \frac{(3)}{(42)} x \frac{(1)}{(42)} x \frac{(1)}{(42)} x \frac{(1)}{(42)} x \frac{(1)}{(42)} \\ &\quad x \frac{(1)}{(42)} x \frac{(1)}{(42)} = 7,37696e^{-15} \end{aligned}$$

$$P(\text{negatif}|D2) = P(\text{negatif}) * \prod_{w \in d} P(w|\text{negatif})$$

$$\begin{aligned} &= \frac{1}{2} x P(\text{pokoknya}|\text{negatif}) x P(\text{tidak}|\text{negatif}) x P(\text{setuju}|\text{negatif}) \\ &\quad x P(\text{kalau}|\text{negatif}) x P(\text{ibukota}|\text{negatif}) x P(\text{pindah}|\text{negatif}) \\ &\quad x P(\text{semoga}|\text{negatif}) x P(\text{saja}|\text{negatif}) x P(\text{tidak}|\text{negatif}) \\ &\quad x P(\text{jadi}|\text{negatif}) x P(\text{pindah}|\text{negatif}) \\ &= \frac{1}{2} x \frac{(0+1)}{(19+28)} x \frac{(1+1)}{(19+28)} x \frac{(0+1)}{(19+28)} x \frac{(0+1)}{(19+28)} x \frac{(2+1)}{(19+28)} \\ &\quad x \frac{(2+1)}{(19+28)} x \frac{(1+1)}{(19+28)} x \frac{(0+1)}{(19+28)} x \frac{(1+1)}{(19+28)} x \frac{(0+1)}{(19+28)} x \frac{(2+1)}{(19+28)} \\ &= \frac{1}{2} x \frac{(1)}{(47)} x \frac{(2)}{(47)} x \frac{(1)}{(47)} x \frac{(1)}{(47)} x \frac{(3)}{(47)} x \frac{(3)}{(47)} x \frac{(2)}{(47)} x \frac{(1)}{(47)} x \frac{(2)}{(47)} \\ &\quad x \frac{(1)}{(47)} x \frac{(3)}{(47)} = 3,21678e^{-14} \end{aligned}$$



Berdasarkan hasil perhitungan probabilitas *Posterior* dengan *stoplist* 30 persen diatas, maka dapat disimpulkan bahwa, secara perhitungan manual D2 termasuk ke dalam kelas **Negatif**.

4.2.6.5 Klasifikasi dengan *Stoplist* 40 Persen

Dari Tabel 4.24 kemudian dilakukan perhitungan probabilitas *Posterior* dengan *stoplist* 40 persen untuk setiap data pada tiap kelas sentimen. Perhitungan dilakukan dengan menggunakan rumus Naïve Bayes *Multinomial Model* pada Persamaan 2.3. Proses perhitungannya adalah seperti berikut:

$$P(\text{positif}|D1) = P(\text{positif}) * \prod_{w \in d} P(w|\text{positif})$$

$$= \frac{1}{2} \times P(\text{saya}|\text{positif}) \times P(\text{priyadi}|\text{positif}) \times P(\text{setuju}|\text{positif})$$

$$\times P(\text{ibukota}|\text{positif}) \times P(\text{dipindahkan}|\text{positif}) \times P(\text{jakarta}|\text{positif})$$

$$\times P(\text{sudah}|\text{positif}) \times P(\text{padat}|\text{positif}) \times P(\text{sekali}|\text{positif})$$

$$= \frac{1}{2} \times \frac{(0+1)}{(14+24)} \times \frac{(0+1)}{(14+24)} \times \frac{(1+1)}{(14+24)} \times \frac{(2+1)}{(14+24)} \times \frac{(2+1)}{(14+24)}$$

$$\times \frac{(0+1)}{(14+24)} \times \frac{(0+1)}{(14+24)} \times \frac{(0+1)}{(14+24)} \times \frac{(0+1)}{(14+24)}$$

$$= \frac{1}{2} \times \frac{(4)}{(38)} \times \frac{(1)}{(38)} \times \frac{(2)}{(38)} \times \frac{(3)}{(38)} \times \frac{(3)}{(38)} \times \frac{(1)}{(38)} \times \frac{(1)}{(38)} \times \frac{(1)}{(38)}$$

$$\times \frac{(1)}{(38)} = 5,44741e^{-14}$$

$$P(\text{negatif}|D1) = P(\text{negatif}) * \prod_{w \in d} P(w|\text{negatif})$$

$$= \frac{1}{2} \times P(\text{saya}|\text{negatif}) \times P(\text{priyadi}|\text{negatif}) \times P(\text{setuju}|\text{negatif})$$

$$\times P(\text{ibukota}|\text{negatif}) \times P(\text{dipindahkan}|\text{negatif}) \times P(\text{jakarta}|\text{negatif})$$

$$\times P(\text{sudah}|\text{negatif}) \times P(\text{padat}|\text{negatif}) \times P(\text{sekali}|\text{negatif})$$

$$= \frac{1}{2} \times \frac{(0+1)}{(15+24)} \times \frac{(0+1)}{(15+24)} \times \frac{(0+1)}{(15+24)} \times \frac{(2+1)}{(15+24)} \times \frac{(0+1)}{(15+24)}$$

$$\times \frac{(0+1)}{(15+24)} \times \frac{(0+1)}{(15+24)} \times \frac{(0+1)}{(15+24)} \times \frac{(0+1)}{(15+24)}$$

$$= \frac{1}{2} \times \frac{(1)}{(39)} \times \frac{(1)}{(39)} \times \frac{(1)}{(39)} \times \frac{(3)}{(39)} \times \frac{(1)}{(39)} \times \frac{(1)}{(39)} \times \frac{(1)}{(39)} \times \frac{(1)}{(39)}$$

$$= 4,31182e^{-14}$$



Berdasarkan hasil perhitungan probabilitas *Posterior* dengan *stoplist* 40 persen diatas, maka dapat disimpulkan bahwa, secara perhitungan manual D1 termasuk ke dalam kelas **positif**.

$$\begin{aligned} P(\text{positif}|D2) &= P(\text{positif}) * \prod_{w \in d} P(w|\text{positif}) \\ &= \frac{1}{2} x P(\text{pokoknya}|\text{positif}) x P(\text{tidak}|\text{positif}) x P(\text{setuju}|\text{positif}) \\ &\quad x P(\text{kalau}|\text{positif}) x P(\text{ibukota}|\text{positif}) x P(\text{pindah}|\text{positif}) \\ &\quad x P(\text{semoga}|\text{positif}) x P(\text{saja}|\text{positif}) x P(\text{tidak}|\text{positif}) \\ &\quad x P(\text{jadi}|\text{positif}) x P(\text{pindah}|\text{positif}) \\ &= \frac{1}{2} x \frac{(0+1)}{(14+24)} x \frac{(0+1)}{(14+24)} x \frac{(1+1)}{(14+24)} x \frac{(0+1)}{(14+24)} x \frac{(2+1)}{(14+24)} \\ &\quad x \frac{(0+1)}{(14+24)} x \frac{(0+1)}{(14+24)} x \frac{(0+1)}{(14+24)} x \frac{(0+1)}{(14+24)} x \frac{(0+1)}{(14+24)} x \frac{(0+1)}{(14+24)} \\ &= \frac{1}{2} x \frac{(1)}{(38)} x \frac{(1)}{(38)} x \frac{(2)}{(38)} x \frac{(1)}{(38)} x \frac{(3)}{(38)} x \frac{(1)}{(38)} x \frac{(1)}{(38)} x \frac{(1)}{(38)} x \frac{(1)}{(38)} \\ &\quad x \frac{(1)}{(38)} x \frac{(1)}{(38)} = 1,8158e^{-14} \end{aligned}$$

$$P(\text{negatif}|D2) = P(\text{negatif}) * \prod_{w \in d} P(w|\text{negatif})$$

$$\begin{aligned} &= \frac{1}{2} x P(\text{pokoknya}|\text{negatif}) x P(\text{tidak}|\text{negatif}) x P(\text{setuju}|\text{negatif}) \\ &\quad x P(\text{kalau}|\text{negatif}) x P(\text{ibukota}|\text{negatif}) x P(\text{pindah}|\text{negatif}) \\ &\quad x P(\text{semoga}|\text{negatif}) x P(\text{saja}|\text{negatif}) x P(\text{tidak}|\text{negatif}) \\ &\quad x P(\text{jadi}|\text{negatif}) x P(\text{pindah}|\text{negatif}) \\ &= \frac{1}{2} x \frac{(0+1)}{(15+24)} x \frac{(0+1)}{(15+24)} x \frac{(0+1)}{(15+24)} x \frac{(0+1)}{(15+24)} x \frac{(2+1)}{(15+24)} \\ &\quad x \frac{(2+1)}{(15+24)} x \frac{(0+1)}{(15+24)} x \frac{(0+1)}{(15+24)} x \frac{(0+1)}{(15+24)} x \frac{(0+1)}{(15+24)} x \frac{(2+1)}{(15+24)} \\ &= \frac{1}{2} x \frac{(1)}{(39)} x \frac{(1)}{(39)} x \frac{(1)}{(39)} x \frac{(3)}{(39)} x \frac{(3)}{(39)} x \frac{(1)}{(39)} x \frac{(1)}{(39)} x \frac{(1)}{(39)} \\ &\quad x \frac{(1)}{(39)} x \frac{(3)}{(39)} = 1,43727e^{-14} \end{aligned}$$



Berdasarkan hasil perhitungan probabilitas *Posterior* dengan *stoplist* 40 persen diatas, maka dapat disimpulkan bahwa, secara perhitungan manual D2 termasuk ke dalam kelas **Positif**.

4.2.6.6 Klasifikasi dengan *Stoplist* 50 Persen

Dari Tabel 4.24 kemudian dilakukan perhitungan probabilitas *Posterior* dengan *stoplist* 50 persen untuk setiap data pada tiap kelas sentimen. Perhitungan dilakukan dengan menggunakan rumus Naïve Bayes *Multinomial Model* pada Persamaan 2.3. Proses perhitungannya adalah seperti berikut:

$$\begin{aligned} P(\text{positif}|D1) &= P(\text{positif}) * \prod_{w \in d} P(w|\text{positif}) \\ &= \frac{1}{2} \times P(\text{saya}|\text{positif}) \times P(\text{priyadi}|\text{positif}) \times P(\text{setuju}|\text{positif}) \\ &\quad \times P(\text{ibukota}|\text{positif}) \times P(\text{dipindahkan}|\text{positif}) \times P(\text{jakarta}|\text{positif}) \\ &\quad \times P(\text{sudah}|\text{positif}) \times P(\text{padat}|\text{positif}) \times P(\text{sekali}|\text{positif}) \\ &= \frac{1}{2} \times \frac{(0+1)}{(14+20)} \times \frac{(0+1)}{(14+20)} \times \frac{(1+1)}{(14+20)} \times \frac{(2+1)}{(14+20)} \times \frac{(2+1)}{(14+20)} \\ &\quad \times \frac{(0+1)}{(14+20)} \times \frac{(0+1)}{(14+20)} \times \frac{(0+1)}{(14+20)} \times \frac{(0+1)}{(14+20)} \\ &= \frac{1}{2} \times \frac{(4)}{(34)} \times \frac{(1)}{(34)} \times \frac{(2)}{(34)} \times \frac{(3)}{(34)} \times \frac{(3)}{(34)} \times \frac{(1)}{(34)} \times \frac{(1)}{(34)} \times \frac{(1)}{(34)} \\ &\quad \times \frac{(1)}{(34)} = 1,48229e^{-13} \end{aligned}$$

$$P(\text{negatif}|D1) = P(\text{negatif}) * \prod_{w \in d} P(w|\text{negatif})$$

$$\begin{aligned} &\bar{=} \frac{1}{2} \times P(\text{saya}|\text{negatif}) \times P(\text{priyadi}|\text{negatif}) \times P(\text{setuju}|\text{negatif}) \\ &\quad \times P(\text{ibukota}|\text{negatif}) \times P(\text{dipindahkan}|\text{negatif}) \times P(\text{jakarta}|\text{negatif}) \\ &\quad \times P(\text{sudah}|\text{negatif}) \times P(\text{padat}|\text{negatif}) \times P(\text{sekali}|\text{negatif}) \\ &= \frac{1}{2} \times \frac{(0+1)}{(11+20)} \times \frac{(0+1)}{(11+20)} \times \frac{(0+1)}{(11+20)} \times \frac{(2+1)}{(11+20)} \times \frac{(0+1)}{(11+20)} \\ &\quad \times \frac{(0+1)}{(11+20)} \times \frac{(0+1)}{(11+20)} \times \frac{(0+1)}{(11+20)} \times \frac{(0+1)}{(11+20)} \\ &= \frac{1}{2} \times \frac{(1)}{(31)} \times \frac{(1)}{(31)} \times \frac{(1)}{(31)} \times \frac{(3)}{(31)} \times \frac{(1)}{(31)} \times \frac{(1)}{(31)} \times \frac{(1)}{(31)} \times \frac{(1)}{(31)} \\ &\quad \times \frac{(1)}{(31)} = 3,40398e^{-13} \end{aligned}$$



Berdasarkan hasil perhitungan probabilitas *Posterior* dengan *stoplist* 50 persen diatas, maka dapat disimpulkan bahwa, secara perhitungan manual D1 termasuk ke dalam kelas **Negatif**.

$$\begin{aligned} P(\text{positif}|D2) &= P(\text{positif}) * \prod_{w \in d} P(w|\text{positif}) \\ &= \frac{1}{2} x P(\text{pokoknya}|\text{positif}) x P(\text{tidak}|\text{positif}) x P(\text{setuju}|\text{positif}) \\ &\quad x P(\text{kalau}|\text{positif}) x P(\text{ibukota}|\text{positif}) x P(\text{pindah}|\text{positif}) \\ &\quad x P(\text{semoga}|\text{positif}) x P(\text{saja}|\text{positif}) x P(\text{tidak}|\text{positif}) \\ &\quad x P(\text{jadi}|\text{positif}) x P(\text{pindah}|\text{positif}) \\ &= \frac{1}{2} x \frac{(0+1)}{(14+20)} x \frac{(0+1)}{(14+20)} x \frac{(1+1)}{(14+20)} x \frac{(0+1)}{(14+20)} x \frac{(2+1)}{(14+20)} \\ &\quad x \frac{(0+1)}{(14+20)} x \frac{(0+1)}{(14+20)} x \frac{(0+1)}{(14+20)} x \frac{(0+1)}{(14+20)} x \frac{(0+1)}{(14+20)} x \frac{(0+1)}{(14+20)} \\ &= \frac{1}{2} x \frac{(1)}{(34)} x \frac{(1)}{(34)} x \frac{(2)}{(34)} x \frac{(1)}{(34)} x \frac{(3)}{(34)} x \frac{(1)}{(34)} x \frac{(1)}{(34)} x \frac{(1)}{(34)} x \frac{(1)}{(34)} \\ &\quad x \frac{(1)}{(34)} x \frac{(1)}{(34)} = 4,94096e^{-14} \end{aligned}$$

$$P(\text{negatif}|D2) = P(\text{negatif}) * \prod_{w \in d} P(w|\text{negatif})$$

$$\begin{aligned} &= \frac{1}{2} x P(\text{pokoknya}|\text{negatif}) x P(\text{tidak}|\text{negatif}) x P(\text{setuju}|\text{negatif}) \\ &\quad x P(\text{kalau}|\text{negatif}) x P(\text{ibukota}|\text{negatif}) x P(\text{pindah}|\text{negatif}) \\ &\quad x P(\text{semoga}|\text{negatif}) x P(\text{saja}|\text{negatif}) x P(\text{tidak}|\text{negatif}) \\ &\quad x P(\text{jadi}|\text{negatif}) x P(\text{pindah}|\text{negatif}) \\ &= \frac{1}{2} x \frac{(0+1)}{(11+20)} x \frac{(0+1)}{(11+20)} x \frac{(0+1)}{(11+20)} x \frac{(0+1)}{(11+20)} x \frac{(2+1)}{(11+20)} \\ &\quad x \frac{(2+1)}{(11+20)} x \frac{(0+1)}{(11+20)} x \frac{(0+1)}{(11+20)} x \frac{(0+1)}{(11+20)} x \frac{(0+1)}{(11+20)} x \frac{(2+1)}{(11+20)} \\ &= \frac{1}{2} x \frac{(1)}{(31)} x \frac{(1)}{(31)} x \frac{(1)}{(31)} x \frac{(3)}{(31)} x \frac{(3)}{(31)} x \frac{(1)}{(31)} x \frac{(1)}{(31)} x \frac{(1)}{(31)} \\ &\quad x \frac{(1)}{(31)} x \frac{(3)}{(31)} = 1,13466e^{-13} \end{aligned}$$



Berdasarkan hasil perhitungan probabilitas *Posterior* dengan *stoplist* 50 persen diatas, maka dapat disimpulkan bahwa, secara perhitungan manual D2 termasuk ke dalam kelas **Negatif**.

4.2.6.7 Klasifikasi dengan *Stoplist* 60 Persen

Dari Tabel 4.24 kemudian dilakukan perhitungan probabilitas *Posterior* dengan *stoplist* 60 persen untuk setiap data pada tiap kelas sentimen. Perhitungan dilakukan dengan menggunakan rumus Naïve Bayes *Multinomial Model* pada Persamaan 2.3. Proses perhitungannya adalah seperti berikut:

$$P(\text{positif}|D1) = P(\text{positif}) * \prod_{w \in d} P(w|\text{positif})$$

$$= \frac{1}{2} \times P(\text{saya}|\text{positif}) \times P(\text{priyadi}|\text{positif}) \times P(\text{setuju}|\text{positif})$$

$$\times P(\text{ibukota}|\text{positif}) \times P(\text{dipindahkan}|\text{positif}) \times P(\text{jakarta}|\text{positif})$$

$$\times P(\text{sudah}|\text{positif}) \times P(\text{padat}|\text{positif}) \times P(\text{sekali}|\text{positif})$$

$$= \frac{1}{2} \times \frac{(0+1)}{(14+16)} \times \frac{(0+1)}{(14+16)} \times \frac{(1+1)}{(14+16)} \times \frac{(2+1)}{(14+16)} \times \frac{(2+1)}{(14+16)}$$

$$\times \frac{(0+1)}{(14+16)} \times \frac{(0+1)}{(14+16)} \times \frac{(0+1)}{(14+16)} \times \frac{(0+1)}{(14+16)}$$

$$= \frac{1}{2} \times \frac{(4)}{(30)} \times \frac{(1)}{(30)} \times \frac{(2)}{(30)} \times \frac{(3)}{(30)} \times \frac{(3)}{(30)} \times \frac{(1)}{(30)} \times \frac{(1)}{(30)} \times \frac{(1)}{(30)}$$

$$\times \frac{(1)}{(30)} = 4,57247e^{-13}$$

$$P(\text{negatif}|D1) = P(\text{negatif}) * \prod_{w \in d} P(w|\text{negatif})$$

$$= \frac{1}{2} \times P(\text{saya}|\text{negatif}) \times P(\text{priyadi}|\text{negatif}) \times P(\text{setuju}|\text{negatif})$$

$$\times P(\text{ibukota}|\text{negatif}) \times P(\text{dipindahkan}|\text{negatif}) \times P(\text{jakarta}|\text{negatif})$$

$$\times P(\text{sudah}|\text{negatif}) \times P(\text{padat}|\text{negatif}) \times P(\text{sekali}|\text{negatif})$$

$$= \frac{1}{2} \times \frac{(0+1)}{(7+16)} \times \frac{(0+1)}{(7+16)} \times \frac{(0+1)}{(7+16)} \times \frac{(2+1)}{(7+16)} \times \frac{(0+1)}{(7+16)}$$

$$\times \frac{(0+1)}{(7+16)} \times \frac{(0+1)}{(7+16)} \times \frac{(0+1)}{(7+16)} \times \frac{(0+1)}{(7+16)}$$

$$= \frac{1}{2} \times \frac{(1)}{(23)} \times \frac{(1)}{(23)} \times \frac{(1)}{(23)} \times \frac{(3)}{(23)} \times \frac{(1)}{(23)} \times \frac{(1)}{(23)} \times \frac{(1)}{(23)} \times \frac{(1)}{(23)}$$

$$= 4,9968e^{-12}$$



Berdasarkan hasil perhitungan probabilitas *Posterior* dengan *stoplist* 60 persen diatas, maka dapat disimpulkan bahwa, secara perhitungan manual D1 termasuk ke dalam kelas **Negatif**.

$$\begin{aligned} P(\text{positif}|D2) &= P(\text{positif}) * \prod_{w \in d} P(w|\text{positif}) \\ &= \frac{1}{2} x P(\text{pokoknya}|\text{positif}) x P(\text{tidak}|\text{positif}) x P(\text{setuju}|\text{positif}) \\ &\quad x P(\text{kalau}|\text{positif}) x P(\text{ibukota}|\text{positif}) x P(\text{pindah}|\text{positif}) \\ &\quad x P(\text{semoga}|\text{positif}) x P(\text{saja}|\text{positif}) x P(\text{tidak}|\text{positif}) \\ &\quad x P(\text{jadi}|\text{positif}) x P(\text{pindah}|\text{positif}) \\ &= \frac{1}{2} x \frac{(0+1)}{(14+16)} x \frac{(0+1)}{(14+16)} x \frac{(1+1)}{(14+16)} x \frac{(0+1)}{(14+16)} x \frac{(2+1)}{(14+16)} \\ &\quad x \frac{(0+1)}{(14+16)} x \frac{(0+1)}{(14+16)} x \frac{(0+1)}{(14+16)} x \frac{(0+1)}{(14+16)} x \frac{(0+1)}{(14+16)} x \frac{(0+1)}{(14+16)} \\ &= \frac{1}{2} x \frac{(1)}{(30)} x \frac{(1)}{(30)} x \frac{(2)}{(30)} x \frac{(1)}{(30)} x \frac{(3)}{(30)} x \frac{(1)}{(30)} x \frac{(1)}{(30)} x \frac{(1)}{(30)} x \frac{(1)}{(30)} \\ &\quad x \frac{(1)}{(30)} x \frac{(1)}{(30)} = 1,52416e^{-13} \end{aligned}$$

$$P(\text{negatif}|D2) = P(\text{negatif}) * \prod_{w \in d} P(w|\text{negatif})$$

$$\begin{aligned} &= \frac{1}{2} x P(\text{pokoknya}|\text{negatif}) x P(\text{tidak}|\text{negatif}) x P(\text{setuju}|\text{negatif}) \\ &\quad x P(\text{kalau}|\text{negatif}) x P(\text{ibukota}|\text{negatif}) x P(\text{pindah}|\text{negatif}) \\ &\quad x P(\text{semoga}|\text{negatif}) x P(\text{saja}|\text{negatif}) x P(\text{tidak}|\text{negatif}) \\ &\quad x P(\text{jadi}|\text{negatif}) x P(\text{pindah}|\text{negatif}) \\ &= \frac{1}{2} x \frac{(0+1)}{(7+16)} x \frac{(0+1)}{(7+16)} x \frac{(0+1)}{(7+16)} x \frac{(0+1)}{(7+16)} x \frac{(2+1)}{(7+16)} \\ &\quad x \frac{(2+1)}{(7+16)} x \frac{(0+1)}{(7+16)} x \frac{(0+1)}{(7+16)} x \frac{(0+1)}{(7+16)} x \frac{(0+1)}{(7+16)} x \frac{(2+1)}{(7+16)} \\ &= \frac{1}{2} x \frac{(1)}{(23)} x \frac{(1)}{(23)} x \frac{(1)}{(23)} x \frac{(1)}{(23)} x \frac{(3)}{(23)} x \frac{(3)}{(23)} x \frac{(1)}{(23)} x \frac{(1)}{(23)} x \frac{(1)}{(23)} \\ &\quad x \frac{(1)}{(23)} x \frac{(3)}{(23)} = 1,6656e^{-12} \end{aligned}$$



Berdasarkan hasil perhitungan probabilitas *Posterior* dengan *stoplist* 60 persen diatas, maka dapat disimpulkan bahwa, secara perhitungan manual D2 termasuk ke dalam kelas **Negatif**.

4.2.7 Hasil Analisis Sentimen

Setelah semua data uji berhasil diproses oleh sistem, maka setiap data uji akan mendapatkan label hasil klasifikasi. Hasil klasifikasi tiap data ditentukan oleh nilai *probability* terbesar diantara semua nilai *probability* di tiap kelas pada setiap data uji. Hasil klasifikasi data uji kemudian dapat digambarkan dalam bentuk grafik untuk keperluan analisis lebih lanjut, misalnya untuk mengetahui sentimen mayoritas masyarakat terhadap suatu kebijakan yang dicanangkan oleh pemerintah, menyimpulkan sentimen konsumen terhadap suatu produk, mengetahui sentimen penonton terhadap suatu film, dan lain-lain.

4.3 Perancangan Pengujian

Pengujian dilakukan untuk mengukur atau menilai seberapa baik kinerja sistem. Pada penelitian ini, *Confusion Matrix* jenis *multiclass* digunakan untuk membantu proses pengujian. Dalam perhitungan manual digunakan 2 data uji yang telah diberi label. Hasil label dari *ground truth* dan hasil perhitungan manual kemudian digunakan untuk membuat *confusion matrix*. Setelah *confusion matrix* terbuat, kemudian dilakukan perhitungan nilai parameter evaluasi dari sistem. Parameter evaluasi yang digunakan yaitu akurasi, *precision* dan *recall*. Berikut adalah *Confusion Matrix* dari proses klasifikasi dengan *stoplist* 0 persen, 10 persen, 20 persen dan 30 persen yang ditampilkan pada Tabel 4.25.

Tabel 4.25 Confusion Matrix Klasifikasi dengan Stoplist 0 persen, 10 persen, 20 persen dan 30 persen

Confusion Matrix		Prediksi	
		Positif	Negatif
Aktual	Positif	1	0
	Negatif	0	1

Berdasarkan Persamaan 2.5, Persamaan 2.6 dan Persamaan 2.7 dapat dihitung nilai FP, FN dan TN pada tiap kelas:

$$FP(\text{positif}) = \sum_i^l E_{li} - TP_i = (1 + 0) - 1 = 0$$

$$FN(\text{positif}) = \sum_i^l E_{il} - TP_i = (1 + 0) - 1 = 0$$



$$TN(\text{positif}) = \sum_i^l \sum_k^l E_{lk} - TP_i - FP_i - FN_i$$
$$= (1 + 0 + 1 + 0) - 1 - 0 - 1 = 0$$

Perhitungan evaluasinya sebagai berikut:

$$\text{akurasi(positif)} = \frac{1 + 1}{1 + 0 + 0 + 1} = 1$$

$$\text{precision(positif)} = \frac{1}{1 + 0} = 1$$

$$\text{recall(positif)} = \frac{1}{1 + 0} = 1$$

$$F - \text{measure(positif)} = 2 \times \frac{(1 \times 1)}{(1 + 1)} = 1$$

Perhitungan FP, FN dan TN untuk kelas negatif:

$$FP(\text{negatif}) = \sum_i^l E_{li} - TP_i = (0 + 1) - 1 = 0$$

$$FN(\text{negatif}) = \sum_i^l E_{il} - TP_i = (0 + 1) - 1 = 0$$

$$TN(\text{negatif}) = \sum_i^l \sum_k^l E_{lk} - TP_i - FP_i - FN_i$$
$$= (1 + 0 + 0 + 1) - 1 - 0 - 0 = 1$$

Perhitungan evaluasinya sebagai berikut:

$$\text{akurasi(negatif)} = \frac{1 + 1}{1 + 0 + 0 + 1} = 1$$

$$\text{precision(negatif)} = \frac{1}{1 + 0} = 1$$



$$\text{recall(negatif)} = \frac{1}{1+0} = 1$$

$$F - \text{measure(negatif)} = 2 \times \frac{(1 \times 1)}{(1+1)} = 1$$

Perhitungan *macroaverage* parameter evaluasi seluruh kelas:

$$\text{akurasi} = \frac{1+1}{2} = 1$$

$$\text{precision} = \frac{1+1}{2} = 1$$

$$\text{recall} = \frac{1+1}{2} = 1$$

$$F - \text{measure} = \frac{1+1}{2} = 1$$

Sementara itu, hasil klasifikasi dengan menggunakan *stoplist* 40 persen memiliki hasil yang berbeda dengan hasil klasifikasi dengan menggunakan *stoplist* 0 hingga 30 persen. Saat digunakan *stoplist* 40 persen, terjadi kesalahan klasifikasi data uji berkelas negatif yang diklasifikasikan ke dalam kelas positif berdasarkan hasil perhitungan manual. Tabel 4.26 menunjukkan *confusion matrix* dari hasil klasifikasi dengan menggunakan *stoplist* 40 persen.

Tabel 4.26 Confusion Matrix Klasifikasi dengan *Stoplist* 40 persen

<i>Confusion Matrix</i>		<i>Prediksi</i>	
		<i>Positif</i>	<i>Negatif</i>
<i>Aktual</i>	<i>Positif</i>	1	0
	<i>Negatif</i>	1	0

Berdasarkan Persamaan 2.5, Persamaan 2.6 dan Persamaan 2.7 dapat dihitung nilai FP, FN dan TN pada tiap kelas:

$$FP(\text{positif}) = \sum_i E_{ii} - TP_i = (1+1) - 1 = 0$$



$$FN(\text{positif}) = \sum_i^l E_{il} - TP_i = (1 + 0) - 1 = 1$$

$$TN(\text{positif}) = \sum_i^l \sum_k^l E_{lk} - TP_i - FP_i - FN_i = (1 + 0 + 1 + 0) - 1 - 0 - 1 = 0$$

Perhitungan evaluasinya sebagai berikut:

$$\text{akurasi(positif)} = \frac{1 + 0}{1 + 0 + 1 + 0} = 0,5$$

$$\text{precision(positif)} = \frac{1}{1 + 0} = 1$$

$$\text{recall(positif)} = \frac{1}{1 + 1} = 0,5$$

$$F\text{-measure(positif)} = 2 \times \frac{(1 \times 0,5)}{(1 + 0,5)} = 0,66$$

Perhitungan FP, FN dan TN untuk kelas negatif:

$$FP(\text{negatif}) = \sum_i^l E_{li} - TP_i = (0 + 0) - 0 = 0$$

$$FN(\text{negatif}) = \sum_i^l E_{il} - TP_i = (0 + 1) - 0 = 1$$

$$TN(\text{negatif}) = \sum_i^l \sum_k^l E_{lk} - TP_i - FP_i - FN_i$$

$$= (1 + 0 + 1 + 0) - 0 - 0 - 1 = 1$$

Perhitungan evaluasinya sebagai berikut:

$$\text{akurasi(negatif)} = \frac{1 + 0}{1 + 0 + 1 + 0} = 0,5$$

$$precision(negatif) = \frac{0}{0+0} = 0$$

$$recall(negatif) = \frac{0}{0+1} = 0$$

$$F - measure(negatif) = 2 \times \frac{(0 \times 0)}{(0+0)} = 0$$

Perhitungan *macroaverage* parameter evaluasi seluruh kelas:

$$akurasi = \frac{0,5 + 0,5}{2} = 0,5$$

$$precision = \frac{1+0}{2} = 0,5$$

$$recall = \frac{0,5 + 0}{2} = 0,25$$

$$F - measure = \frac{0,66 + 0}{2} = 0,33$$

Hasil perhitungan manual klasifikasi dengan menggunakan *stoplist* 50 persen dan 60 persen juga menghasilkan kesalahan klasifikasi. Terdapat data uji kelas positif yang diklasifikasikan ke dalam kelas negatif berdasarkan hasil perhitungan manual klasifikasi dengan menggunakan *stoplist* 50 persen dan 60 persen. Tabel 4.27 menunjukkan *confusion matrix* dari hasil klasifikasi dengan menggunakan *stoplist* 50 persen dan 60 persen.

Tabel 4.27 Confusion Matrix Klasifikasi dengan *Stoplist 50 persen dan 60 persen*

		Prediksi	
Confusion Matrix		Positif	Negatif
Aktual	Positif	0	1
	Negatif	0	1

Berdasarkan Persamaan 2.5, Persamaan 2.6 dan Persamaan 2.7 dapat dihitung nilai FP, FN dan TN pada tiap kelas:



$$FP(\text{positif}) = \sum_i E_{li} - TP_i = (0 + 0) - 0 = 0$$

$$FN(\text{positif}) = \sum_i E_{il} - TP_i = (0 + 1) - 0 = 1$$

$$TN(\text{positif}) = \sum_i \sum_k E_{lk} - TP_i - FP_i - FN_i$$

$$= (0 + 1 + 0 + 1) - 0 - 0 - 1 = 1$$

Perhitungan evaluasinya sebagai berikut:

$$\text{akurasi(positif)} = \frac{0 + 1}{1 + 0 + 1 + 0} = 0,5$$

$$\text{precision(positif)} = \frac{0}{0 + 0} = 0$$

$$\text{recall(positif)} = \frac{0}{0 + 1} = 0$$

$$F\text{-measure(positif)} = 2 \times \frac{(0 \times 0)}{(0 + 0)} = 0$$

Perhitungan FP, FN dan TN untuk kelas negatif:

$$FP(\text{negatif}) = \sum_i E_{li} - TP_i = (1 + 1) - 1 = 1$$

$$FN(\text{negatif}) = \sum_i E_{il} - TP_i = (0 + 1) - 1 = 0$$

$$TN(\text{negatif}) = \sum_i \sum_k E_{lk} - TP_i - FP_i - FN_i$$

$$= (0 + 1 + 0 + 1) - 1 - 1 - 0 = 0$$

Perhitungan evaluasinya sebagai berikut:

$$\text{akurasi(negatif)} = \frac{1 + 0}{1 + 0 + 1 + 0} = 0,5$$



$$\text{precision(negatif)} = \frac{1}{1+1} = 0,5$$

$$\text{recall(negatif)} = \frac{1}{1+0} = 1$$

$$F - \text{measure(negatif)} = 2 \times \frac{(0,5 \times 1)}{(0,5 + 1)} = 0,66$$

Perhitungan *macroaverage* parameter evaluasi seluruh kelas:

$$\text{akurasi} = \frac{0,5 + 0,5}{2} = 0,5$$

$$\text{precision} = \frac{0 + 0,5}{2} = 0,25$$

$$\text{recall} = \frac{0 + 1}{2} = 0,5$$

$$F - \text{measure} = \frac{0 + 0,66}{2} = 0,33$$

Dari perhitungan manual pengujian dapat dilihat bahwa klasifikasi yang dilakukan dengan menggunakan *stoplist* 0 persen hingga 30 persen memiliki nilai evaluasi yang lebih baik dari klasifikasi yang dilakukan menggunakan *stoplist* 40 persen. Sementara itu, klasifikasi yang dilakukan dengan menggunakan *stoplist* 50 persen dan 60 persen menghasilkan nilai evaluasi yang yang lebih buruk dari klasifikasi yang dilakukan menggunakan *stoplist* 40 persen. Oleh karena itu, dapat disimpulkan bahwa klasifikasi menghasilkan nilai evaluasi terbaik dengan menggunakan *stoplist* 0 hingga 30 persen.



BAB 5 IMPLEMENTASI

5.1 Implementasi Kode Program

Setelah dilakukan perancangan sistem, kemudian dilakukan implementasi perancangan ke dalam kode program. Hasil dari implementasi ini membentuk sistem analisis sentimen yang digunakan untuk menganalisis kebijakan pemindahan Ibukota dari sudut pandang warganet media sosial Twitter. Pada subbab dibawah ini akan ditampilkan kumpulan kode program yang digunakan dalam sistem.

5.1.1 Kode Program Void Main

Pada kode program ini dilakukan pemanggilan fungsi-fungsi yang telah dibuat untuk melakukan proses analisis sentimen. Dalam kode program ini juga dilakukan proses mencetak output hasil analisis sentimen dan juga hasil evaluasi sistem.

Pada Kode Program 5.1 ditampilkan kode program untuk *void main*.

Algoritma 1: Void Main

```
1 =====main process=====
2 #data training untuk label positif, negatif dan neutral
3 berjumlah sama, maka nilai prob. prior tiap label pun sama
4 prior_pos, prior_neg = 1/2, 1/2
5 #Inisialisasi data training untuk membentuk stoplist
6 data_pos = file('positif.txt')
7 data_neg = file('negatif.txt')
8 token_pos = token(data_pos)
9 token_neg = token(data_neg)
10 token_keseluruhan = []
11 for i in range(len(token_neg)):
12     token_keseluruhan.append(token_pos[i])
13     token_keseluruhan.append(token_neg[i])
14 stoplist = tbrs(token_keseluruhan,
15 (len(data_pos)+len(data_neg)), 30)
16 =====perhitungan TF pada data latih di tiap kelas=====
17 #Data Positif
18 hasil_filter_pos = filtering(token_pos, stoplist)
19 type_pos = typee(hasil_filter_pos, len(data_pos))
20 term_pos = term_(type_pos)
21 rtf_pos = raw_term_frequency(term_pos, hasil_filter_pos,
22 len(data_pos))
23 #Data Negatif
24 hasil_filter_neg = filtering(token_neg, stoplist)
25 type_neg = typee(hasil_filter_neg, len(data_neg))
26 term_neg = term_(type_neg)
27 rtf_neg = raw_term_frequency(term_neg, hasil_filter_neg,
28 len(data_neg))
29 =====Testing=====
30 data_uji = file('data_uji.txt')
31 token_uji = token(data_uji)
32 hasil_filter_uji = filtering(token_uji, stoplist)
33 type_uji = typee(hasil_filter_uji, len(data_uji))
34 nb_pos = likelihood(type_uji, term_pos, rtf_pos)
35 nb_neg = likelihood(type_uji, term_neg, rtf_neg)
36 nb_pos = naive bayes(nb_pos, prior_pos)
```

```
37 nb_neg = naive_bayes(nb_neg, prior_neg)
38 #=====penentuan kelas data uji=====
39 best_value = []
40 for i in range(len(nb_pos)):
41     best_value.append(max(nb_pos[i], nb_neg[i]))
42 hasil_analisis = []
43 for i in range(len(best_value)):
44     if(best_value[i] == nb_pos[i]):
45         hasil_analisis.append('positif')
46     else:
47         hasil_analisis.append('negatif')
48 print('hasil penentuan kelas data uji:\n', hasil_analisis)
49 #=====mencetak hasil analisis sentimen=====
50 print_hasil (data_uji, 'positif', hasil_analisis)
51 print_hasil (data_uji, 'negatif', hasil_analisis)
52 #=====evaluasi sistem=====
53 label_data_uji = buka_file('label_uji.txt')
54 TN, FP, FN, TP = 0, 0, 0, 0
55 for i in range(len(label_data_uji)):
56     if (label_data_uji[i] == 'positif'):
57         if (hasil_analisis[i] == 'positif'):
58             TP += 1
59         elif (hasil_analisis[i] == 'negatif'):
60             FN += 1
61     elif (label_data_uji[i] == 'negatif'):
62         if (hasil_analisis[i] == 'positif'):
63             FP += 1
64         elif (hasil_analisis[i] == 'negatif'):
65             TN += 1
66 akurasi = (TP + TN) / (TP + TN + FP + FN)
67 precision = TP / (TP + FP)
68 recall = TP / (TP + FN)
69 print('True Positive = ', TP, '\nFalse Positive = ', FP, '\nTrue Negative = ', TN, '\nFalse Negative = ', FN)
70 print('\nHasil evaluasi sistem :\n\n1. Akurasi : ', akurasi, '\n2. Precision : ', precision, '\n3. Recall : ', recall)
```

Kode Program 5.1 Kode Program Void Main

5.1.2 Kode Program Import Library

Pada kode program ini dilakukan proses pemanggilan *library* oleh sistem untuk digunakan pada proses selanjutnya. Terdapat beberapa *library* yang dibutuhkan oleh sistem antara lain:

1. *math*, digunakan untuk proses perhitungan matematika.
2. *nltk*, digunakan dalam proses tokenisasi.
3. *np*, digunakan pada proses *cleaning*.
4. *collections*, digunakan untuk proses pembuatan *stoplist* pada *Term-Based Random Sampling*.
5. *random*, digunakan untuk memperoleh angka acak yang digunakan dalam proses *random sampling* pada *Term-Based Random Sampling*.

6. *operator*, digunakan untuk memperoleh nilai dari suatu *array* pada proses pembuatan *stoplist*.

Pada Kode Program 5.2 ditunjukkan kode program yang digunakan oleh sistem untuk memanggil *library* yang digunakan.

Algoritma 2: Import Library

```
1 import math
2 import nltk
3 import numpy as np
4 from collections import Counter, defaultdict
5 import re
6 from math import log
7 import random
8 from operator import itemgetter
```

Kode Program 5.2 Import Library

5.1.3 Kode Program Buka File

Untuk melakukan analisis sentimen, maka sistem perlu untuk membuka *file* berisi data yang diperlukan untuk keperluan analisis sentimen. Oleh karena itu, digunakan kode program pada Kode Program 5.3 untuk membuka dan membaca *file* untuk memperoleh data. Data yang digunakan dalam penelitian meliputi, data latih, data uji, label uji, *lexicon* kata ambigu, *lexicon* frasa bernegasi, *lexicon non stopword* dan data analisis sentimen.

Algoritme 3: Buka File

```
1 def buka_file (nama_file):
2     file = open(nama_file, encoding="utf8")
3     data = file.read()
4     data = data.split('\n')
5     return data
```

Kode Program 5.3 Buka File

5.1.4 Kode Program Cleaning

Pada Kode Program 5.4 ditunjukkan kode program yang digunakan oleh sistem untuk proses *cleaning*. Pada proses *cleaning* dilakukan penggantian karakter tanda baca dan angka-angka yang tidak dibutuhkan dalam proses analisis sentimen menjadi spasi. Dalam kode program dapat dilihat bahwa karakter dan angka yang akan diganti, disimpan dalam variabel *symbols*, lalu dengan menggunakan *library numpy* dilakukan proses penggantian karakter tanda baca atau angka dengan spasi.

Algoritme 4: Cleaning

```
1 def hapus_simbol (data):
2     for i in range (len(data)):
3         data[i] = re.sub(r'\w+:\}\{2}[\d\w-]+(\.\.[\d\w-]+)*:(?:\^/[^\s/]*))*', ' ', data[i])
4         symbols = "!"#$%&()**'`-+_
5         ,.,/:;<=>?@[\]^{\}~\n1234567890"
```

Algoritme 4: Cleaning

```
1 datax = [0 for x in range(len(data))]  
2     for i in range (len(data)):  
3         datax[i] = data[i]  
4     for i in range (len(datax)):  
5         for j in symbols:  
6             datax[i] = np.char.replace(datax[i], j, ' ')  
7     return datax
```

Kode Program 5.4 Cleaning**5.1.5 Kode Program Tokenisasi**

Kode Program 5.5 berisi kode program untuk melakukan tokenisasi terhadap data twit. Pada proses ini setiap data akan dipecah menjadi kata per kata. Proses tokenisasi dilakukan oleh sistem dengan menggunakan *library* dari *nltk Python*.

Algoritme 5: Tokenisasi

```
1 =====Tokenisasi=====  
2 def tokenisasi (dataa):  
3     tokenisasi = [0 for x in range(len(dataa))]  
4     for i in range(len(dataa)):  
5         tokenisasi[i] = str(dataa[i])  
6         for i in range(len(tokenisasi)):  
7             tokenisasi[i] =  
8             nltk.tokenize.word_tokenize(tokenisasi[i])  
9         return tokenisasi  
10
```

Kode Program 5.5 Tokenisasi**5.1.6 Kode Program Case Folding**

Kode Program 5.6 berisi kode program untuk melakukan *case folding* terhadap data twit. Pada proses ini dilakukan pengubahan karakter huruf kapital menjadi huruf kecil. Hal ini dilakukan untuk mencegah kesalahan interpretasi kata karena adanya kapitalisasi karakter.

Algoritme 6: Case Folding

```
1 =====Case Folding=====  
2 def case_folding (token):  
3     for i in range (len(token)):  
4         for j in range (len(token[i])):  
5             token[i][j] = token[i][j].lower()  
6     return token
```

Kode Program 5.6 Case Folding**5.1.7 Kode Program Normalisasi kata**

Kode Program 5.7 berisi kode program untuk melakukan normalisasi kata. Pada proses ini akan dilakukan perubahan kata-kata kurang jelas atau ambigu menjadi kata yang baku. Selain itu, pada tahap ini juga dilakukan proses penggantian frasa

bernegasi menjadi kata sinonimnya yang tidak mengandung kata negasi, seperti “tidak” atau “bukan” untuk menghindari kesalahan interpretasi makna data oleh sistem karena kata negasi pada umumnya akan terhapus pada tahap *filtering*. Proses ini sangat penting dan efektif untuk mengurangi jumlah kesalahan interpretasi kata dan dapat menambah nilai parameter evaluasi sistem.

Algoritme 7: Normalisasi Kata

```
1 #####Normalisasi kata#####
2 def norm_kata (token):
3     file = open("ambigu.txt", encoding="utf8")
4     ambigu = file.read()
5     ambigu = ambigu.split('\n')
6     for i in range (len(ambigu)):
7         ambigu[i] = ambigu[i].split()
8     for i in range (len(token)):
9         for j in range (len(token[i])):
10            for k in range (len(ambigu)):
11                if(token[i][j] == ambigu[k][0]):
12                    token[i][j] = ambigu[k][1]
13                    break
14
15    #Negasi
16    file = open("negasi.txt", encoding="utf8")
17    negasi = file.read()
18    negasi = negasi.split('\n')
19    for i in range (len(negasi)):
20        negasi[i] = negasi[i].split()
21    for i in range (len(token)):
22        for j in range(len(token[i])):
23            for k in range(len(negasi)):
24                if(token[i][j] == negasi[0]):
25                    if(((j+1) < len(token[i])) and
token[i][j+1] == negasi[1]):
26                        token[i][j] = negasi[2]
27                        token[i].remove(negasi[1])
28
return token
```

Kode Program 5.7 Normalisasi kata

5.1.8 Kode Program *Term-Based Random Sampling*

Kode Program 5.8 menunjukkan kode program untuk proses *term-based random sampling*. Pada tahap ini dilakukan pembuatan *stoplist* dari *term* data latih dengan menggunakan nilai bobot Kullback-Leibler. Bobot Kullback-Leibler didasarkan pada hukum Zipf, sehingga tingkat kepentingan dari suatu kata dapat dihitung berdasarkan tingkat kemunculannya dalam dokumen. Setelah didapatkan bobot Kullback-Leibler, kemudian *term* diurutkan secara *ascending* berdasarkan nilai bobot Kullback-Leiblernya. Setelah itu akan diambil *term* dengan nilai bobot Kullback-Leibler terkecil sebanyak x persen dari total *term* yang berhasil didapatkan pada proses *random sampling*. *Term* yang diambil hanya *term* yang tidak termasuk dalam *lexicon non stopword*.

Algoritme 8: Term-Based Random Sampling

```
1 #####Term-Based Random Sampling#####
2 def tbrs(token_normalisasi, data_len, persen):
```



```
Algoritme 8: Term-Based Random Sampling
1 documents = [""] for x in range(data_len):
2     for i in range(len(token_normalisasi)):
3         for j in range (len(token_normalisasi[i])):
4             documents[i] += token_normalisasi[i][j] + " "
5             #kode untuk mengetahui term dan kemunculannya dalam
6             #dokumen
7             tokens = Counter(re.findall(r'\w+', " ".join(documents)))
8             #fungsi untuk mendapatkan nilai F
9             def F(word):
10                 return tokens.get(word)
11             #mendapatkan nilai token_c
12             TOKEN_C = 0
13             for i in tokens:
14                 TOKEN_C += tokens[i]
15             #mendapatkan nilai P_c
16             def P_c(word):
17                 return float(F(word))/TOKEN_C
18             # membuat inverted index yang berisi term dan
19             #kemunculannya dalam semua dokumen
20             def buat_inverted_indeks(data):
21                 index = defaultdict(list)
22                 for i, document in enumerate(data):
23                     for token in document.strip().split():
24                         index[token].append(i)
25                 return index
26             inv_index = buat_inverted_indeks(documents)
27             # fungsi untuk menghitung P_x
28             def P_x(word):
29                 sample = [documents[i] for i in inv_index[word]]
30                 tokens_sample = Counter(re.findall(r'\w+', ' '.join(sample)))
31                 L_x = 0
32                 for x in tokens_sample:
33                     L_x += tokens_sample[x]
34                 return float(tokens_sample[word])/L_x
35             # kullback leibler divergence
36             def kl_div(word):
37                 p_x = P_x(word)
38                 p_c = P_c(word)
39                 return p_x * log(p_x/p_c, 2)
40             # Random Sampling dan Perhitungan Kullback-Leibler
41             Divergence
42             file = open("lexicon_non_stopword.txt", encoding="utf8")
43             nonstopword = file.read()
44             nonstopword = nonstopword.split('\n')
45             terms = list(tokens.keys())
46             kl_div_val = []
47             term_pilihan = []
48             for i in range (len(terms)):
49                 angka_acak = random.randint(0,len(terms)-1)
50                 for j in range (len(nonstopword)):
51                     if(terms[angka_acak]==nonstopword[j]):
52                         break
53                     elif(j == (len(nonstopword)-1)):
54                         term_pilihan.append(terms[angka_acak])
55                         kl_div_val.append(kl_div(terms[angka_acak]))
```

Algoritme 8: Term-Based Random Sampling	
61	#Mengurutkan Terms berdasarkan nilai bobot Kullback-
62	Leiblernya
63	kl_urut = [[0 for x in range(2)] for y in
64	range(len(term_pilihan))]
65	for i in range(len(term_pilihan)):
66	kl_urut[i][0] = term_pilihan[i]; kl_urut[i][1] =
67	kl_div_val[i]
68	kl_urut = sorted(kl_urut, key=itemgetter(1))
69	term_urut = []
70	for i in range(len(kl_urut)):
71	term_urut.append(kl_urut[i][0])
72	#Mengambil term yang telah terurut dan menghapus
73	pengulangan term
74	term_terurut = []
75	for i in range(len(term_urut)):
76	[term_terurut.append(item) for item in term_urut if
77	item not in term_terurut]
78	
79	#mengambil term teratas sebanyak x dan menjadikan
80	kumpulan term tersebut sebagai stoplist
81	x = (persen/100) * len(term_terurut)
82	stoplist = []
83	for i in range(int(x)):
84	stoplist.append(term_terurut[i])
	return stoplist

Kode Program 5.8 Term-Based Random Sampling

5.1.9 Kode Program Filtering

Kode Program 5.9 berisi kode program untuk melakukan *filtering* terhadap data twit. Dalam tahap ini, kata-kata yang dianggap kurang penting akan dihapus. Proses ini dilakukan untuk memudahkan proses klasifikasi oleh sistem sehingga sistem dapat bekerja lebih efektif dan efisien.

Algoritme 9: Filtering	
1	#=====Filtering=====
2	def filtering (token_normalisasi, stoplist):
3	filtering = token_normalisasi
4	for i in range (len(filtering)):
5	len_index_j = len(filtering[i])
6	for j in range (len(filtering[i])):
7	for k in range (len(stoplist)):
8	if(j<len_index_j and filtering[i][j] ==
9	stoplist[k]):
10	filtering[i].remove(stoplist[k])
11	len_index_j -= 1
12	return filtering

Kode Program 5.9 Filtering

5.1.10 Kode Program Type

Kode Program 5.10 berisi kode program untuk melakukan *type* terhadap data twit. Dalam tahap ini, dilakukan proses penghapusan pengulangan kata yang terdapat dalam hasil proses *stemming*. Tahap ini dilakukan karena pengulangan kata menyebabkan sistem bekerja lebih berat dan tidak efisien.

Algoritme 10: Type

```
1 #=====Type=====
2 def typee (hasil_filter, data_len):
3     hasil_type = []
4     for x in range(data_len):
5         for i in range (data_len):
6             [hasil_type[i].append(item) for item in
7             hasil_filter[i] if item not in hasil_type[i]]
8     return hasil_type
```

Kode Program 5.10 Type

5.1.11 Kode Program Term

Kode Program 5.11 berisi kode program untuk memperoleh *term* dari data twit. Kode ini pada hakikatnya memiliki tujuan dan keluaran yang sama dengan kode program *type*, akan tetapi kode ini tetap diperlukan karena diperlukannya variabel yang berbeda untuk menyimpan hasil *type* dan *term* tanpa harus mengulang proses *preprocessing* dari awal.

Algoritme 11: Term

```
1 #=====Term=====
2 def term_ (hasil_type):
3     kata = ""
4     term = []
5     for i in range (len(hasil_type)):
6         for j in range (len(hasil_type[i])):
7             kata += hasil_type[i][j] + " "
8     kata = kata.split()
9     [term.append(item) for item in kata if item not in term]
10    return term
```

Kode Program 5.11 Term

5.1.12 Kode Program Perhitungan Raw Term Frequency

Kode Program 5.12 menunjukkan kode program yang digunakan untuk melakukan proses perhitungan *raw term frequency*. Untuk menghitung *Raw term frequency*, maka sebelumnya dilakukan perhitungan *term frequency* dari tiap *term* dalam tiap dokumen. Setelah didapatkan nilai *term frequency*, kemudian nilai *term frequency* dari tiap *term* pada tiap dokumen dijumlahkan, sehingga didapatkan nilai *raw term frequency* dari tiap *term*.

Algoritme 12: Perhitungan Raw term frequency

```
1 #=====Perhitungan RTF=====
2 def raw_term_frequency (term, hasil_stem, data_len):
3     tf = [[0 for x in range(len(term))] for y in
4     range(data_len)]
```

Algoritme 12: Perhitungan Raw term frequency

```
1 5 for i in range (len(term)):  
2      for j in range (len(hasil_stem)):  
3          for k in range (len(hasil_stem[j])):  
4              if (term[i] == hasil_stem[j][k]):  
5                  tf[j][i] += 1  
6  
7 10 rtf2 = [0 for x in range(len(term))]  
8      for i in range (len(term)):  
9          for j in range (data_len):  
10             rtf2[i] += tf[j][i]  
11  
12 return rtf2
```

Kode Program 5.12 Perhitungan Raw term frequency**5.1.13 Kode Program Pembuatan Stoplist**

Kode Program 5.13 menunjukkan kode program yang digunakan untuk pembuatan *stoplist* dari data latih. Semula, *file* data latih positif dan negatif dipanggil oleh sistem, kemudian dilakukan proses *preprocessing* hingga normalisasi kata. Setelah itu, *token* positif dan negatif digabung menjadi *token* keseluruhan untuk keperluan pembuatan *stoplist*. Selanjutnya, dipanggil fungsi *ttrs* dengan parameter variabel yang menyimpan nilai *token* keseluruhan, panjang baris dari data latih dan persentase *term* yang ingin diambil untuk dijadikan *stoplist*.

Algoritme 13: Pembuatan Stoplist

```
1 1 #=====Pembuatan  
2 Stoplist=====  
3 def file (nama_file):  
4     data = buka_file(nama_file)  
5     data = hapus_simbol(data)  
6     return data  
7 def token (data):  
8     token = tokenisasi(data)  
9     token = case_folding(token)  
10    token = norm_kata(token)  
11    return token  
12 #data training untuk label positif, negatif dan neutral  
13 berjumlah sama, maka nilai prob. prior tiap label pun sama  
14 prior_pos, prior_neg = 1/2, 1/2  
15 #Inisialisasi data training untuk membentuk stoplist  
16 data_pos = file('positif.txt')  
17 data_neg = file('negatif.txt')  
18 token_pos = token(data_pos)  
19 token_neg = token(data_neg)  
20 token_keseluruhan = []  
21 for i in range(len(token_neg)):  
22     token_keseluruhan.append(token_pos[i])  
23     token_keseluruhan.append(token_neg[i])  
24 stoplist = ttrs(token_keseluruhan,  
25 (len(data_pos)+len(data_neg)), 30)
```

Kode Program 5.13 Pembuatan Stoplist

5.1.14 Kode Program *Preprocessing* hingga Perhitungan *Raw term frequency* Data Latih

Kode Program 5.14 menunjukkan kode program untuk perhitungan *Raw term frequency* data latih. Sistem melakukan pemanggilan fungsi *preprocessing* terhadap tiap data latih. Setelah proses *preprocessing* pada tiap data latih dilakukan, kemudian dilakukan pemanggilan fungsi *rtf* terhadap hasil *preprocessing* pada tiap tiap data latih untuk mendapatkan nilai *Raw term frequency*-nya.

```
Algoritme 14: Perhitungan Raw term frequency Data Latih
1 #=====perhitungan RTF pada data latih di tiap
2 kelas=====
3 #Data Positif
4 hasil_filter_pos = filtering(token_pos, stoplist)
5 type_pos = typee(hasil_filter_pos, len(data_pos))
6 term_pos = term_(type_pos)
7 rtf_pos = raw_term_frequency(term_pos, hasil_filter_pos,
8 len(data_pos))
9 #Data Negatif
10 hasil_filter_neg = filtering(token_neg, stoplist)
11 type_neg = typee(hasil_filter_neg, len(data_neg))
12 term_neg = term_(type_neg)
13 rtf_neg = raw_term_frequency(term_neg, hasil_filter_neg,
14 len(data neg))
```

Kode Program 5.14 Perhitungan *Raw term frequency* Data Latih

5.1.15 Kode Program Perhitungan Jumlah Kemunculan Seluruh Term dalam Kelas

Kode Program 5.15 berisi kode program untuk menghitung jumlah kemunculan *term* dalam suatu kelas. Nilai tersebut diperlukan untuk menghitung probabilitas likelihood. Nilai yang didapatkan kemudian akan dimasukkan dalam persamaan Naïve Bayes untuk menghitung probabilitas posterior dari tiap kelas terhadap suatu data uji.

```
Algoritme 15: Hitung Jumlah TF Seluruh Term dalam Kelas
1 =====fungsi u/ menghitung jumlah kemunculan seluruh term
2 dalam kelas=====
3 def count (tf):
4     jumlah = 0
5     for i in range(len(tf)):
6         jumlah += tf[i]
7     return jumlah
```

Kode Program 5.15 Perhitungan Jumlah Kemunculan Seluruh Term dalam Kelas

5.1.16 Kode Program Perhitungan Probabilitas *Likelihood*

Kode Program 5.16 berisi kode program untuk menghitung nilai probabilitas *likelihood* dari suatu kelas. Nilai probabilitas *likelihood* didapatkan dengan

menggunakan *Multinomial Model*. Nilai tersebut kemudian digunakan untuk proses perhitungan probabilitas *posterior*.

Algoritme 16: Perhitungan Probabilitas Likelihood

```
1 #=====klasifikasi=====
2 def likelihood(type_uji, term, tf):
3     prob = [1 for x in range(len(type_uji))]
4     for i in range(len(type_uji)):
5         for j in range(len(type_uji[i])):
6             for k in range(len(term)):
7                 if (type_uji[i][j] == term[k]):
8                     prob[i] *= ((1+tf[k])/(count(tf)+len(term)))
9                     break;
10                elif (k == (len(term)-1) and type_uji[i][j]
11 != term[k]):
12                    prob[i] *= ((1)/(count(tf)+len(term)))
13
14    return prob
```

Kode Program 5.16 Perhitungan Probabilitas Likelihood

5.1.17 Kode Program Preprocessing dan Perhitungan Nilai Probabilitas Likelihood Data Uji

Setelah melakukan tahapan-tahapan sebelumnya, kemudian dilakukan pengujian oleh sistem. Untuk melakukan pengujian sistem, ditambahkan kode program dalam sistem yang ditampilkan dalam Kode Program 5.17. Langkah pertama yaitu, memanggil file yang berisi data uji. Setelah itu melakukan *preprocessing* terhadap data uji. Setelah selesai dilakukan *preprocessing*, kemudian dilakukan perhitungan nilai probabilitas *likelihood* dari data uji berdasarkan *classifier* yang didapatkan pada proses *training*.

Algoritme 17: Preprocessing dan Perhitungan Likelihood Data Uji

```
1 #=====Testing=====
2 data_uji = file('data_uji.txt')
3 token_uji = token(data_uji)
4 hasil_filter_uji = filtering(token_uji, stoplist)
5 type_uji = typee(hasil_filter_uji, len(data_uji))
6 nb_pos = likelihood(type_uji, term_pos, rtf_pos)
7 nb_neg = likelihood(type_uji, term neg, rtf neg)
```

Kode Program 5.17 Preprocessing dan Perhitungan Nilai Probabilitas Likelihood Data Uji

5.1.18 Kode Program Penentuan Kelas Data Uji

Kode Program 5.18 berisi kode program yang digunakan untuk melakukan perhitungan probabilitas *posterior* atau nilai klasifikasi Naïve Bayes. Terdapat 3 nilai probabilitas *posterior* yang didapatkan untuk tiap data, yaitu masing-masing 1 untuk tiap kelas sentimen. Oleh karena itu, perlu dilakukan penentuan kelas data uji berdasarkan nilai probabilitas *posterior* terbesar yang didapatkan.

```
Algoritme 18: Penentuan Kelas Data Uji
1 #=====penentuan kelas data uji=====
2 def naive_bayes (nb, prior):
3     for i in range (len(nb)):
4         nb[i] *= prior
5     return nb
6 nb_pos = naive_bayes(nb_pos, prior_pos)
7 nb_neg = naive_bayes(nb_neg, prior_neg)
8 best_value = []
9 for i in range (len(nb_pos)):
10    best_value.append(max(nb_pos[i], nb_neg[i]))
11 hasil_analisis = []
12 for i in range (len(best_value)):
13     if(best_value[i] == nb_pos[i]):
14         hasil_analisis.append('positif')
15     else:
16         hasil_analisis.append('negatif')
17 print(hasil_analisis)
```

Kode Program 5.18 Penentuan Kelas Data Uji

Gambar 5.1 menunjukkan contoh hasil keluaran yang dihasilkan saat Kode Program 5.18 dijalankan. Hasil keluaran berupa kelas pada tiap data uji. Urutan kelas yang ditampilkan sesuai dengan urutan data uji.

Gambar 5.1 Contoh Hasil Keluaran Kode Program 5.18

5.1.19 Kode Program Mencetak Hasil Analisis Sentimen

Kode Program 5.19 berisi kode program yang digunakan untuk melakukan mencetak hasil analisis sentimen. Hasil sentimen dari tiap data uji telah diketahui dari proses sebelumnya. Setelah itu, pada tahap ini dilakukan proses mencetak hasil analisis sentimen data uji berdasarkan kelas sentimennya.

```
Algoritme 19: Mencetak Hasil Analisis Sentimen  
1 | =====Mencetak hasil analisis sentimen=====  
2 | data_ = buka_file('data_analisis_sentimen.txt')  
3 | def print_hasil (data, sentimen, hasil_analisis):  
4 |     nomor = 0  
5 |     print ('\nHasil sentimen ',sentimen,' :\n')  
6 |     for i in range(len(hasil_analisis)):
```

```
Algoritme 19: Mencetak Hasil Analisis Sentimen
7   if(hasil_analisis[i] ==sentimen):
8       print((nomor+1),',',data_[i],', =>
9           ',hasil_analisis[i])
10      nomor+=1
11  print_hasil (data_uji, 'positif', hasil_analisis)
12  print_hasil (data_uji, 'negatif', hasil_analisis)
```

Kode Program 5.19 Mencetak Hasil Analisis Sentimen

Gambar 5.2 menunjukkan contoh hasil keluaran saat Kode Program 5.19 dijalankan. Data uji ditampilkan oleh sistem berdasarkan hasil klasifikasi tiap data uji yang telah dilakukan oleh sistem. Dari sini, dapat diketahui berapa jumlah data uji yang memiliki sentimen positif dan negatif.

```
Hasil sentimen negatif :
1 . Konglomerat di Balik Pemindahan Ibukota, Deal Sejak Pilpres 2019 ... @PojokSatu "Pemindahan Ibukota kental nilai politik & bisnis. Ada deal politik sblm Pilpres 2019," Ujang Komaruddin Univ Al Azhar => negatif
2 . "Gubernur DKI Djarot Kurang Setuju Ibukota Dipindah (Juli 3 1, 2017)" Hahahahh.. Ternyata bukan cuma si ahok, mantan Gub. Dj arat Saiful pun pernah ngga setuju dgn wacana pemindahan Ibukota. Uhuy.. uhuy.. => negatif
3 . Pak, pertanyaan selanjutnya untuk saya adalah. Dengan Pemindahan Ibukota, hutan itu apakah kelak tidak berubah menjadi perumahan? Begini pak.. jgn lupa loh, pola ekonomi & pola kebijakan semua msh centralisasi! Semua kebijakkan jd diambil dr pusat! Bahkan papua jd hrs setor pusat! => negatif
```

Gambar 5.2 Contoh Hasil Keluaran Kode Program 5.19

5.1.20 Kode Program Perhitungan Nilai Parameter Evaluasi Sistem

Sistem juga melakukan perhitungan evaluasi terhadap hasil klasifikasi sistem itu sendiri. Parameter evaluasi yang digunakan untuk menghitung evaluasi kinerja sistem yaitu, akurasi, *precision* dan *recall*. Kode program yang digunakan untuk melakukan perhitungan parameter-parameter evaluasi terdapat pada Kode Program 5.20.

```
Algoritme 20: Perhitungan Nilai Parameter Evaluasi Sistem
1 =====evaluasi sistem=====
2 label_data_uji = buka_file('label_uji.txt')
3 TN, FP, FN, TP = 0, 0, 0, 0
4 for i in range(len(label_data_uji)):
5     if (label_data_uji[i] == 'positif'):
6         if (hasil_analisis[i] == 'positif'):
7             TP += 1
8         elif (hasil_analisis[i] == 'negatif'):
9             FN += 1
10        elif (label_data_uji[i] == 'negatif'):
```

```
Algoritme 20: Perhitungan Nilai Parameter Evaluasi Sistem
11     if(hasil_analisis[i] == 'positif'):
12         FP += 1
13     elif (hasil_analisis[i] == 'negatif'):
14         TN += 1
15     akurasi = (TP + TN) / (TP + TN + FP + FN)
16     precision = TP / (TP + FP)
17     recall = TP / (TP + FN)
18     print('True Positive = ',TP,'\\nFalse Positive = ',FP,'\\nTrue Negative = ',TN,'\\nFalse Negative = ',FN)
19     print('\\nHasil evaluasi sistem :\\n\\n1. Akurasi :')
20     print(akurasi,'\\n2. Precision :',precision,'\\n3. Recall :')
21     print(recall)
```

Kode Program 5.20 Perhitungan Nilai Parameter Evaluasi Sistem

Gambar 5.3 menunjukkan contoh hasil keluaran saat Kode Program 5.20 dijalankan. Parameter evaluasi yang dihitung antara lain, akurasi, *precision* dan *recall*. Perhitungan dilakukan berdasarkan *confusion matrix* yang dihasilkan dari pencocokan hasil klasifikasi sistem dengan nilai *ground truth* dari data uji.

```
True Positive = 25
False Positive = 5
True Negative = 20
False Negative = 0

Hasil evaluasi sistem :

1. Akurasi : 0.9
2. Presisi : 0.8333333333333334
3. Recall : 1.0
```

Gambar 5.3 Contoh Hasil Keluaran Kode Program 5.20

BAB 6 HASIL DAN PEMBAHASAN

6.1 Hasil Penelitian

6.1.1 Evaluasi

Evaluasi dilakukan untuk mengukur atau menilai seberapa baik kinerja sistem.

Confusion matrix akan digunakan untuk membantu proses evaluasi. Dalam perhitungannya, akan digunakan 50 data yang telah diberi label untuk kemudian dibandingkan dengan label hasil klasifikasi sistem. Hasil label dari *ground truth* dan hasil sistem kemudian akan digunakan untuk membuat *confusion matrix*. Setelah *confusion matrix* terbuat, kemudian akan dihitung evaluasi dari sistem. Evaluasi yang akan digunakan yaitu akurasi, *precision* dan *recall*.

6.1.1.1 Evaluasi Hasil Klasifikasi Sistem dengan Stoplist 0 Persen dan 10 Persen

Evaluasi pertama dilakukan berdasarkan hasil klasifikasi sistem dengan menggunakan *stoplist* 0 persen. Kemudian evaluasi kedua dilakukan dengan menggunakan *stoplist* 10 persen dan ternyata memiliki hasil klasifikasi yang sama dengan *stoplist* 0 persen. Berikut adalah cuplikan 10 dari 50 data yang digunakan beserta labelnya dan hasil klasifikasi sistem dengan *stoplist* 0 persen dan 10 persen yang ditampilkan pada Tabel 6.1.

Tabel 6.1 Cuplikan Hasil Klasifikasi Data Uji dengan *Stoplist* 0 Persen

No	Data	GT	P
1	Seandainya Pemerintah lebih kedepankan kepentingan Rakyat banyak, maka kalaupun pemerintah punya anggaran yg agak longgar, mestinya dipergunakan unt selesaikan masalah yg mendesak spt soal BPJS ini. Itu lebih diperlukan masyarakat menengah kebawah, ketimbang pemindahan Ibukota.	-	-
2	Payung hukum pemindahan Ibukota belum ada.	-	-
3	eksekutif ga bisa memutuskan pindah Ibukota secara sepikah. kok presiden kaya ga ngerti prosedur gini ya? sapa sih yg milih? lol	-	-
4	Kali ini sy tidak setuju. dana yang dikeluarkan sangat banyak dan itu rencananya bakal NGUTANG!	-	+
5	pindah negara tdk meningkatkan nilai ekonomi bagi rakyat maupun negara..	-	-
6	Saya rakyat dan mendukung Ibukota pindah.. biar Kalimantan juga cepat dibangun infrastrukturnya..	+	+
7	Saya setuju Ibukota pindah ke daerah kalimantan.	+	+
8	Saya mendukung Ibukota pindah karena dampak URBANISASI disekitar Jakarta,, sawah sawah produktif banyak dijadikan PERUMAHAN	+	+



No	Data	GT	P
9	Saya org Jawa juga setuju Ibukota pindah ke Kalimantan...biar pembangunan Indonesia sentris	+	+
....
50	Dukung 100 persen Pemindahan Ibukota Insya Allah...selalu husnudzon ada kebaikan di dalamnya, walaupun banyak beredar kabar ini kedengkian Pak Jokowi pada Pak Anies. Wallahu'alam bishawab	+	+

Keterangan:

GT = *Ground Truth*

P = Prediksi

Sistem juga melakukan perhitungan evaluasi terhadap hasil klasifikasi sistem itu sendiri. Evaluasi dilakukan dengan menggunakan *confusion matrix* untuk mengukur kinerja sistem dengan menggunakan parameter evaluasi yang meliputi akurasi, *precision* dan *recall*. *Confussion matrix* dari hasil pengujian sistem dengan menggunakan *stoplist* 0 persen dan 10 persen ditampilkan dalam Tabel 6.2.

Tabel 6.2 *Confusion Matrix* Hasil Klasifikasi Sistem

<i>Confusion Matrix</i>		Prediksi	
		Positif	Negatif
Aktual	Positif	25	0
	Negatif	5	20

Berdasarkan Persamaan 2.5, Persamaan 2.6 dan Persamaan 2.7 dapat dihitung nilai FP, FN dan TN pada tiap kelas:

$$FP(\text{positif}) = \sum_i^l E_{li} - TP_i = (25 + 5) - 25 = 5$$

$$FN(\text{positif}) = \sum_i^l E_{il} - TP_i = (25 + 0) - 25 = 0$$

$$TN(\text{positif}) = \sum_i^l \sum_k^l E_{lk} - TP_i - FP_i - FN_i \\ = (25 + 0 + 5 + 20) - 25 - 5 - 0 = 20$$



Perhitungan evaluasinya sebagai berikut:

$$\text{akurasi(positif)} = \frac{25 + 20}{25 + 20 + 5 + 0} = 0,9$$

$$\text{precision(positif)} = \frac{25}{25 + 5} = 0,8333$$

$$\text{recall(positif)} = \frac{25}{25 + 0} = 1$$

$$F\text{-measure(positif)} = 2 \times \frac{(0,8333 \times 1)}{(0,8333 + 1)} = 0,909$$

Perhitungan FP, FN dan TN untuk kelas negatif:

$$FP(\text{negatif}) = \sum_i^l E_{li} - TP_i = (0 + 20) - 20 = 0$$

$$FN(\text{negatif}) = \sum_i^l E_{il} - TP_i = (5 + 20) - 20 = 5$$

$$\begin{aligned} TN(\text{negatif}) &= \sum_i^l \sum_k^l E_{ik} - TP_i - FP_i - FN_i \\ &= (25 + 0 + 5 + 20) - 20 - 0 - 5 = 25 \end{aligned}$$

Perhitungan evaluasinya sebagai berikut:

$$\text{akurasi(negatif)} = \frac{25 + 20}{25 + 20 + 5 + 0} = 0,9$$

$$\text{precision(negatif)} = \frac{20}{20 + 0} = 1$$

$$\text{recall(negatif)} = \frac{20}{20 + 5} = 0,8$$

$$F\text{-measure(negatif)} = 2 \times \frac{(1 \times 0,8)}{(1 + 0,8)} = 0,88$$



Perhitungan *macroaverage* parameter evaluasi seluruh kelas:

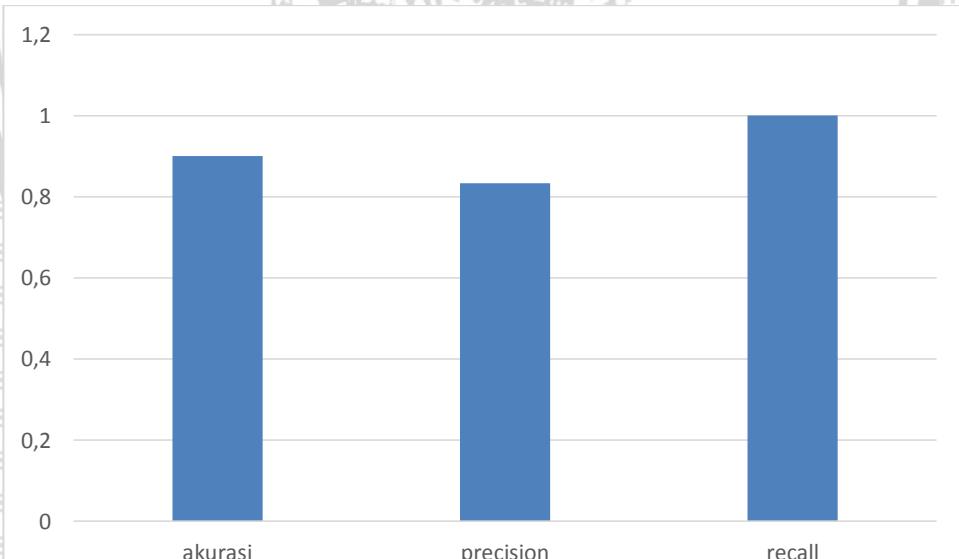
$$\text{akurasi} = \frac{0,9 + 0,9}{2} = 0,9$$

$$\text{precision} = \frac{0,8333 + 1}{2} = 0,917$$

$$\text{recall} = \frac{1 + 0,8}{2} = 0,9$$

$$F - \text{measure} = \frac{0,909 + 0,88}{2} = 0,89$$

Gambar 6.1 menampilkan hasil perhitungan evaluasi terhadap kinerja sistem dalam bentuk grafik. Perhitungan evaluasi didasarkan pada hasil klasifikasi data uji oleh sistem dilakukan menggunakan *stoplist* 0 persen dan 10 persen dengan nilai *ground truth* dari data uji.



Gambar 6.1 Grafik Evaluasi Sistem dengan *Stoplist* 0 Persen dan 10 Persen

6.1.1.2 Evaluasi Hasil Klasifikasi Sistem dengan *Stoplist* 20 Persen

Setelah selesai melakukan pengujian berdasarkan hasil klasifikasi sistem dengan menggunakan *stoplist* 0 persen dan 10 persen, kemudian dilanjutkan



dengan pengujian berdasarkan hasil klasifikasi sistem dengan stoplist 20 persen. Pada Tabel 6.3 ditunjukkan cuplikan hasil klasifikasi data uji oleh sistem dengan menggunakan stoplist 20 persen.

Tabel 6.3 Cuplikan Hasil Klasifikasi Data Uji dengan Stoplist 20 Persen

No	Data	GT	P
1	Seandainya Pemerintah lebih kedepankan kepentingan Rakyat banyak, maka kalaupun pemerintah punya anggaran yg agak longgar, mestinya dipergunakan unt selesaikan masalah yg mendesak spt soal BPJS ini. Itu lebih diperlukan masyarakat menengah kebawah, ketimbang pemindahan Ibukota.	-	-
2	Payung hukum pemindahan Ibukota belum ada.	-	-
3	eksekutif ga bisa memutuskan pindah Ibukota secara sepikak. kok presiden kaya ga ngerti prosedur gini ya? sapa sih yg milih? lol	-	-
4	Kali ini sy tidak setuju. dana yang dikeluarkan sangat banyak dan itu rencananya bakal NGUTANG!	-	+
5	pindah negara tdk meningkatkan nilai ekonomi bagi rakyat maupun negara..	-	-
6	Saya rakyat dan mendukung Ibukota pindah.. biar Kalimantan juga cepat dibangun infrastrukturnya..	+	+
7	Saya setuju Ibukota pindah ke daerah kalimantan.	+	+
8	Saya mendukung Ibukota pindah karena dampak URBANISASI disekitar Jakarta,, sawah sawah produktif banyak dijadikan PERUMAHAN	+	+
9	Saya org Jawa juga setuju Ibukota pindah ke Kalimantan...biar pembangunan Indonesia sentris	+	+
....
50	Dukung 100 persen Pemindahan Ibukota Insya Allah...selalu husnudzon ada kebaikan di dalamnya, walaupun banyak beredar kabar ini kedengkian Pak Jokowi pada Pak Anies. Wallahu'alam bishawab	+	+

Sistem juga melakukan perhitungan evaluasi terhadap hasil klasifikasi sistem itu sendiri. Evaluasi dilakukan dengan menggunakan *confusion matrix* untuk mengukur kinerja sistem dengan menggunakan parameter evaluasi yang meliputi akurasi, *precision* dan *recall*. *Confussion matrix* dari hasil pengujian sistem dengan menggunakan stoplist 20 Persen ditampilkan dalam Tabel 6.4.

Tabel 6.4 Confusion Matrix Hasil Klasifikasi Sistem

Confusion Matrix		Prediksi	
		Positif	Negatif
Aktual	Positif	25	0
	Negatif	3	22

Berdasarkan Persamaan 2.5, Persamaan 2.6 dan Persamaan 2.7 dapat dihitung nilai FP, FN dan TN pada tiap kelas:

$$FP(\text{positif}) = \sum_i^t E_{li} - TP_i = (25 + 3) - 25 = 3$$

$$FN(\text{positif}) = \sum_i^l E_{il} - TP_i = (25 + 0) - 25 = 0$$

$$TN(\text{positif}) = \sum_i^l \sum_k^l E_{lk} - TP_i - FP_i - FN_i \\ = (25 + 0 + 3 + 22) - 25 - 3 - 0 = 22$$

Perhitungan evaluasinya sebagai berikut:

$$akurasi(positif) = \frac{25 + 22}{25 + 22 + 3 + 0} = 0,94$$

$$precision(positif) = \frac{25}{25+3} = 0,89$$

$$recall(\text{positif}) = \frac{25}{25+0} = 1$$

$$F - measure(positif) = 2 \times \frac{(0,89 \times 1)}{(0,89 + 1)} = 0,94$$

Perhitungan FP, FN dan TN untuk kelas negatif:

$$FP(negatif) = \sum E_{li} - TP_i = (0 + 22) - 22 = 0$$



$$FN(negatif) = \sum_i^l E_{il} - TP_i = (3 + 22) - 22 = 3$$

$$TN(negatif) = \sum_i^l \sum_k^l E_{lk} - TP_i - FP_i - FN_i$$

$$= (25 + 0 + 3 + 22) - 22 - 0 - 3 = 25$$

Perhitungan evaluasinya sebagai berikut:

$$akurasi(negatif) = \frac{25 + 22}{25 + 22 + 3 + 0} = 0,94$$

$$precision(negatif) = \frac{22}{22 + 0} = 1$$

$$recall(negatif) = \frac{22}{22 + 3} = 0,88$$

$$F-measure(negatif) = 2 \times \frac{(1 \times 0,88)}{(1 + 0,88)} = 0,936$$

Perhitungan *macroaverage* parameter evaluasi seluruh kelas:

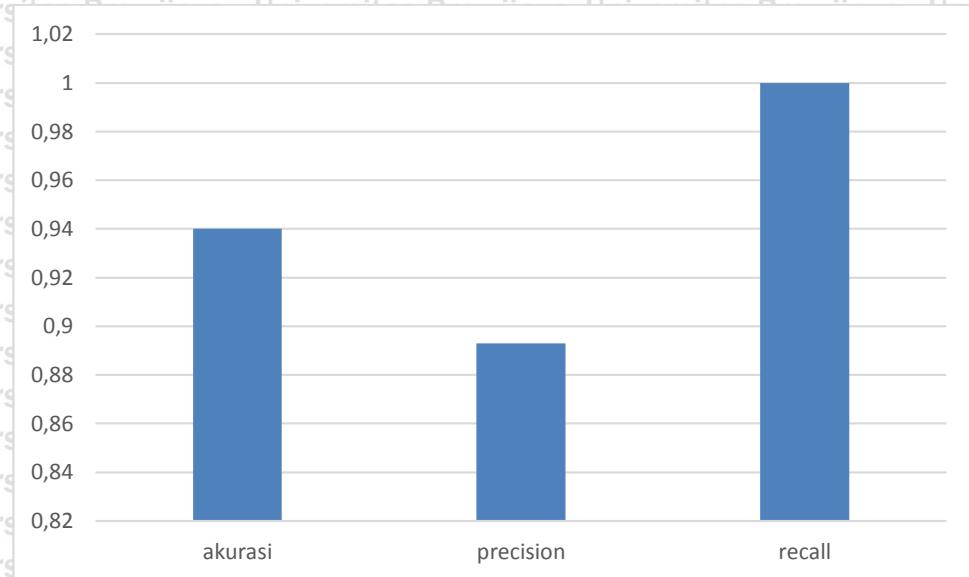
$$akurasi = \frac{0,94 + 0,94}{2} = 0,94$$

$$precision = \frac{0,89 + 1}{2} = 0,945$$

$$recall = \frac{1 + 0,88}{2} = 0,94$$

$$F-measure = \frac{0,94 + 0,936}{2} = 0,938$$

Gambar 6.2 menampilkan hasil perhitungan evaluasi terhadap kinerja sistem dalam bentuk grafik. Perhitungan evaluasi didasarkan pada hasil klasifikasi data uji oleh sistem dilakukan menggunakan *stoplist* 20 persen dengan nilai *ground truth* dari data uji.



Gambar 6.2 Grafik Evaluasi Sistem dengan *Stoplist* 20 Persen

6.1.1.3 Evaluasi Hasil Klasifikasi Sistem dengan *Stoplist* 30 Persen

Setelah selesai melakukan pengujian berdasarkan hasil klasifikasi sistem dengan menggunakan *stoplist* 20 persen, kemudian dilanjutkan dengan pengujian berdasarkan hasil klasifikasi sistem dengan *stoplist* 30 persen. Pada Tabel 6.1 ditunjukkan cuplikan hasil klasifikasi data uji oleh sistem dengan menggunakan *stoplist* 30 persen.

Tabel 6.5 Cuplikan Hasil Klasifikasi Data Uji dengan *Stoplist* 30 Persen

No	Data	GT	P
1	Seandainya Pemerintah lebih kedepankan kepentingan Rakyat banyak, maka kalaupun pemerintah punya anggaran yg agak longgar, mestinya dipergunakan unt selesaikan masalah yg mendesak spt soal BPJS ini. Itu lebih diperlukan masyarakat menengah kebawah, ketimbang pemindahan Ibukota.	-	-
2	Payung hukum pemindahan Ibukota belum ada.	-	+
3	eksekutif ga bisa memutuskan pindah Ibukota secara sepihak. kok presiden kaya ga ngerti prosedur gini ya? sapa sih yg milih? lol	-	-
4	Kali ini sy tidak setuju. dana yang dikeluarkan sangat banyak dan itu rencananya bakal NGUTANG!	-	+
5	pindah negara tdk meningkatkan nilai ekonomi bagi rakyat maupun negara..	-	-
6	Saya rakyat dan mendukung Ibukota pindah.. biar Kalimantan juga cepat dibangun infrastrukturnya..	+	+

No	Data	GT	P
7	Saya setuju Ibukota pindah ke daerah kalimantan.	+	+
8	Saya mendukung Ibukota pindah karena dampak URBANISASI disekitar Jakarta,, sawah sawah produktif banyak dijadikan PERUMAHAN	+	+
9	Saya org Jawa juga setuju Ibukota pindah ke Kalimantan...biar pembangunan Indonesia sentris	+	+
....
50	Dukung 100 persen Pemindahan Ibukota Insya Allah...selalu husnudzon ada kebaikan di dalamnya, walaupun banyak beredar kabar ini kedengkian Pak Jokowi pada Pak Anies. Wallahu'alam bishawab	+	+

Sistem juga melakukan perhitungan evaluasi terhadap hasil klasifikasi sistem itu sendiri. Evaluasi dilakukan dengan menggunakan *confusion matrix* untuk mengukur kinerja sistem dengan menggunakan parameter evaluasi yang meliputi akurasi, *precision* dan *recall*. *Confussion matrix* dari hasil pengujian sistem dengan menggunakan *stoplist* 30 Persen ditampilkan dalam Tabel 6.6.

Tabel 6.6 Confusion Matrix Hasil Klasifikasi Sistem

Confusion Matrix		Prediksi	
		Positif	Negatif
Aktual	Positif	25	0
	Negatif	4	21

Berdasarkan Persamaan 2.5, Persamaan 2.6 dan Persamaan 2.7 dapat dihitung nilai FP, FN dan TN pada tiap kelas:

$$FP(\text{positif}) = \sum_i^l E_{li} - TP_i = (25 + 4) - 25 = 4$$

$$FN(\text{positif}) = \sum_i^l E_{il} - TP_i = (25 + 0) - 25 = 0$$

$$TN(\text{positif}) = \sum_i^l \sum_k^l E_{lk} - TP_i - FP_i - FN_i \\ = (25 + 0 + 4 + 21) - 25 - 4 - 0 = 21$$

Perhitungan evaluasinya sebagai berikut:



$$\text{akurasi(positif)} = \frac{25 + 21}{25 + 21 + 4 + 0} = 0,92$$

$$\text{precision(positif)} = \frac{25}{25 + 4} = 0,86$$

$$\text{recall(positif)} = \frac{25}{25 + 0} = 1$$

$$F\text{-measure(positif)} = 2 \times \frac{(0,86 \times 1)}{(0,86 + 1)} = 0,925$$

Perhitungan FP, FN dan TN untuk kelas negatif:

$$FP(\text{negatif}) = \sum_i^l E_{li} - TP_i = (0 + 21) - 21 = 0$$

$$FN(\text{negatif}) = \sum_i^l E_{il} - TP_i = (4 + 21) - 21 = 4$$

$$\begin{aligned} TN(\text{negatif}) &= \sum_i^l \sum_k^l E_{lk} - TP_i - FP_i - FN_i \\ &= (25 + 0 + 4 + 21) - 21 - 0 - 4 = 25 \end{aligned}$$

Perhitungan evaluasinya sebagai berikut:

$$\text{akurasi(negatif)} = \frac{25 + 21}{25 + 21 + 4 + 0} = 0,92$$

$$\text{precision(negatif)} = \frac{21}{21 + 0} = 1$$

$$\text{recall(negatif)} = \frac{21}{21 + 4} = 0,84$$

$$F\text{-measure(negatif)} = 2 \times \frac{(1 \times 0,84)}{(1 + 0,84)} = 0,913$$



Perhitungan *macroaverage* parameter evaluasi seluruh kelas:

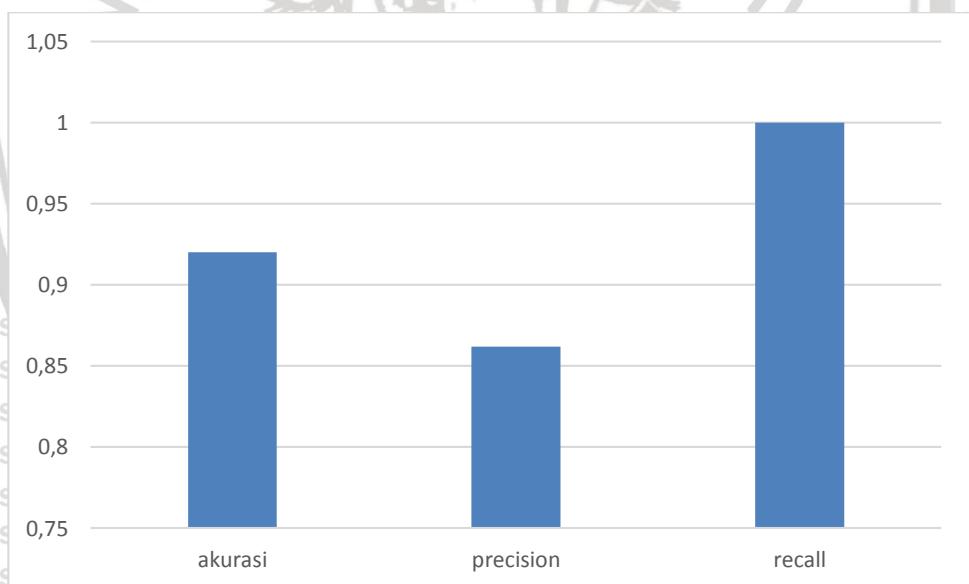
$$\text{akurasi} = \frac{0,92 + 0,92}{2} = 0,92$$

$$\text{precision} = \frac{0,86 + 1}{2} = 0,93$$

$$\text{recall} = \frac{1 + 0,84}{2} = 0,92$$

$$F - \text{measure} = \frac{0,94 + 0,936}{2} = 0,919$$

Gambar 6.3 menampilkan hasil perhitungan evaluasi terhadap kinerja sistem dalam bentuk grafik. Perhitungan evaluasi didasarkan pada hasil klasifikasi data uji oleh sistem dilakukan menggunakan *stoplist* 30 persen dengan nilai *ground truth* dari data uji.



Gambar 6.3 Grafik Evaluasi Sistem dengan *Stoplist* 30 Persen

6.1.1.4 Evaluasi Hasil Klasifikasi Sistem dengan *Stoplist* 40 Persen

Evaluasi pertama dilakukan berdasarkan hasil klasifikasi sistem dengan menggunakan *stoplist* 30 persen. Kemudian evaluasi kedua dilakukan dengan menggunakan *stoplist* 40 persen. Berikut adalah cuplikan 10 dari 50 data yang digunakan beserta labelnya dan hasil klasifikasi sistem dengan *stoplist* 40 persen yang ditampilkan pada Tabel 6.7.

Tabel 6.7 Cuplikan Hasil Klasifikasi Data Uji dengan Stoplist 40 Persen

No	Data	GT	P
1	Seandainya Pemerintah lebih kedepankan kepentingan Rakyat banyak, maka kalaupun pemerintah punya anggaran yg agak longgar, mestinya dipergunakan unt selesaikan masalah yg mendesak spt soal BPJS ini. Itu lebih diperlukan masyarakat menengah kebawah, ketimbang pemindahan Ibukota.	-	-
2	Payung hukum pemindahan Ibukota belum ada.	-	-
3	eksekutif ga bisa memutuskan pindah Ibukota secara sepikak. kok presiden kaya ga ngerti prosedur gini ya? sapa sih yg milih? lol	-	-
4	Kali ini sy tidak setuju. dana yang dikeluarkan sangat banyak dan itu rencananya bakal NGUTANG!	-	+
5	pindah negara tdk meningkatkan nilai ekonomi bagi rakyat maupun negara..	-	-
6	Saya rakyat dan mendukung Ibukota pindah.. biar Kalimantan juga cepat dibangun infrastrukturnya..	+	+
7	Saya setuju Ibukota pindah ke daerah kalimantan.	+	+
8	Saya mendukung Ibukota pindah karena dampak URBANISASI disekitar Jakarta,, sawah sawah produktif banyak dijadikan PERUMAHAN	+	+
9	Saya org Jawa juga setuju Ibukota pindah ke Kalimantan...biar pembangunan Indonesia sentris	+	+
....
50	Dukung 100 persen Pemindahan Ibukota Insya Allah...selalu husnudzon ada kebaikan di dalamnya, walaupun banyak beredar kabar ini kedengkian Pak Jokowi pada Pak Anies. Wallahu'alam bishawab	+	+

Tabel 6.8 Confusion Matrix Hasil Klasifikasi Sistem

Confusion Matrix		Prediksi	
		Positif	Negatif
Aktual	Positif	24	1
	Negatif	5	20



Berdasarkan Persamaan 2.5, Persamaan 2.6 dan Persamaan 2.7 dapat dihitung nilai FP, FN dan TN pada tiap kelas:

$$FP(\text{positif}) = \sum_i^l E_{li} - TP_i = (24 + 5) - 24 = 5$$

$$FN(\text{positif}) = \sum_i^l E_{il} - TP_i = (24 + 1) - 24 = 1$$

$$TN(\text{positif}) = \sum_i^l \sum_k^l E_{lk} - TP_i - FP_i - FN_i$$

$$= (24 + 1 + 5 + 20) - 24 - 5 - 1 = 20$$

Perhitungan evaluasinya sebagai berikut:

$$\text{akurasi(positif)} = \frac{24 + 20}{24 + 20 + 5 + 1} = 0,88$$

$$\text{precision(positif)} = \frac{24}{24 + 5} = 0,828$$

$$\text{recall(positif)} = \frac{24}{24 + 1} = 0,96$$

$$F\text{-measure(positif)} = 2 \times \frac{(0,828 \times 0,96)}{(0,828 + 0,96)} = 0,89$$

Perhitungan FP, FN dan TN untuk kelas negatif:

$$FP(\text{negatif}) = \sum_i^l E_{li} - TP_i = (1 + 20) - 20 = 1$$

$$FN(\text{negatif}) = \sum_i^l E_{il} - TP_i = (5 + 20) - 20 = 5$$

$$TN(\text{negatif}) = \sum_i^l \sum_k^l E_{lk} - TP_i - FP_i - FN_i$$

$$= (24 + 1 + 5 + 20) - 20 - 1 - 5 = 24$$



Perhitungan evaluasinya sebagai berikut:

$$\text{akurasi(negatif)} = \frac{24 + 20}{24 + 20 + 5 + 1} = 0,88$$

$$\text{precision(negatif)} = \frac{20}{20 + 1} = 0,95$$

$$\text{recall(negatif)} = \frac{20}{20 + 5} = 0,8$$

$$F - \text{measure(negatif)} = 2 \times \frac{(0,95 \times 0,8)}{(0,95 + 0,8)} = 0,87$$

Perhitungan *macroaverage* parameter evaluasi seluruh kelas:

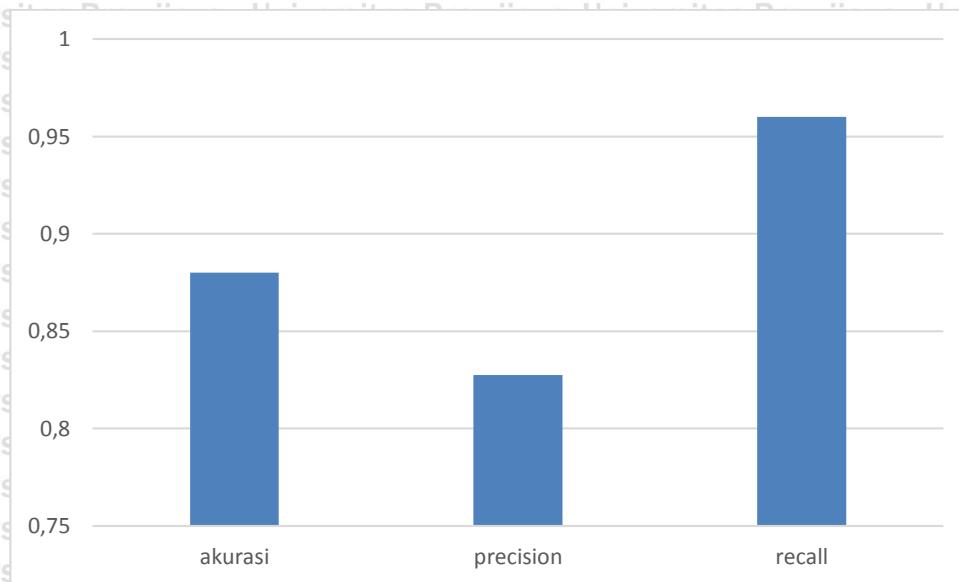
$$\text{akurasi} = \frac{0,88 + 0,88}{2} = 0,88$$

$$\text{precision} = \frac{0,828 + 0,95}{2} = 0,889$$

$$\text{recall} = \frac{0,96 + 0,8}{2} = 0,88$$

$$F - \text{measure} = \frac{0,89 + 0,87}{2} = 0,88$$

Gambar 6.4 menampilkan hasil perhitungan evaluasi terhadap kinerja sistem dalam bentuk grafik. Perhitungan evaluasi didasarkan pada hasil klasifikasi data uji oleh sistem dilakukan menggunakan *stoplist* 40 persen dengan nilai *ground truth* dari data uji.

**Gambar 6.4 Grafik Evaluasi Sistem dengan *Stoplist* 40 Persen****6.1.1.5 Evaluasi Hasil Klasifikasi Sistem dengan *Stoplist* 50 Persen**

Evaluasi pertama dilakukan berdasarkan hasil klasifikasi sistem dengan menggunakan *stoplist* 40 persen. Kemudian evaluasi kedua dilakukan dengan menggunakan *stoplist* 50 persen. Berikut adalah cuplikan 10 dari 50 data yang digunakan beserta labelnya dan hasil klasifikasi sistem dengan *stoplist* 50 persen yang ditampilkan pada Tabel 6.9.

Tabel 6.9 Cuplikan Hasil Klasifikasi Data Uji dengan *Stoplist* 50 Persen

No	Data	GT	P
1	Seandainya Pemerintah lebih kedepankan kepentingan Rakyat banyak, maka kalaupun pemerintah punya anggaran yg agak longgar, mestinya dipergunakan unt selesaikan masalah yg mendesak spt soal BPJS ini. Itu lebih diperlukan masyarakat menengah kebawah, ketimbang pemindahan Ibukota.	-	-
2	Payung hukum pemindahan Ibukota belum ada.	-	-
3	eksekutif ga bisa memutuskan pindah Ibukota secara sepahak. kok presiden kaya ga ngerti prosedur gini ya? sapa sih yg milih? lol	-	-
4	Kali ini sy tidak setuju. dana yang dikeluarkan sangat banyak dan itu rencananya bakal NGUTANG!	-	+
5	pindah negara tdk meningkatkan nilai ekonomi bagi rakyat maupun negara..	-	-
6	Saya rakyat dan mendukung Ibukota pindah.. biar Kalimantan juga cepat dibangun infrastrukturnya..	+	+

No	Data	GT	P
7	Saya setuju Ibukota pindah ke daerah kalimantan.	+	+
8	Saya mendukung Ibukota pindah karena dampak URBANISASI disekitar Jakarta,, sawah sawah produktif banyak dijadikan PERUMAHAN	+	+
9	Saya org Jawa juga setuju Ibukota pindah ke Kalimantan...biar pembangunan Indonesia sentris	+	+
....
50	Dukung 100 persen Pemindahan Ibukota Insya Allah...selalu husnudzon ada kebaikan di dalamnya, walaupun banyak beredar kabar ini kedengkian Pak Jokowi pada Pak Anies. Wallahu'alam bishawab	+	+

Sistem juga melakukan perhitungan evaluasi terhadap hasil klasifikasi sistem itu sendiri. Evaluasi dilakukan dengan menggunakan *confusion matrix* untuk mengukur kinerja sistem dengan menggunakan parameter evaluasi yang meliputi akurasi, *precision* dan *recall*. *Confussion matrix* dari hasil pengujian sistem dengan menggunakan *stoplist* 50 persen ditampilkan dalam Tabel 6.10.

Tabel 6.10 Confusion Matrix Hasil Klasifikasi Sistem

Confusion Matrix		Prediksi	
		Positif	Negatif
Aktual	Positif	24	1
	Negatif	6	19

Berdasarkan Persamaan 2.5, Persamaan 2.6 dan Persamaan 2.7 dapat dihitung nilai FP, FN dan TN pada tiap kelas:

$$FP(\text{positif}) = \sum_i^l E_{li} - TP_i = (24 + 6) - 24 = 6$$

$$FN(\text{positif}) = \sum_i^l E_{il} - TP_i = (24 + 1) - 24 = 1$$

$$TN(\text{positif}) = \sum_i^l \sum_k^l E_{lk} - TP_i - FP_i - FN_i \\ = (24 + 1 + 6 + 19) - 24 - 6 - 1 = 19$$



Perhitungan evaluasinya sebagai berikut:

$$\text{akurasi(positif)} = \frac{24 + 19}{24 + 19 + 6 + 1} = 0,86$$

$$\text{precision(positif)} = \frac{24}{24 + 6} = 0,8$$

$$\text{recall(positif)} = \frac{24}{24 + 1} = 0,96$$

$$F\text{-measure(positif)} = 2 \times \frac{(0,8 \times 0,96)}{(0,8 + 0,96)} = 0,872$$

Perhitungan FP, FN dan TN untuk kelas negatif:

$$FP(\text{negatif}) = \sum_i^l E_{li} - TP_i = (1 + 19) - 19 = 1$$

$$FN(\text{negatif}) = \sum_i^l E_{il} - TP_i = (6 + 19) - 19 = 6$$

$$\begin{aligned} TN(\text{negatif}) &= \sum_i^l \sum_k^l E_{lk} - TP_i - FP_i - FN_i \\ &= (24 + 1 + 6 + 19) - 19 - 1 - 6 = 24 \end{aligned}$$

Perhitungan evaluasinya sebagai berikut:

$$\text{akurasi(negatif)} = \frac{24 + 19}{24 + 19 + 6 + 1} = 0,86$$

$$\text{precision(negatif)} = \frac{19}{19 + 1} = 0,95$$

$$\text{recall(negatif)} = \frac{19}{19 + 6} = 0,76$$

$$F\text{-measure(negatif)} = 2 \times \frac{(0,95 \times 0,76)}{(0,95 + 0,76)} = 0,84$$



Perhitungan *macroaverage* parameter evaluasi seluruh kelas:

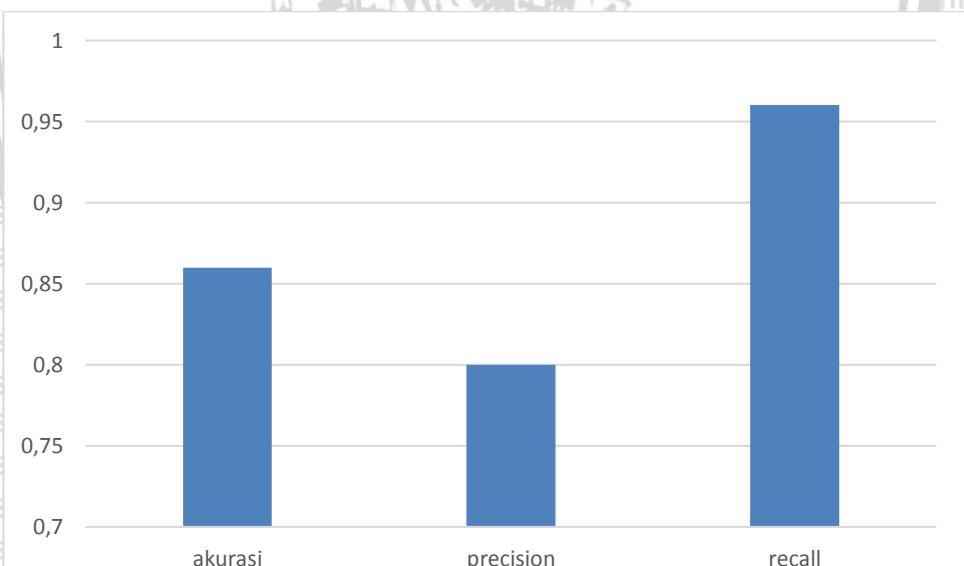
$$\text{akurasi} = \frac{0,86 + 0,86}{2} = 0,86$$

$$\text{precision} = \frac{0,8 + 0,95}{2} = 0,875$$

$$\text{recall} = \frac{0,96 + 0,76}{2} = 0,86$$

$$F - \text{measure} = \frac{0,872 + 0,84}{2} = 0,856$$

Gambar 6.5 menampilkan hasil perhitungan evaluasi terhadap kinerja sistem dalam bentuk grafik. Perhitungan evaluasi didasarkan pada hasil klasifikasi data uji oleh sistem dilakukan menggunakan *stoplist* 50 persen dengan nilai *ground truth* dari data uji.



Gambar 6.5 Grafik Evaluasi Sistem dengan Stoplist 50 Persen

6.1.1.6 Evaluasi Hasil Klasifikasi Sistem dengan Stoplist 60 Persen

Evaluasi pertama dilakukan berdasarkan hasil klasifikasi sistem menggunakan *stoplist* 50 persen. Kemudian evaluasi kedua dilakukan



menggunakan *stoplist* 60 persen. Berikut adalah cuplikan 10 dari 50 data yang digunakan beserta labelnya dan hasil klasifikasi sistem dengan *stoplist* 60 persen yang ditampilkan pada Tabel 6.11.

Tabel 6.11 Cuplikan Hasil Klasifikasi Data Uji dengan Stoplist 60 Persen

No	Data	GT	P
1	Seandainya Pemerintah lebih kedepankan kepentingan Rakyat banyak, maka kalaupun pemerintah punya anggaran yg agak longgar, mestinya dipergunakan unt selesaikan masalah yg mendesak spt soal BPJS ini. Itu lebih diperlukan masyarakat menengah kebawah, ketimbang pemindahan Ibukota.	-	-
2	Payung hukum pemindahan Ibukota belum ada.	-	+
3	eksekutif ga bisa memutuskan pindah Ibukota secara sepikak. kok presiden kaya ga ngerti prosedur gini ya? sapa sih yg milih? lol	-	+
4	Kali ini sy tidak setuju. dana yang dikeluarkan sangat banyak dan itu rencananya bakal NGUTANG!	-	+
5	pindah negara tdk meningkatkan nilai ekonomi bagi rakyat maupun negara..	-	-
6	Saya rakyat dan mendukung Ibukota pindah.. biar Kalimantan juga cepat dibangun infrastrukturnya..	+	+
7	Saya setuju Ibukota pindah ke daerah kalimantan.	+	+
8	Saya mendukung Ibukota pindah karena dampak URBANISASI disekitar Jakarta,, sawah sawah produktif banyak dijadikan PERUMAHAN	+	+
9	Saya org Jawa juga setuju Ibukota pindah ke Kalimantan...biar pembangunan Indonesia sentris	+	+
....
50	Dukung 100 persen Pemindahan Ibukota Insya Allah...selalu husnudzon ada kebaikan di dalamnya, walaupun banyak beredar kabar ini kedengkian Pak Jokowi pada Pak Anies. Wallahu'alam bishawab	+	+

Sistem juga melakukan perhitungan evaluasi terhadap hasil klasifikasi sistem itu sendiri. Evaluasi dilakukan dengan menggunakan *confusion matrix* untuk mengukur kinerja sistem dengan menggunakan parameter evaluasi yang meliputi akurasi, *precision* dan *recall*. *Confussion matrix* dari hasil pengujian sistem dengan menggunakan *stoplist* 60 persen ditampilkan dalam Tabel 6.12.

**Tabel 6.12 Confusion Matrix Hasil Klasifikasi Sistem**

		Confusion Matrix		Prediksi
		Positif	Negatif	
Aktual	Positif	24	1	
	Negatif	6	19	

Berdasarkan Persamaan 2.5, Persamaan 2.6 dan Persamaan 2.7 dapat dihitung nilai FP, FN dan TN pada tiap kelas:

$$FP(\text{positif}) = \sum_i^l E_{li} - TP_i = (24 + 6) - 24 = 6$$

$$FN(\text{positif}) = \sum_i^l E_{il} - TP_i = (24 + 1) - 24 = 1$$

$$\begin{aligned} TN(\text{positif}) &= \sum_i^l \sum_k^l E_{lk} - TP_i - FP_i - FN_i \\ &= (24 + 1 + 6 + 19) - 24 - 6 - 1 = 19 \end{aligned}$$

Perhitungan evaluasinya sebagai berikut:

$$\text{akurasi(positif)} = \frac{24 + 19}{24 + 19 + 6 + 1} = 0,86$$

$$\text{precision(positif)} = \frac{24}{24 + 6} = 0,8$$

$$\text{recall(positif)} = \frac{24}{24 + 1} = 0,96$$

$$F\text{-measure(positif)} = 2 \times \frac{(0,8 \times 0,96)}{(0,8 + 0,96)} = 0,872$$

Perhitungan FP, FN dan TN untuk kelas negatif:

$$FP(\text{negatif}) = \sum_i^l E_{li} - TP_i = (1 + 19) - 19 = 1$$



$$FN(negatif) = \sum_i^l E_{il} - TP_i = (6 + 19) - 19 = 6$$

$$TN(negatif) = \sum_i^l \sum_k^l E_{lk} - TP_i - FP_i - FN_i$$

$$= (24 + 1 + 6 + 19) - 19 - 1 - 6 = 24$$

Perhitungan evaluasinya sebagai berikut:

$$akurasi(negatif) = \frac{24 + 19}{24 + 19 + 6 + 1} = 0,86$$

$$precision(negatif) = \frac{19}{19 + 1} = 0,95$$

$$recall(negatif) = \frac{19}{19 + 6} = 0,76$$

$$F-measure(negatif) = 2 \times \frac{(0,95 \times 0,76)}{(0,95 + 0,76)} = 0,84$$

Perhitungan *macroaverage* parameter evaluasi seluruh kelas:

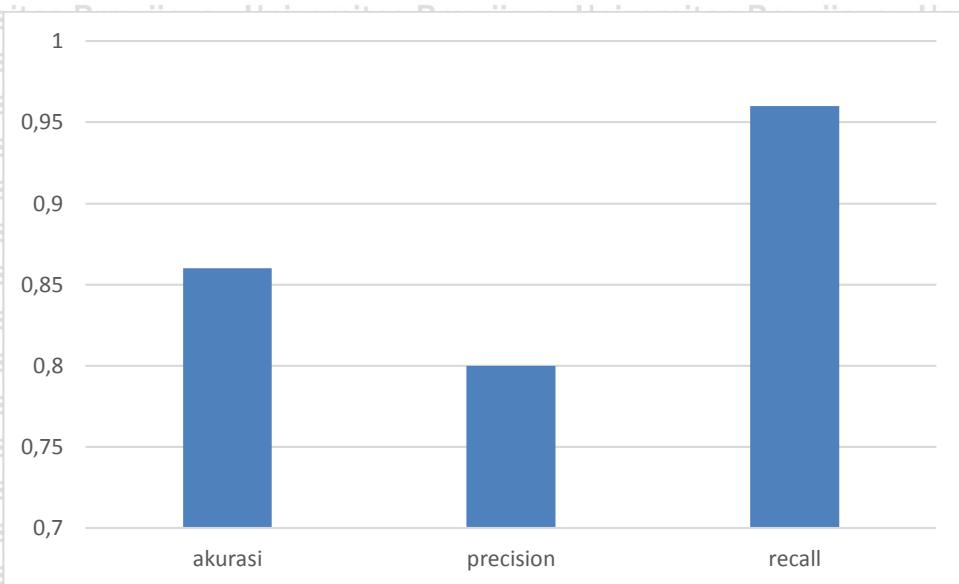
$$akurasi = \frac{0,86 + 0,86}{2} = 0,86$$

$$precision = \frac{0,8 + 0,95}{2} = 0,875$$

$$recall = \frac{0,96 + 0,76}{2} = 0,86$$

$$F-measure = \frac{0,872 + 0,84}{2} = 0,856$$

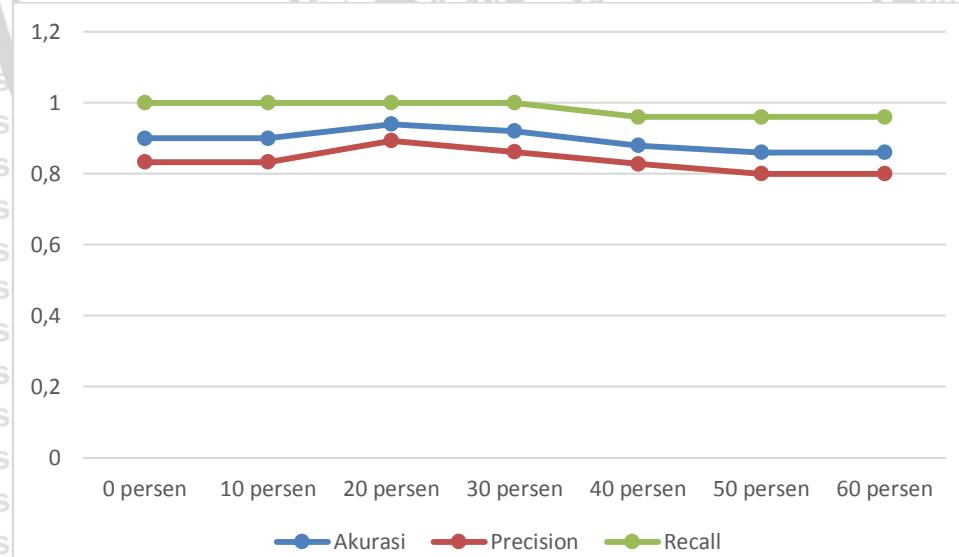
Tabel 6.11 menampilkan hasil perhitungan evaluasi terhadap kinerja sistem dalam bentuk grafik. Perhitungan evaluasi didasarkan pada hasil klasifikasi data uji oleh sistem dilakukan menggunakan *stoplist* 60 persen dengan nilai *ground truth* dari data uji.



Gambar 6.6 Grafik Evaluasi Sistem dengan *Stoplist* 60 Persen

Dari seluruh percobaan, dapat dibuat grafik secara keseluruhan untuk membandingkan hasil evaluasi dari setiap percobaan yang telah dilakukan. Pada hasil keseluruhan tersebut, didapatkan perhitungan *macroaverage* parameter evaluasi terbaik pada klasifikasi dengan menggunakan *stoplist* 20 persen dengan nilai *macroaverage* akurasi sebesar 0,94, *macroaverage* precision sebesar 0,945, *macroaverage* recall sebesar 0,94, dan *macroaverage f-measure* sebesar 0,938.

Pada Gambar 6.7 ditunjukkan grafik hasil percobaan perhitungan parameter evaluasi secara keseluruhan.

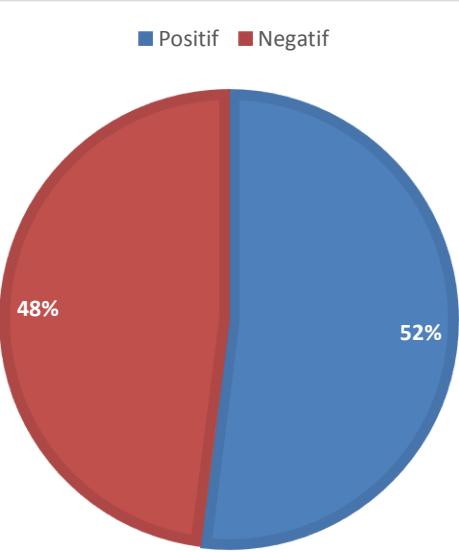


Gambar 6.7 Grafik *Macroaverage* Evaluasi Keseluruhan

6.1.2 Hasil Analisis Sentimen

Untuk melakukan analisis sentimen dilakukan pengujian terhadap 100 data twit tanpa label untuk mengetahui persebaran sentimen dari warganet media sosial Twitter terkait kebijakan pemindahan Ibukota yang telah dicanangkan oleh Presiden Joko Widodo. Berdasarkan hasil pengujian, didapatkan hasil terbaik pada klasifikasi dengan menggunakan *stoplist* 20 persen, sehingga pada proses analisis sentimen akan digunakan *stoplist* 20 persen untuk memperoleh sentimen warganet Twitter berdasarkan 100 data twit tanpa label.

Gambar 6.8 menunjukkan diagram lingkaran hasil klasifikasi data analisis sentimen dengan menggunakan *stoplist* 20 persen. Hasil klasifikasi menunjukkan sentimen antara warganet yang pro maupun kontra terhadap kebijakan pemindahan Ibukota.



Gambar 6.8 Persentase Sentimen Warganet Twitter Berdasarkan Klasifikasi dengan *Stoplist* 20 persen

Berdasarkan hasil klasifikasi sistem terhadap 100 data twit tanpa label terkait kebijakan pemindahan Ibukota, dapat dilihat bahwa perbedaan sentimen antara warganet Twitter terhadap kebijakan pemindahan Ibukota tidak terlalu signifikan. Persentase warganet Twitter yang pro terhadap kebijakan pemindahan Ibukota yaitu sebesar 52 persen. Sementara itu, 48 persen warganet Twitter menyatakan kontra terhadap kebijakan tersebut.

6.2 Pembahasan

Implementasi algortime Naïve Bayes ke dalam kode program untuk membentuk sistem yang ditujukan untuk melakukan analisis sentimen terhadap kebijakan pemindahan Ibukota dapat dilakukan hingga tuntas dalam penelitian ini.

Terdapat beberapa tahap dalam membuat sistem analisis sentimen dalam penelitian ini, yaitu *data scraping*, *preprocessing* data (termasuk pembuatan *stoplist* dengan menggunakan *term-based random sampling* didalamnya), perhitungan *raw term frequency* dan klasifikasi dengan metode Naïve Bayes.

Setelah data uji berhasil diklasifikasikan oleh sistem, kemudian dapat dilakukan analisis sentimen terhadap data-data hasil klasifikasi tersebut. Dari hasil tersebut dapat diketahui apakah warganet media sosial Twitter secara mayoritas menerima atau justru menolak kebijakan pemindahan Ibukota yang telah dicanangkan oleh Presiden Joko Widodo.

Berdasarkan pengujian yang dilakukan terhadap 100 data uji analisis sentimen tanpa label, didapatkan persebaran persentase rata-rata, 52 persen warganet Twitter menyatakan positif terhadap kebijakan pemindahan Ibukota dan 48 persen menyatakan tidak setuju dengan kebijakan tersebut. Hal ini berarti, mayoritas warganet Twitter merasa bahwa kebijakan pemindahan Ibukota dapat memberi dampak baik bagi negara Indonesia dengan alasan-alasan yang diutarakan dalam twit pada media sosial Twitter. Dengan kata lain, kebijakan pemerintah ini mendapat dukungan dari mayoritas warganet Twitter berdasarkan 100 data yang digunakan untuk analisis sentimen tersebut.

Pengujian telah dilakukan untuk mengukur kinerja sistem. Pengujian dilakukan dengan menggunakan *confusion matrix*. Parameter yang digunakan dalam penelitian ini yaitu akurasi, *precision*, dan *recall*. Akurasi dihitung untuk mengetahui tingkat ketepatan antara hasil prediksi sistem dengan nilai aktual. *Precision* dihitung untuk mengetahui ketepatan informasi yang dihasilkan oleh sistem dengan apa yang diminta oleh pengguna. *Recall* dihitung untuk mengukur tingkat keberhasilan sistem dalam menemukan informasi yang dicari oleh pengguna. Sementara itu, *f-measure* merupakan parameter pengukuran yang digunakan untuk mengukur bobot rata-rata dari *precision* dan *recall*. Berdasarkan hasil pengujian sistem, didapatkan hasil *macroaverage* terbaik yaitu pada klasifikasi dengan menggunakan *stoplist* 20 persen, dengan *macroaverage akurasi* sebesar 0,94, *macroaverage precision* sebesar 0,945, *macro average recall* sebesar 0,94, dan *macroaverage f-measure* sebesar 0,938.

Berdasarkan *confusion matrix* pada tahap pengujian, dapat diketahui bahwa sistem dapat melakukan kesalahan klasifikasi data uji. Hal tersebut dapat terjadi karena pada sistem ini digunakan fitur *bag of words* yang berarti hasil klasifikasi bergantung pada jumlah kemunculan *term* data latih pada dokumen data latih. Sering kali *term* pada data uji tidak terdapat dalam *lexicon term* data latih, hal ini juga mendukung terjadinya kesalahan klasifikasi. Selain itu, suatu *term* dapat muncul pada kedua kelas, baik itu kelas positif dan kelas negatif. Hal tersebut mengakibatkan adanya peluang terjadinya kesalahan klasifikasi yang dilakukan oleh sistem.

BAB 7 PENUTUP

7.1 Kesimpulan

1. Penelitian analisis sentimen terkait kebijakan pemindahan Ibukota dilakukan dalam beberapa tahap. Tahap pertama yang dilakukan yaitu *data scraping*. *Data scraping* dilakukan dengan menggunakan *webscraper* dan fitur *search* pada media sosial Twitter dan didapatkan total 450 data twit dengan persebaran, 300 data untuk data latih, 100 data untuk analisis sentimen warganet Twitter terkait kebijakan pemindahan Ibukota dan 50 data untuk data uji. Tahap kedua yaitu *preprocessing*. Tahapan *preprocessing* yang dilakukan antara lain *cleaning*, tokenisasi, *case folding*, normalisasi kata, *filtering* (termasuk pembuatan *stoplist* dengan menggunakan algoritme *term-based random sampling*), *stemming* dan *type*. Setelah dilakukan *preprocessing*, kemudian dilakukan tahap ketiga, yaitu perhitungan *raw term frequency*. *raw term frequency* didapatkan dengan menjumlahkan nilai *term frequency* dari suatu *term* dalam semua dokumen. Tahap terakhir yaitu proses klasifikasi dengan menggunakan metode Naïve Bayes. Hasil akhir klasifikasi berupa nilai probabilitas *posterior* dari data uji. Terdapat 2 nilai probabilitas *posterior* dari perhitungan klasifikasi, yaitu probabilitas *posterior* untuk kelas positif dan negatif. Untuk menentukan hasil klasifikasi dipilih nilai probabilitas *posterior* tertinggi dari suatu data uji untuk menentukan kelas sentimen dari data uji tersebut. Sistem analisis sentimen dibuat dengan menggunakan bahasa pemrograman Python 3.
2. Berdasarkan hasil analisis sentimen terhadap 100 data tanpa label didapatkan kesimpulan bahwa persentase antara warganet media sosial Twitter yang setuju dan warganet media sosial Twitter yang tidak setuju tidak menunjukkan perbedaan yang tidak signifikan. Berdasarkan hasil analisis sentimen sistem, didapatkan warganet Twitter yang menyatakan setuju terhadap kebijakan pemindahan Ibukota sebesar 52 persen, sementara warganet Twitter yang menyatakan tidak setuju terhadap kebijakan tersebut yaitu sebesar 48 persen.
3. Berdasarkan hasil evaluasi sistem, didapatkan nilai *macroaverage* parameter evaluasi terbaik pada klasifikasi dengan menggunakan *stoplist* 20 persen dengan nilai *macroaverage recall* dan akurasi sebesar 0,94, sedangkan nilai *macroaverage precision* sebesar 0,945. Selain itu, sistem juga memperoleh nilai *macroaverage f-measure* sebesar 0,938. Hal ini mengindikasikan, bahwa sistem bekerja lebih baik dengan menggunakan *stoplist* dibandingkan dengan tanpa *stoplist* (*stoplist* 0 persen). Namun, perlu dilakukan pengujian terkait persentase *stoplist* yang digunakan dalam klasifikasi untuk mendapatkan hasil terbaik. Pada penelitian ini dilakukan pengujian *stoplist* 0 persen, 10 persen, 20 persen, 30 persen, 40 persen, 50 persen dan 60 persen. Dari hasil pengujian hasil klasifikasi dengan *stoplist* 20 persen mendapatkan hasil evaluasi terbaik.



7.2 Saran

1. Untuk memperoleh hasil evaluasi sistem yang lebih baik, sebaiknya data latih yang digunakan diperbanyak lagi karena pada sistem ini digunakan fitur *bag of words* yang berarti hasil klasifikasi ditentukan oleh jumlah kemunculan *term* data latih.
2. Menambahkan sentimen netral pada hasil klasifikasi, karena pada penelitian ini hanya digunakan 2 kelas sentimen, yaitu positif dan negatif.
3. Menambahkan fitur yang digunakan untuk proses klasifikasi, karena dalam penelitian ini hanya digunakan fitur *bag of words* untuk menentukan sentimen dari suatu data.



DAFTAR REFERENSI

- Assuja, M. A. & Saniati, 2016. Analisis Sentimen Tweet menggunakan Backpropagation Neural Network. *Jurnal TEKNOINFO*, Volume 10, pp. 23-28.
- BBC, 2019. *BBC Indonesia*. [Online] Available at: <https://www.bbc.com/indonesia/majalah-49469253> [Accessed 17 10 2019].
- Cheng, J. & Greiner, R., 1999. *Comparing Bayesian Network Classifiers*. s.l., s.n., pp. 101-108.
- Hadi, F., 2019. *Warta Kota*: *Tribunnews*. [Online] Available at: <https://warkota.tribunnews.com/2019/09/08/cuma-342-persen-netizen-yang-setuju-ibu-kota-dipindahkan?page=all> [Accessed 16 10 2019].
- Hall, M., 2006. A Decision Tree-Based Attribute Weighting Filter for Naive Bayes. *Knowledge-Based Systems*, Volume 20, pp. 120-126.
- Hananto, B. K., Pinandito, A. & Kharisma, A. P., 2018. Penerapan Maximum TF-IDF Normalization terhadap Metode KNN untuk Klasifikasi Dataset Multiclass Panichella pada Review Aplikasi Mobile. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume 2, pp. 6812-6823.
- Han, J., Kamber, M. & Pei, J., 2012. *Data Mining: Concepts and Techniques*. 3rd ed. San Francisco: Morgan Kaufmann Publishers Inc..
- Indriati & Ridok, A., 2016. Sentiment Analysis for Review Mobile Applications using Neighbor Method Weighted K-Nearest Neighbor (NWKNN). *Journal of Environmental Engineering & Sustainable Technology*.
- Kaplan, A. M. & Haenlein, M., 2010. *Users of the World, Unite! The Challenges and Opportunities of Social Media*. s.l.:Bussiness Horizons.
- Listari, Ihsan, M., Paradistia, E. R. & Widodo, E., 2019. Analisis Sentimen Twitter terhadap Bom Bunuh Diri di Surabaya 13 Mei 2018 menggunakan Pendekatan Support Vector Machine. *PRISMA*, Volume Prosiding Seminar Nasional Matematika 2.
- Lo, R. T., He, B. & Ounis, L., 2005. Automatically Building a Stopword List for an Information Retrieval System. *Journal of Digital Information Management*, Volume III.
- Medhat, W., Hassan, A. & Korashy, H., 2014. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, V(4), pp. 1093-1113.
- Nurhuda, F., Sihwi, S. & Doewes, A., 2013. Analisis Sentimen Masyarakat terhadap Calon Presiden Indonesia 2014 berdasarkan Opini dari Twitter Menggunakan Metode Naive Bayes Classifier. *Jurnal ITSMART*, II(2).



- Pintoko, B. M. & Muslim, K., 2018. Analisis Sentimen Jasa Transportasi Online pada Twitter Menggunakan Metode Naïve Bayes Classifier. *e-Proceeding of Engineering*, V(3), p. 8121.
- Pravina, A. M., Cholissodin, I. & Adikara, P. P., 2019. Analisis Sentimen Tentang Opini Maskapai Penerbangan pada Dokumen Twitter Menggunakan Algoritme Support Vector Machine (SVM). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, III(3), pp. 2789-2797.
- Ratnawati, F., 2018. Implementasi Algoritma Naive Bayes Terhadap Analisis Sentimen Opini Film Pada Twitter. *Jurnal INOVTEK POLBENG*, III(1).
- Rozi, I. F., Pramono, S. H. & Dahlan, E. A., 2012. Implementasi Opinion Mining (Analisis Sentimen) untuk Ekstraksi Data Opini Publik pada Perguruan Tinggi. *Jurnal EECCIS*, VI(1).
- Syah, A. P., Adiwijaya & Faraby, S. A., 2017. Analisis Sentimen pada Data Ulasan Produk Toko Online menggunakan Metode Maximum Entropy. *e-Proceeding of Engineering*, Volume 4, p. 4632.
- Vohra, S. M. & Teraiya, J., 2014. A Comparative Study of Sentiment Analysis Techniques. *Journal of Information, Knowledge and Research in Computer Engineering*, Volume II.