Analisis Kesesuaian Komentar Mahasiswa Pada Sistem Akademi Online Angket Penilaian Dosen Menggunakan *Supervised Model*

Sigit Wijanarko[1]

***Abstract*—The existence of an online academic system of lecturer assessment questionnaire in the Faculty of Computer Science - Budi Luhur University can help the management and lecturers in seeing the lecturer performance index based on student assessment. The questionnaire includes the level of student satisfaction in the form of a Likert scale numerical score against 19 (nineteen) categories, and student comments in natural language which can contain criticisms and suggestions for lecturers. Problems arise in classifying student comments because the writing style is free and not rigid, the online academic system of lecturer assessment questionnaire has not been able to predict the sentiment of student comments and correlation with the category of lecturer assessment. This study tries to classify comments in Indonesian language for comment analysis, which is correlated with the category's lecturer assessment. It is hoped that the results of this study can help the university management and lecturers know the correlation between numerical scores and student comments.**

***Intisari*—Adanya sistem angket penilaian dosen sistem akademik online di Fakultas Ilmu Komputer - Universitas Budi Luhur dapat membantu pihak manajemen dan dosen dalam melihat indeks kinerja dosen berdasarkan penilaian mahasiswa. Kuesioner tersebut meliputi tingkat kepuasan mahasiswa berupa skor numerik skala Likert terhadap 19 (sembilan belas) kategori, dan komentar mahasiswa dalam bahasa natural yang dapat berisi kritik dan saran bagi dosen. Permasalahan muncul dalam pengklasifikasian komentar mahasiswa karena gaya penulisannya yang bebas dan tidak kaku, sistem akademik online angket penilaian dosen belum mampu memprediksi sentimen komentar mahasiswa dan korelasinya dengan kategori penilaian dosen. Penelitian ini mencoba mengklasifikasikan komentar dalam bahasa Indonesia untuk analisis komentar, yang dikorelasikan dengan penilaian kategori dosen. Diharapkan hasil penelitian ini dapat membantu pihak manajemen universitas dan dosen mengetahui hubungan antara nilai numerik dengan komentar mahasiswa.**

***Kata Kunci*—Text classification, Natural Language, Naïve Bayes, K-Nearest Neighbor, Support Vector Machine.**

I. PENDAHULUAN

Evaluasi Kinerja dosen dilakukan tiap akhir semester dalam bentuk: Penilaian Laporan Beban Kerja Dosen yang sesuai aturan Kopertis Wilayah 3 yang terdapat pada Standar Acuan Penilaian Beban Kerja Dosen Kopertis Wilayah III Jakarta; dan Rapor Dosen, terkait dengan penilaian proses pembelajaran oleh mahasiswa yang diisi ketika mereka mengakses sistem akademik online web http://studentpasca.budiluhur.ac.id. Sekretariat mencetak hasil penilaian angket melalui web http://angketmhs.budiluhur.ac.id yang dibandingkan tingkat akurasi untuk mengetahui metode dengan tingkat error rate paling rendah. Berdasarkan metode klasifikasi terbaik tersebut diharapkan dapat memberikan pengetahuan baru berupa kecenderungan positif, netral atau negatif sentimen mahasiswa terhadap proses pembelajaran.

[1] *Jurusan Teknik Informatika, STMIK Antar Bangsa, Kawasan Bisnis CBD Ciledug, Jl. HOS Cokroaminoto, Blok A5 No.29-36, RT.001/RW.001, Karang Tengah, Kota Tangerang, Banten, 15157 INDONESIA (telp:021-50686099; e-mail: sgtwijanarko23@gmail.com*

II. KAJIAN LITERATUR

Cara paling mudah untuk memenuhi persyaratan format penulisan adalah dengan menggunakan dokumen ini sebagai template. Kemudian ketikkan teks Anda ke dalamnya

A. *Kecerdasan Buatan*

Kecerdasan buatan biasanya dilakukan dengan mengikuti atau meniru karakteristik dan analogi pemikiran kecerdasan manusia, dan menerapkannya sebagai algoritma yang dikenal oleh komputer. *Artificial Intelligence* (AI) atau kecerdasan buatan adalah salah satu cabang ilmu yang berhubungan dengan penggunaan mesin untuk memecahkan masalah yang rumit [1].

Dengan Kemampuan menalar dan pengetahuan yang diberikan kepada komputer maka komputer dapat bertindak sebaik dan selayaknya manusia. Agar komputer menjadi pintar maka kecerdasan buatan memberikan dua komponen tersebut. Kecerdasan buatan dapat dilihat dari beberapa sudut pandang yaitu:

1) Dari sudut pandang kecerdasan, kecerdasan buatan adalah bagaimana menjadikan mesin dapat mengerjakan suatu pekerjaan yang awalnya hanya bisa dikerjakan oleh manusia dan membuatnya menjadi pintar.
2) Dari sudut pandang bisnis, Kecerdasan buatan adalah kumpulan dari beberapa alat bantu (*tools*) yang praktis dan untuk menyelesaikan masalah bisnis dengan menggunakan alat-alat bantu tersebut.
3) Dari sudut pandang pemrograman, Kecerdasan buatan adalah studi mengenai pemograman yang meliputi proses pencarian (*search*), simbolik dan pemecahan masalah.
4) Dari sudut pandang penelitian:
   - Menciptakan aplikasi permainan catur, *general problem solving* dan membuktikan suatu teori adalah awal dari penelitian mengenai kecerdasan buatan yang dimulai pada awal tahun 1960-an.
   - *Artificial intelligence* adalah nama pada akar dari studi area.

B. *Konsep Kecerdasan Buatan*

Dalam kecerdasan buatan terdapat beberapa konsep yang harus dipahami yaitu[1]:

1) *Turing Test* – Metode Pengujian Kecerdasan: Nama *turing test* diambil dari pembuatnya yaitu Alan turing,

turing test adalah sebuah metode untuk menguji kecerdasan. Dua objek yang ditanyai dan seorang penanya (manusia) dilibatkan dalam proses uji ini. Yang satu adalah mesin yang akan ditanyai dan yang satunya lagi adalah seorang manusia.Penanya tidak bisa melihat langsung terhadap objek yang akan ditanyai. Membedakan jawaban antara mana jawaban manusia dan mana jawaban komputer berdasarkan jawaban kedua objek tersebut adalah tugas dari penanya. Apabila jawaban komputer dan jawaban manusia tidak dapat dibedakan oleh penanya maka dapat ditarik kesimpulan bahwa CERDAS.

2) Pemrosesan Simbolik: Mengerjakan proses nonalgoritmik dan simbolik dalam suatu penyelesaian masalah adalah sifat penting dari kecerdasan buatan. Tidak berdasarkan pada komputasi matematika atau mengacu kepada rumus adalah kecenderungan manusia dalam menyelesaikan masalah karena manusia lebih bersifat simbolik. Sementara komputer semula diciptakan untuk memproses suatu bilangan atau angka-angka. Kecerdasan buatan merupakan cabang ilmu komputer yang mengerjakan proses nonalgoritmik dan simbolik dalam suatu penyelesaian masalah adalah sifat penting dari kecerdasan buatan.
3) *Heuristic*: Istilah *heuristic* berasal dari bahasa Yunani yang mempunyai makna menemukan. *Heuristic* adalah cara untuk melakukan proses pencarian (*search*) secara selektif ruang permasalahan (*problem*), dan memandu proses pencarian yang kita lakukan disepanjang jalur yang memiliki kemungkinan sukses paling besar.
4) Penarikan Kesimpulan: Kemampuan mempertimbangkan (*reasoning*) atau kemampuan berfikir dicoba dibuat oleh kecerdasan buatan atau AI. Kemampuan berfikir (*reasoning*) termasuk di dalamnya penarikan kesimpulan (*inferencing*) berdasarkan aturan dengan memakai metode *heuristic* atau pencarian lainnya dan berdasarkan fakta-fakta.
5) Pencocokan Pola (*Pattern Matching*): Cara kerja kecerdasan buatan adalah dengan menggunakan metode pencocokan pola (*pattern matching*) yang berusaha untuk menjelaskan kejadian (*event*), proses atau objek, dalam hubungan logika atau komputasional.

C. *Lingkup Utama Kecerdasan Buatan*

Lingkup utama kecerdasan buatan adalah:

1) Sistem pakar (*ExpertSystem*): Komputer dipergunakan menjadi alat untuk menyimpan pengetahuan para pakar. Dengan begitu komputer akan mempunyai kepandaian agar dapat menyelesaikan suatu masalah dengan meniru kepakaran atau kemampuan yang dipunyai oleh para pakar.
2) Pengolahan bahasa alami (*Natural Language Processing*): Dengan bahasa sehari-hari pengguna bisa berkomunikasi dengan komputer adalah harapan dari pengolahan bahasa alami ini.
3) Pengenalan ucapan (*Pattern Recognition*): Dengan memakai suara diharapkan manusia bisa berkomunikasi dengan komputer melalui pengenalan ucapan.
4) Sistem sensor dan robotika
5) *Computer vision*: Objek-objek visual atau gambar dapat di intrepetasikan melalui computer.
6) *Intelligent Computer aid Instruction*: Penggunaan komputer sebagai tutor yang mempunyai keahlian dalam melatih dan mengajar.

D. *Bahasa Alami*

Bentuk representasi dari suatu pesan yang akan dikomunikasikan antara manusia adalah pengertian bahasa alami. Bentuk representasi utamanya adalah berupa ucapan/suara (*spoken language*), akan tetapi sering dinyatakan dalam bentuk tulisan. Jenis bahasa dapat dipisahkan menjadi (1) bahasa buatan dan (2) bahasa alami. Bahasa buatan merupakan bahasa yang diciptakan secara spesifik untuk memecahkan kebutuhan tertentu, seperti bahasa pemograman komputer atau bahasa pemodelan. Bahasa alami adalah bahasa untuk berkomunikasi sesama manusia yang biasa digunakan, seperti bahasa Inggis, Indonesia, jawa dan sebagainya.

Pertama kali orang merepresentasikan bahasa untuk rangkaian simbol adalah Chomsky. Dia berhasil membuktikan atau menunjukan bahwa segala sesuatu dapat direpresentasikan memakai cara yang lebih umum atau universal. Dari hasil Chomsky yang mengatur susunan simbol-simbol dan merepresentasikan bahasa sebagai sekumpulan simbol-simbol tersebut maka peluang pemrosesan bahasa secara simbolik dengan komputer peluangnya terbuka, sehingga mencetuskan cabang ilmu baru yaitu *Natural Language Processing* (NLP).

Leksikon dan perbendaharaan kata adalah pembahasan salah satu bidang ilmu *linguistic*. *Linguistic* sendiri adalah cabang ilmu komputer yang secara spesifik mengkaji bagaimana suatu bahasa itu distrukturkan. Leksikon adalah kamus yang mendaftar kata-kata bahasa itu secara alfabet. Perbendaharaan kata adalah sekumpulan kata-kata dan frase-frase yang digunakan dalam bahasa tertentu. Sebagai bagian dari pengkajian bahasa, linguis mendefinisikan semua kata-kata dan frase-frase yang digunakan secara umum kemudian mengorganisasikannya ke dalam sebuah leksikon.

E. *Pemrosesan Bahasa Alami (Natural Language Processing)*

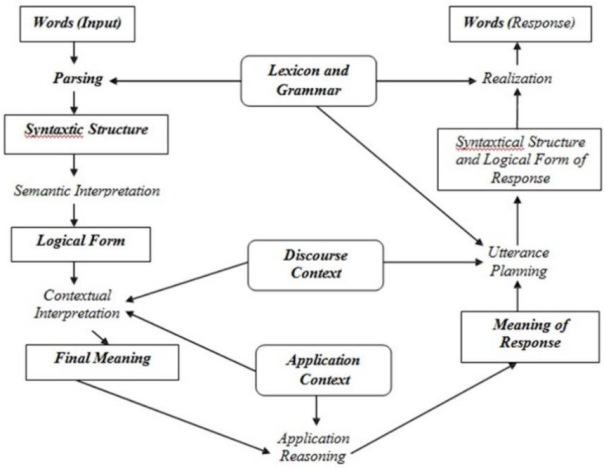
Pemrosesan Bahasa Alami tidak bertujuan untuk mengubah bahasa yang diterima dalam bentuk suara ke dalam data digital dan / atau sebaliknya, tetapi lebih bertujuan untuk memahami makna dari teks yang diberikan dalam format bahasa alami dan merespon dengan tepat, misalnya dengan melakukan tindakan spesifik atau menampilkan data tertentu. *Natural Language Processing* (NLP) adalah salah satu bidang kecerdasan buatan (*Artificial Intelligence*) yang mempelajari komunikasi antara manusia dan computer.

Teknik NLP memungkinkan komputer untuk memproses dan memahami bahasa alami manusia dan memanfaatkannya

lebih lanjut untuk memberikan hasil yang bermanfaat. Hal ini membuat NLP berkaitan dengan area *Human-Computer Interaction* (HCI). Hal tersebut terutama berkaitan dengan merancang dan membangun aplikasi dan sistem yang memungkinkan interaksi antara mesin dan bahasa alami dan memanfaatkannya lebih lanjut agar dapat digunakan oleh manusia. NLP didefinisikan sebagai bidang khusus ilmu komputer dan teknik dan kecerdasan buatan dengan akar dalam linguistik komputasional[2].

Ketika komputer telah memahami ucapan yang diberikan pengguna, maka komputer dapat melakukan hal-hal yang diharapkan pengguna / tanggapan kembali diungkapkan atau diungkapkan dalam bahasa alami juga. Pelacakan klasik dan teknik pencocokan pola digunakan bersama dengan basis pengetahuan sehingga komputer dapat memahami apa yang dimasukkan pengguna dalam bahasa alami. Agar komputer memahami pertanyaan dalam bahasa alami, komputer harus memiliki pengetahuan analitis dan interpretasi masukan dalam data *knowledgable*-nya. Dalam hubungan ini teknik AI digunakan untuk menampilkan pengetahuan internal dan masukan proses. Komputer harus memahami gramatika dan definisi kata-kata. Untuk mencapai tujuan tersebut dibutuhkan tiga tahap proses. Proses yang pertama adalah parsing atau analisa sintaksis yang memeriksa kebenaran struktur kalimat berdasarkan suatu *grammar* (tata bahasa) dan *lexicon* (kosa kata) tertentu.

Proses kedua adalah *semantic interpretation* (interpretasi semantik) yang bertujuan untuk merepresentasikan arti dari kalimat secara contextindependent untuk keperluan lebih lanjut. Sedangkan proses ketiga adalah contextual interpretation atau interpretasi kontekstual yang bertujuan untuk merepresentasikan arti secara *context-dependent* dan menentukan maksud dari penggunaan kalimat. Gambaran sebuah organisasi sistem NLP dapat dilihat pada Gbr. 1.



Gbr. 1 gambaran umum sistem NLP

Jenis aplikasi yang bisa dibuat pada bidang bidang *natural language* adalah *text - based application* dan *dialogue-based applications*.

1) *Text - based application*: Meliputi berbagai aplikasi yang memproses teks tertulis seperti misalnya buku, berita di surat kabar, *e-mail* dan lain sebagainya. Contoh penggunaan dari *text-based* application ini adalah:
   - Mencari topik tertentu dari buku yang ada pada perpustakaan.
   - Memberikan respon atas input yang diberikan.
   - Mencari isi dari surat atau *e-mail*.
   - Menterjemahkan dokumen dari satu bahasa ke bahasa yang lain.
2) *Dialogue-based application*: Idealnya pedekatan ini melibatkan bahasa lisan atau pengenalan suara,akan tetapi bidang ini juga memasukkan interaksi dengan cara memasukkan teks pertanyaan melalui*keyboard*. Aplikasi yang sering ditemui untuk bidang ini adalah:
   - Sistem tanya jawab, dimana natural language digunakan dalam mendapatkan informasi dari suatu database.
   - Sistem otomatis pelayanan melalui telepon.
   - Kontrol suara pada perlatan sistem sistem *problem solving* yang membantu untuk melakukan penyelesaian masalah yang umum dihadapi dalam suatu pekerjaan.

F. Penelitian Terdahulu

Penelitian yang dilakukan oleh Samuel Cunningham-Nelson dengan judul *Linking Numerical Scores with Sentiment analysis of students' teaching and subject evaluation surveys* dengan perbandingan metode *machine learning supervised model* antara *Linear Regression*, *kSupport Vector Machine* dan *Naïve Bayes* untuk polaritas sentimen komentar siswa dengan menggunakan *dictionary* sebagai *feature extraction* menunjukan dengan menggunakan *dictionary* (*eksternal*, *internal* dan *combined dictionary*) memberikan metode *Support Vector Machine* unggul dari pengklasifikasi *Naïve Bayes* dan *Linear Regression*, yaitu 57.70% di banding 39.67% dan 53.54%.

Penilitian yang dilakukan oleh Muhammad Rezwanul Huq, Ahmad Ali, dan Anika Rahman dengan judul *Sentiment Analysis on Twitter Data using KNN and SVM*[10] dengan perbandingan metode *machine learning supervised model* antara *K-Nearest Neighbor*, dan *Support Vector Machine* untuk polaritas sentimen komentar pada twitter dengan menggunakan *Word Base*, *N-Gram,Pattern*, *Punctuation*, *Key-based* sebagai *feature extraction* menunjukkan dengan menggunakan *Word Base feature extraction* memberikan metode *K-Nearest Neighbor* ungggul dari pengklasifikasi *Support Vector Machine*, yaitu 84.32% dibanding 77.97%.

Penelitian yang dilakukan oleh Muhammad Aman Ullah dengan judul *Sentiment Analysis of Students Feedback: A Study towards Optimal Tools*[11] dengan perbandingan metode *machine learning supervised model* antara metode *Support Vector Machine*, *Naïve Bayes*, *Complement Naïve Bayes* dan *Maximum Entropy* untuk polaritas sentimen

komentar siswa dengan menggunakan *Term Co-Occurance* dan *Opinion Word* sebagai *feature extraction* menunjukkan dengan menggunakan *Preprocessing* P5 memberikan metode *Support Vector Machine* ungggul dari *Maximum Entropy, Naïve Bayes, dan Complement Naïve Bayes*, yaitu 96.2% dibanding 87.1%, 89.1% dan 93.1%.

III. METODE PENELITIAN

Penelitian ini dilakukan dengan menggunakan metode eksperimen. Metode eksperimen ini dilakukan peneliti dengan memanipulasi kondisi sesuai dengan kebutuhan permasalahan yang dihadapi di dalam penelitian. Dengan memanipulasi kondisi ini, nantinya hasil dari penelitian ini akan menghasilkan suatu informasi polaritas sentiment terhadap dosen berdasarkan komentar mahasiswa, yang mencapai tingkat akurasi di atas 80% dihitung berdasarkan *confusion matrix*.

A. *Pengumpulan Data*

Dataset dalam penelitian ini diambil dari angket penilaian dosen yang telah diisi oleh mahasiswa pada sistem akademik online Universitas Budi Luhur dengan alamat http://studentpasca.budiluhur.ac.id. Dari data tersebut akan diambil lebih dari 1000 angket yang diisi mahasiswa dalam tahun ajar 2018/2019 Gasal dengan klasifikasi sebagai berikut:

1) Jika skor nilai angket adalah Baik (B) atau Baik Sekali (BS), berarti polaritas kategori tersebut adalah positif.
2) Jika skor nilai angket adalah Kurang (K) atau Kurang Sekali (KS), berarti polaritas kategori tersebut adalah Negatif.
3) Jika skor nilai angket adalah Cukup (C), berarti polaritas kategori tersebut adalah netral.

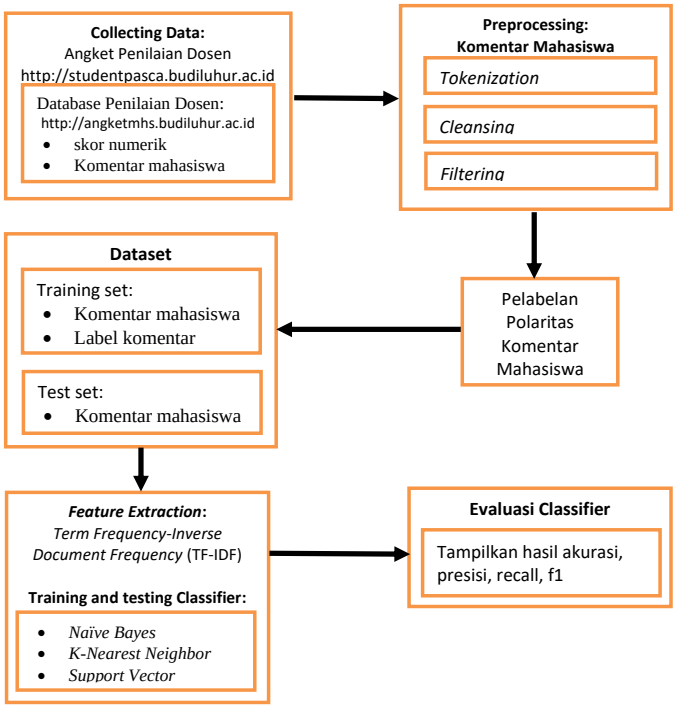
B. *Sistem Yang di Usulkan*

Pengumpulan data komentar mahasiswa dari database sistem akademik online dengan cara *query* kepada database, Selanjutnya data hasil *query* tersebut akan mengalami proses *preprocessing*. Pada tahap *preprocessing* ini dilakukan pembersihan data komentar yang terdiri dari *case folding*, tokenisasi, konversi *slangword*, penghapusan *stopword*.

Kemudian data komentar hasil *preprocessing* disimpan dalam database dengan tempat yang berbeda dari data komentar asli. Kemudian data angket hasil *preprocessing* yang sudah diberi label secara manual akan dirubah menjadi *vector* dengan *feature extraction Term Frequency-Inverse Document Frequency* (TD-IDF) dan menggunakan metode *Naïve Bayes*, *K-Nearest Neighbor* dan *Support Vector Machine* menghasilkan model klasifikasi polaritas sentimen.

Kemudian tahap terakhir adalah pengujian klasifikasi polaritas komentar yang meliputi pengujian akurasi, presisi, recall, dan f1-score yaitu menghitung kesesuaian komentar pada system.

Pada penelitian ini penulis mengusulkan secara umum arsitektur sistem ini terdiri dari lima bagian diantaranya adalah pengumpulan data, preprocessing, pelabelan, klasifikasi sentimen, hasil akurasi sentimen. Arsitektur sistem yang dirancang seperti terlihat pada Gbr. 2.



Gbr. 2 rancangan usulan sistem

C. *Pre-Processing*

Setelah dilakukan label data, tahap selanjutnya yang harus dilakukan adalah praproses. Data yang diambil adalah berupa komentar yang merupakan bagian dari angket penilaian dosen pada sistem akademik online Universitas Budi Luhur. Data berupa komentar yang berisikan kritik dan saran untuk dosen dalam suatu periode. Data tersebut sesuai usulan penulis akan dilakukan proses ekstraksi data sebagai berikut:

1) Tokenisasi (*Tokenization*): Tokenisasi adalah sebuah proses yang dilakukan untuk memotong atau memecah kalimat menjadi beberapa bagian atau kata (Manning et al., 2009). Hasil dari pemotongan ini disebut dengan token. Model tokenization yang digunakan yaitu n-gram.
2) *Cleansing*: *Cleansing* adalah suatu tahap di mana karakter maupun tanda baca yang tidak diperlukan dibuang dari teks. Contoh karakter yang dibuang adalah tanda seru, tanda tanya, koma dan titik.
3) *Filtering*: *Filtering* adalah tahap menghilangkan kata-kata yang muncul dalam jumlah besar, namun dianggap tidak memiliki makna (*stop words*). Pada dasarnya, *stop words list* adalah sekumpulan kata- kata yang banyak digunakan dalam berbagai bahasa. Alasan penghapusan *stop words* dalam banyak program aplikasi yang berkaitan dengan *text mining* adalah karena penggunaannya yang terlalu umum, sehingga pengguna dapat berfokus pada kata-kata lain yang jauh lebih penting [3].

D. Pembobotan Kata

Pembobotan kata adalah suatu mekanisme untuk memberikan skor terhadap frekuensi kemunculan sebuah kata dalam dokumen teks. Salah satu metode populer untuk melakukan pembobotan kata adalah TF-IDF (*Term Frequency-Inverse Document Frequency*). *Term Frequency–Inverse Document Frequency* adalah sebuah metode pembobotan yang menggabungkan dua konsep, yaitu *Term Frequency* dan *Document Frequency* (Asrofi, 2015). Term Frequency adalah konsep pembobotan dengan mencari seberapa sering (frekuensi) munculnya sebuah term dalam satu dokumen [4].

Dikarenakan setiap dokumen memiliki panjang yang berbeda-beda, bisa saja terjadi sebuah kata muncul lebih banyak di dokumen yang panjang dibandingkan dengan dokumen-dokumen yang pendek. Dengan demikian, term frequency sering dibagi dengan panjangnya dokumen (total kata yang ada di dokumen tersebut), dalam hal ini dokumen adalah komentar mahasiswa.

Sedangkan *Document Frequency* adalah banyaknya jumlah dokumen di mana sebuah term itu muncul[4]. Semakin kecil frekuensi kemunculannya, maka semakin kecil pula nilai bobotnya. Ketika proses perhitungan term frequency, semua kata di dalamnya dianggap sama pentingnya. Namun, terdapat kata yang sebenarnya kurang penting dan tidak perlu diperhitungkan (stopword) seperti “di-”, “ke-”, “dan” dan lain sebagainya. Oleh sebab itu, kata-kata yang kurang penting tersebut perlu dikurangi bobotnya dan menambah bobot kata penting lainnya.

Skor TF-IDF dapat diperoleh menggunakan Persamaan berikut:

$$tf - idf_{t,d} = tf_{t,d} * idf_t$$

E. Klasifikasi

Pengklasifikasian data dilakukan terhadap dataset berdasarkan atribut yang dimiliki oleh dataset. Pada penelitian ini penulis menggunakan metode *supervised model* dengan algoritma yaitu SVM, Naïve Bayes dan *K-Nearest Neighbor* (K-NN).

Masing-masing algoritma memiliki kelebihan dan kekurangan sebagai berikut:

1) SVM
   - Kelebihan: mempunyai generalisasi data yang tinggi. Mampu menghasilakan model klasifikasi yang baik meskipun dilatih dengan dengan himpunan data yang relatif sedikit (disbanding ruang masalah yang harus diselesaikan) hanya dengan dengan pengaturan parameter yang sederhana. Relatif mudah diimplementasikan karena penentuan support vector dapat dirumuskan dalam masalah *Quadratic Programming* (QP). Maka dengan mudah mengimplementasikan SVM.
   - Kekurangan: sulit diaplikasikan untuk himpunan data dengan jumlah sampel dengan dimensi yang sangat besar. Umumnya hanya diformulasikan untuk menyelesaikan masalah klasifikasi dua kelas. Walupun dapat dikembangkan untuk menyelesaikan masalah klasifikasi multi kelas, namun masing-masing strategi multi class SVM juga memiliki kelemahan.
2) K-NN
   - Kelebihan: kuat dalam melatih data yang noisy, sangat efektif dengan jumlah data yang besar, dan mudah diimplementasikan.
   - Kekurangan: diperlukan penentuan nilai parameter K, sensitif pada data pencilan, dan kerentanan pada variable yang non-informatif.
3) *Naive Bayes*
   - Kelebihan: diperlukan data yang sedikit untuk klasifikasi, cepat, efisien, dan mudah diimplementasikan, kuat pada atribut yang tidak relevan.
   - Kekurangan: akurasi berkurang karena independence antar atribut, tidak berlaku jika nilai probabilitasnya adalah nol (0) / *Zero Frequency*.

F. Support Vector Machine (SVM)

SVM adalah perhitungan pembelajaran mesin yang umum (siap untuk menangani berbagai macam informasi) dengan menggunakan pembatas linear sebagai dasarnya. Meskipun demikian, tidak semua informasi dapat dipecah secara langsung dalam dua ukuran. Dengan cara ini, pekerjaan pembatas linear kemudian diubah menjadi hyperplanes dengan memakai metode fungsi kernel sehingga hyperplanes dapat memecah data dalam ruang dimensi yang lebih tinggi.

Dalam SVM, fungsi pemisah bermaksud untuk menentukan kelas. Bidang pemisah pendukung kelas +1 dan bidang pemisah pendukung kelas -1. Secara numerik, menemukan pembagi terbaik sama dengan memaksimalkan margin antara kedua kelas. Memaksimalkan margin antara dua kelas setara dengan membatasi fungsi pembatas dengan memperhatikan pemisah. Bidang pemisah terbaik adalah bidang pemisah yang memberikan nilai margin terbesar dan berada di pusat antara dua set objek dari dua. Nilai margin adalah jarak antara spacer dan komponen eksternal element dari dua kelas tersebut. Untuk situasi ini pemisah yang dicari adalah fungsi linear [5] seperti berikut:

$$f(x) = sign(w^T x_i + b = 0)$$

Dengan nilai W adalah bobot yang merepresentasikan posisi *hyperplane* pada bidang normal, X adalah vektor data masukan. B adalah bias yang merepresentasikan posisi bidang relatif terhadap pusat koordinat. Selanjutnya. Selanjutnya data dikelompokkan dengan menggunakan fungsi pemisah yang sudah ditemukan, di mana untuk menentukan kelasnya: Fungsi w.xi + b = +1 adalah bidang pemisah pendukung dari kelas +1. Fungsi w.xi + b = −1 adalah bidang pemisah pendukung dari kelas −1. Bidang pemisah terbaik ekuivalen dengan memaksimalkan margin antara dua kelas yang dihitung dengan formula 2‖w‖ 2. Memaksimalkan margin

antara kedua kelas sama dengan meminimumkan fungsi tujuan 1 2‖w‖ 2 dengan memperhatikan pembatas yi(w·xi+b) ≥1 dengan xi adalah data input dan yi adalah keluaran dari data xi. Selanjutnya, masalah klasifikasi diformulasikan ke dalam quadratic programming (QP) yang diselesaikan dengan *lagrange multiplier*[6].

G. K-Nearest Neighbor (K-NN)

K-Nearest Neighbor menunda proses pemodelan data pelatihan sampai dibutuhkan untuk mengklasifikasikan sampel data uji. Sampel data latih dijelaskan oleh atribut-atribut numerik pada n-dimensi dan disimpan dalam ruang n-dimensi. Klasifikasi *nearest-neighbor* didasarkan pada pembelajaran dengan analogi, yaitu dengan membandingkan *data testing* dengan *data training* yang mirip [6](Han dkk, 2012). *K-Nearest Neighbor* merupakan salah satu metode *machine learning* yang melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Adapun rumus K-NN yang digunakan:

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} \left(a_r(x_i) - a_r(x_j)\right)^2}$$

Keterangan:

$d(x_i, x_j)$ = Jarak *Euclidean (Euclidean Distance)*

$(x| \quad |i)$ = *record* ke-i; $(x| \quad |j)$ = *record* ke-j

$(a| \quad |r)$ = data ke-r; i, j = 1, 2, 3, … , n

Langkah-langkah dalam menghitung metode Algoritma K-NN:

1) Menentukan Parameter K (Jumlah tetangga paling dekat).
2) Menghitung kuadrat jarak *euclid* (*queri instance*) masing-masing objek terhadap data sampel yang diberikan.
3) Kemudian mengurutkan objek-objek tersebut ke dalam kelompok yang mempunyai jarak *euclid* terkecil.
4) Mengumpulkan kategori Y (Klasifikasi *Nearest Neighbor*).
5) Dengan menggunakan kategori *Nearest Neighbor* yang paling mayoritas maka dapat diprediksi nilai *query instance* yang telah dihitung.

Algoritma K-NN mempunyai kelebihan dan kekurangan sebagai berikut:

1) Kelebihan K-NN: Kuat dalam melatih data yang noisy, sangat efektif dengan jumlah data yang besar, dan mudah diimplementasikan.
2) Kekurangan K-NN: diperlukan penentuan nilai parameter K, sensitif pada data pencilan, dan kerentanan pada variable yang non-informatif.

H. Naïve Bayes

Naive Bayes merupakan sebuah pengklasifikasian probabilistik sederhana yang menghitung sekumpulan probabilitas dengan menjumlahkan setiap frekuensi dan kombinasi nilai dari dataset yang diberikan. *Naive Bayes* didasarkan pada asumsi penyederhanaan bahwa nilai atribut secara kondisional saling bebas jika diberikan nilai *output*. Keuntungan penggunaan *Naive Bayes* adalah bahwa metode ini hanya membutuhkan jumlah data pelatihan (*Training Data*) yang kecil untuk menentukan estimasi parameter yang diperlukan dalam proses pengklasifikasian. *Naive Bayes* sering bekerja jauh lebih baik dalam kebanyakan situasi dunia nyata yang kompleks dari pada yang diharapkan.

$$V_{MAP} = argmax_{v_j \epsilon v} P(v_j) \prod_{i=1}^{n} P(a_i|v_j)$$

Keterangan

a_i = atribut atau fitur ke-i

v_j = kelas ke-j (positif atau negatif)

V = himpunan kelas target

V_{MAP} = kelas sentimen suatu komentar

Menghitung probabilitas $P(v_i)$ ditentukan pada saat pelatihan, yang nilainya didekati dengan:

$$P(v_j) = doc_j \vee \frac{a_i}{|contoh|}$$

Dimana $|doc_j|$ adalah banyaknya dokumen yang memiliki kategori j dalam pelatihan dan |contoh| adalah banyaknya dokumen dalam contoh yang digunakan untuk pelatihan. Untuk nilai $P(a_i \mid v_j)$, yaitu menentukan probabilitas kata a_i dalam kategori j ditentukan dengan persamaan berikut :

$$P(a_i|v_j = \frac{n_i + 1}{n + vocabulary \vee}$$

Berdasarkan persamaan, n_i adalah frekuensi munculnya kata a_i dalam dokumen berkategori v_j, sedangkan nilai n adalah banyaknya seluruh kata dalam berkategori v_j, dan |*vocabulary*| adalah banyaknya kata dalam contoh pelatihan. Berdasarakan persamaan diatas, dapat dilihat bahwa setiap atribut atau fitur diasumsikan tidak memiliki keterhubungan satu sama lainnya. *Naïvebayes* menggunakan asumsi dalam sebuah dokumen kemunculan kata tidak mempengaruhi kemunculan kata yang lain. Meskipun asumsi ini bertentangan dengan aturan bahasa, namun tidak mengurangi keakuratan metode *Naïve Bayes* [7](Nur, 2011).

IV. HASIL DAN PEMBAHASAN

Analisis sistem dilakukan untuk mengetahui kebutuhan yang diperlukan untuk membangun sistem prediksi polaritas sentimen berdasarkan komentar mahasiswa. Selain itu, analisis sistem juga akan berguna untuk perancangan sistem prediksi polaritas sentimen berdasarkan komentar mahasiswa.

Pada sistem prediksi polaritas sentimen berdasarkan komentar mahasiswa ini dibuat berdasar perhitungan pembobotan kemunculan kata pada komentar dan pada seluruh komentar. Proses yang akan dilakukan untuk membuat sistem prediksi adalah proses pelatihan yang akan dilanjutkan dengan proses pengujian untuk mendapatkan hasil prediksinya.

Data komentar yang digunakan pada sistem ini terbatas yaitu terdiri dari 1323 komentar mahasiswa dalam bahasa Indonesia. Sentimen komentar mahasiswa terdiri dari 3

polaritas yang berbeda dan tersebar di 19 kategori penilaian dosen.

ID	Komentar	Tagging
1	sangat memberikan motivasi untuk mahasiswa dalam berusaha dan berbisnis karir sendiri	Positif
2	baik dalam mengajar dan memiliki wawasan luas	Positif
.....	.....	.....
.....	.....	.....
.....	.....	.....
1322	kurang baik dalam membagi waktu antara dosen 1 dan dosen 2	Negatif
1323	tidak ada karena blm pernah di ajar beliau	Netral

Gbr. 3 data yang digunakan pada penelitian

A. *Implementasi*

Pada tahap ini berisi hasil implementasi dari proses yang telah dirancang sebelumnya yaitu *preprocessing*, ekstraksi fitur (*feature extraction*), dan implementasi proses klasifikasi untuk prediksi polaritas komentar mahasiswa. Seluruh dataset dipanggil dan disimpan dalam variabel dengan namakomentar_mahasiswa. Kode untuk proses impor konten adalah seperti terlihat pada Gbr. 3 berikut:

```
# importing the dataset
import pandas as pd
file_input = './data/komentar_mahasiswa.csv'
komentar_mahasiswa = pd.read_csv(file_input, sep=";")
```

Gbr. 4 kode python impor komentar mahasiswa

Proses *stemming* digunakan untuk mengubah kata-kata dalam kalimat dataset menjadi kata dasar dalam morfologi bahasa Indonesia. Dalam penelitian ini, proses *stemming* dilakukan dengan menggunakan *Stemmer StemmerFactory* yang merupakan pustaka *Sastrawiopen-source*, proses yang dimaksud dapat dilihat pada Gbr. 5 dan Gbr. 6.

```
import lxml.html
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
# create stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()
processed_features = []
for sentence in range(0, len(features)):
    # remove html tags
    processed_feature = lxml.html.fromstring(features[sentence]).text_content()
    # stemming
    processed_feature = stemmer.stem(processed_feature)
    # add to the list
    processed_features.append(processed_feature)
print('Sebelum di stemming: ', features[0])
print('Setelah stemming: ', processed_features[0])
```

Gbr. 5 kode proses *stemming*

	Sebelum di stemming	Setelah stemming
0	YANG DI AJARKAN MUDAH DIPAHAMI	YANG DI AJAR MUDAH PAHAM
1	WALUPUN JAUH JARAK SANGAT KOMPETEN SEKALI	WALUPUN JAUH JARAK SANGAT KOMPETEN SEKALI
2	WAKTUNYA YANG TIDAK KONSISTEN DENGAN JADWAL PERKULIAHAN	WAKTU YANG TIDAK KONSISTEN DENGAN JADWAL KULIAH
3	WAKTU MENGAJAR SERING KELEWATAN TAPI TIDAK MASALAH SIH KARENA KELAS JUGA JADI MENYENANGKAN	WAKTU AJAR SERING LEWAT TAPI TIDAK MASALAH SIH KARENA KELAS JUGA JADI SENANG
4	VOKAL PERLU DIKERASKAN	VOKAL PERLU KERAS
5	UNTUK SELALU DI TINGKATKAN DAN DI PERTAHANKAN.	UNTUK SELALU DI TINGKAT DAN DI TAHAN
6	UNTUK PENJELASAN PERHITUNGAN SEBAIKNYA LEBIH DETIL	UNTUK JELAS HITUNG BAIK LEBIH DETIL
7	UNTUK MENDATANGKAN DOSEN TAMU KEDEPANNYA TETAP DIPERTAHANKAN	UNTUK DATANG DOSEN TAMU DEPAN TETAP TAHAN
8	UNTUK MASALAH UJIN AGAR TIDAK TERLALU SUSAH KARENA KELAS SAYA KELAS KARYAWAN	UNTUK MASALAH UJIN AGAR TIDAK TERLALU SUSAH KARENA KELAS SAYA KELAS KARYAWAN
9	UNTUK DOSEN LEBIH BERSEMANGAT MENGAJARNYA DAN LEBIH BANYAK MENDAPAT ILMU JIKA DENGAN PEMETAAN PIKIRAN DARI PADA PAKAI SLIDE PPT UNTUK UBL MOHON JANGAN DIPAKSAKAN DOSEN UNTUK MENGGUNAKAN POWERPOINT	UNTUK DOSEN LEBIH SEMANGAT AJAR DAN LEBIH BANYAK DAPAT ILMU JIKA DENGAN META PIKIR DARI PADA PAKAI SLIDE PPT UNTUK UBL MOHON JANGAN PAKSA DOSEN UNTUK GUNA POWERPOINT
10	TUGASNYA BOLEH UNTUK LEBIH DI PERJELAS	TUGAS BOLEH UNTUK LEBIH DI JELAS
11	TUGAS YANG DIBERIKAN SESUAI DENGAN KAMAMPUAN MAHASISWA	TUGAS YANG BERI SESUAI DENGAN KAMAMPUAN MAHASISWA
12	TUGAS YANG DIBERIKAN SANGAT TEPAT UNTUK MAHASISWA	TUGAS YANG BERI SANGAT TEPAT UNTUK MAHASISWA
13	TUGAS TUGAS MATA KULIAHNYA CUKUP MEMBANTU PENAMBAHAN PENGETAHUAN	TUGAS TUGAS MATA KULIAH CUKUP BANTU TAMBAH TAHU
14	TUGAS TUGAS MATA KULIAH KALO BISA DITAMBAH	TUGAS TUGAS MATA KULIAH KALO BISA TAMBAH
15	TUGAS TERLALU BERAT DAN PADAT	TUGAS TERLALU BERAT DAN PADAT
16	TRIMAKASIH BAPAK ATAS ILMU YG TELAH BAPAK BERIKAN SMOGA BAPAK SEHAT DAN SUKSES SLALU BAIK DUNIA DAN AKHIRAT	TRIMAKASIH BAPAK ATAS ILMU YG TELAH BAPAK IKAN SMOGA BAPAK SEHAT DAN SUKSES SLALU BAIK DUNIA DAN AKHIRAT
17	TOLONG UNTUK TUGAS JANGAN TERLALU SULIT	TOLONG UNTUK TUGAS JANGAN TERLALU SULIT

Gbr. 6 hasil proses *stemming*

Pada proses *case folding*, variabel features dilakukan proses *case folding* sehingga semua kata dalam dataset berubah menjadi kata dengan huruf kecil. Proses yang dimaksud seperti terlihat pada Gbr.7 dan Gbr. 8.

```
import lxml.html
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory

# create stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()

processed_features = []

for sentence in range(0, len(features)):
    # remove html tags
    processed_feature = lxml.html.fromstring(features[sentence]).text_content()
    # stemming
    processed_feature = stemmer.stem(processed_feature)
    # converting to lowercase
    processed_feature = processed_feature.lower()
    # add to the list
    processed_features.append(processed_feature)

pd.set_option('display.max_colwidth', 0)
pd.DataFrame(list(zip(features, processed_features)), columns=['Sebelum di casefolding', 'Setelah casefolding'])
```

Gbr. 7 kode proses *case folding*

	Sebelum di casefolding	Setelah casefolding
0	YANG DI AJARKAN MUDAH DIPAHAMI	yang di ajar mudah paham
1	WALUPUN JAUH JARAK SANGAT KOMPETEN SEKALI	walupun jauh jarak sangat kompeten sekali
2	WAKTUNYA YANG TIDAK KONSISTEN DENGAN JADWAL PERKULIAHAN	waktu yang tidak konsisten dengan jadwal kuliah
3	WAKTU MENGAJAR SERING KELEWATAN TAPI TIDAK MASALAH SIH KARENA KELAS JUGA JADI MENYENANGKAN	waktu ajar sering lewat tapi tidak masalah sih karena kelas juga jadi senang
4	VOKAL PERLU DIKERASKAN	vokal perlu keras
5	UNTUK SELALU DI TINGKATKAN DAN DI PERTAHANKAN.	untuk selalu di tingkat dan di tahan
6	UNTUK PENJELASAN PERHITUNGAN SEBAIKNYA LEBIH DETIL	untuk jelas hitung baik lebih detil
7	UNTUK MENDATANGKAN DOSEN TAMU KEDEPANNYA TETAP DIPERTAHANKAN	untuk datang dosen tamu depan tetap tahan
8	UNTUK MASALAH UJIN AGAR TIDAK TERLALU SUSAH KARENA KELAS SAYA KELAS KARYAWAN	untuk masalah ujin agar tidak terlalu susah karena kelas saya kelas karyawan
9	UNTUK DOSEN LEBIH BERSEMANGAT MENGAJARNYA DAN LEBIH BANYAK MENDAPAT ILMU JIKA DENGAN PEMETAAN PIKIRAN DARI PADA PAKAI SLIDE PPT UNTUK UBL MOHON JANGAN DIPAKSAKAN DOSEN UNTUK MENGGUNAKAN POWERPOINT	untuk dosen lebih semangat ajar dan lebih banyak dapat ilmu jika dengan meta pikir dari pada pakai slide ppt untuk ubl mohon jangan paksa dosen untuk guna powerpoint
10	TUGASNYA BOLEH UNTUK LEBIH DI PERJELAS	tugas boleh untuk lebih di jelas
11	TUGAS YANG DIBERIKAN SESUAI DENGAN KAMAMPUAN MAHASISWA	tugas yang beri sesuai dengan kamampuan mahasiswa
12	TUGAS YANG DIBERIKAN SANGAT TEPAT UNTUK MAHASISWA	tugas yang beri sangat tepat untuk mahasiswa
13	TUGAS TUGAS MATA KULIAHNYA CUKUP MEMBANTU PENAMBAHAN PENGETAHUAN	tugas tugas mata kulah cukup bantu tambah tahu
14	TUGAS TUGAS MATA KULIAH KALO BISA DITAMBAH	tugas tugas mata kuliah kalo bisa tambah
15	TUGAS TERLALU BERAT DAN PADAT	tugas terlalu berat dan padat
16	TRIMAKASIH BAPAK ATAS ILMU YG TELAH BAPAK BERIKAN SMOGA BAPAK SEHAT DAN SUKSES SLALU BAIK DUNIA DAN AKHIRAT	trimakasih bapak atas ilmu yg telah bapak ikan smoga bapak sehat dan sukses slalu baik dunia dan akhirat
17	TOLONG UNTUK TUGAS JANGAN TERLALU SULIT	tolong untuk tugas jangan terlalu sulit

Gbr. 8 hasil proses *case folding*

Pada proses penghilangan *stopword*, dibutuhkan modul *Sastrawi.StopWordRemover*. Modul tersebut menyediakan beberapa *corpora teks*, salah satunya adalah *StopRemoverFactory*. Selain kata-kata umum, ada juga kelompok kata *stopword* yang memiliki posisi penting dalam

morfologi dan tidak bisa berdiri sendiri. Proses yang dimaksud seperti terlihat pada Gbr.9 dan Gbr. 10.

```
import lxml.html
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
# create stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()
# create and add new stop words, first get the original data
stop_factory = StopWordRemoverFactory()
more_stopword = ['dengan', 'ia', 'bahwa', 'oleh']
# merge stopword
stop_factory.get_stop_words() + more_stopword
stop_factory = stop_factory.create_stop_word_remover()
processed_features = []
for sentence in range(0, len(features)):
    # remove html tags
    processed_feature = lxml.html.fromstring(features[sentence]).text_content()
    # stop words removal
    processed_feature = stop_factory.remove(processed_feature)
    # add to the list
    processed_features.append(processed_feature)
print('Sebelum di stopword removal: ', features[21])
print('Setelah stopword removal: ', processed_features[21])
```

```
Sebelum di stopword removal:  lebih di tingkatkan dalam waktu
Setelah stopword removal:  tingkatkan waktu
```

Gbr. 9 kode proses *stopword*

	Sebelum di stopword removal	Setelah stopword removal
0	YANG DI AJARKAN MUDAH DIPAHAMI	ajar mudah paham
1	WALUPUN JAUH JARAK SANGAT KOMPETEN SEKALI	walupun jarak kompeten
2	WAKTUNYA YANG TIDAK KONSISTEN DENGAN JADWAL PERKULIAHAN	waktu konsisten jadwal kuliah
3	WAKTU MENGAJAR SERING KELEWATAN TAPI TIDAK MASALAH SIH KARENA KELAS JUGA JADI MENYENANGKAN	waktu ajar sih kelas senang
4	VOKAL PERLU DIKERASKAN	vokal keras
5	UNTUK SELALU DI TINGKATKAN DAN DI PERTAHANKAN.	tingkat tahan
6	UNTUK PENJELASAN PERHITUNGAN SEBAIKNYA LEBIH DETIL	hitung detil
7	UNTUK MENDATANGKAN DOSEN TAMU KEDEPANNYA TETAP DIPERTAHANKAN	dosen tamu tahan
8	UNTUK MASALAH UJIN AGAR TIDAK TERLALU SUSAH KARENA KELAS SAYA KELAS KARYAWAN	ijin susah kelas kelas karyawan
9	UNTUK DOSEN LEBIH BERSEMANGAT MENGAJARNYA DAN LEBIH BANYAK MENDAPAT ILMU JIKA DENGAN PEMETAAN PIKIRAN DARI PADA PAKAI SLIDE PPT UNTUK UBL MOHON JANGAN DIPAKSAKAN DOSEN UNTUK MENGGUNAKAN POWERPOINT	dosen semangat ajar ilmu meta pikir pakai slide ppt ubl paksa dosen powerpoint
10	TUGASNYA BOLEH UNTUK LEBIH DI PERJELAS	tugas
11	TUGAS YANG DIBERIKAN SESUAI DENGAN KAMAMPUAN MAHASISWA	tugas sesuai kamampuan mahasiswa
12	TUGAS YANG DIBERIKAN SANGAT TEPAT UNTUK MAHASISWA	tugas mahasiswa
13	TUGAS TUGAS MATA KULIAHNYA CUKUP MEMBANTU PENAMBAHAN PENGETAHUAN	tugas tugas kuliah bantu
14	TUGAS TUGAS MATA KULIAH KALO BISA DITAMBAH	tugas tugas kuliah kalo
15	TUGAS TERLALU BERAT DAN PADAT	tugas berat padat
16	TRIMAKASIH BAPAK ATAS ILMU YG TELAH BAPAK BERIKAN SMOGA BAPAK SEHAT DAN SUKSES SLALU BAIK DUNIA DAN AKHIRAT	trimakasih ilmu yg ikan smoga sehat sukses slalu dunia akhirat
17	TOLONG UNTUK TUGAS JANGAN TERLALU SULIT	tolong tugas sulit

Gbr. 10 hasil proses *stopword*

Pada proses tokenisasi adalah proses dimana mengubah sebuah kalimat menjadi *unigram* kata. Dalam proses ini dibutuhkan modul *tokenize.word_tokenize* dari *nltk*. Meski Python memiliki kemampuan untuk melakukan tugas-tugas *Natural Language Processing* (NLP) dasar, namun tidak cukup powerful untuk melakukan tugas-tugas standar NLP, maka dari itu muncullah modul *tokenize*. Proses yang dimaksud seperti terlihat pada Gbr.11.

```
from nltk.tokenize import word_tokenize
text1 = processed_features[0]
tokens = word_tokenize(text1)
print('Kalimat: ', processed_features[0])
print('Tokenisasi Kalimat: ', tokens)
```

```
Kalimat:  ajar mudah paham
Tokenisasi Kalimat:  ['ajar', 'mudah', 'paham']
```

Gbr. 11 kode proses tokenisasi

Pada proses pembobotan *Term Frequency-Inverse Document Frequency* (TF-IDF) digunakan modul pustaka *scikit-learn* untuk mengekstraksi teks dengan menggunakan *TfidfVectorizer*. Hasil dari pembobotan TF-IDF berupa matriks. Kode untuk proses pembobotan TF-IDF adalah seperti terlihat pada Gbr. 12 yang menunjukkan hasil TF-IDF yang telah dilakukan. Matriks TF- IDF yang dihasilkan berukuran 1323x624.

```
from nltk.tokenize import word_tokenize
text1 = processed_features[0]
tokens = word_tokenize(text1)
print('Kalimat: ', processed_features[0])
print('Tokenisasi Kalimat: ', tokens)
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(tokens).toarray()
print('Pembobotan TF-IDF: \n', X)
```

```
Kalimat:  ajar mudah paham
Tokenisasi Kalimat:  ['ajar', 'mudah', 'paham']
Pembobotan TF-IDF:
 [[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
```

Gbr. 12 kode dan hasil pembobotan TF-IDF

Pada proses implementasi, selanjutnya akan menggunakan hasil *preprocessing*, tokenisasi dan ekstrasi fitur (*feature extraction*) ke dalam proses klasifikasi menggunakan metode *Naïve Bayes*, *K-Nearest Neighbor* dan *Support Vector Machine*.

```
# Bagi dataset komentar mahasiswa hasil preprocessing, dan feature extraction menjadi training set dan test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(komentar_mahasiswa)

# Load supervised model classifier: Naive Bayes, K-Nearest Neighbor, dan Vector Machine
from sklearn import naive_bayes.MultinomialNB
from sklearn import neighbors.KNeighborsClassifier
from sklearn import svm.LinearSVC
```

Gbr. 13 kode pemanggilan classifier

Implementasi proses klasifikasi komentar mahasiswa dengan *Naïve Bayes*, *K-Nearest Neighbor dan Support Vector Machine* ialah dengan cara melakukan proses pembelajaran data latih (*training set*) terhadap fungsi algoritma masing-masing *classifier* dengan menggunakan modul *fit* seperti yang dapat dilihat pada potongan *script* pada Gbr. 14 di bawah ini.

```
# Pembelajaran Classifier menggunakan training set
Naive_Bayes_Text_Classifier = naive_bayes.MultinomialN().fit(X_train, y_train)
KNN_Text_Classifier = neighbors.KNeighborsClassifier().fit(X_train, y_train)
SVM_Text_Classifier = svm.LinearSVC().fit(X_train, y_train)
```

Gbr. 14 kode implementasi pembelajaran *classifier*

Implementasi proses klasifikasi komentar mahasiswa selanjutnya adalahmelakukan proses prediksi (*prediction*) terhadap data tes (*test set*) dengan fungsi algoritma masing-masing *classifier*yaitu menggunakan modul *predict* seperti yang dapat dilihat pada potongan *script* pada Gbr. 15 di bawah ini.

```
# Prediksi classifier menggunakan test set
Naive_Bayes_Text_Classifier.predict(X_test)
KNN_Text_Classifier.predict(X_test)
SVM_Text_Classifier.predict(X_test)
```

Gbr. 15 kode implementasi prediksi *classifier*

Untuk mengukur akurasi dari implementasi proses klasifikasi komentar masing-masing *classifier* dapat dilakukan

dengan menggunakan modul accuracy_*score* yang dapat dilihat pada potongan *script* pada Gbr. 16 di bawah ini.

```
# 10 fold cross validation to all supervised model
from sklearn.model_selection import cross_val_score

# Naive Bayes confusion matrix and 10 fold cross validation
print(confusion_matrix(y_test, Naive_Bayes_Text_Classifier.predict(X_test)))
print(classification_report(y_test, Naive_Bayes_Text_Classifier.predict(X_test)))
print('Akurasi Naive Bayes Classifier: ', accuracy_score(y_test, Naive_Bayes_Text_Classifier.predict(X_test))
scores_naive_bayes = cross_val_score(Naive_Bayes_Text_Classifier(), X_test, y_test, cv=10)

# KNN confusion matrix and 10 fold cross validation
print(confusion_matrix(y_test, KNN_Text_Classifier.predict(X_test))
print(classification_report(y_test, KNN_Text_Classifier.predict(X_test))
print('Akurasi KNN Classifier: ', accuracy_score(y_test, KNN_Text_Classifier.predict(X_test)))
scores_knn = cross_val_score(KNN_Text_Classifier(), X_test, y_test, cv=10)

# SVM confusion matrix and 10 fold cross validation
print(confusion_matrix(y_test, SVM_Text_Classifier.predict(X_test))
print(classification_report(y_test, SVM_Text_Classifier.predict(X_test))
print('Akurasi SVM Classifier: ', accuracy_score(y_test, SVM_Text_Classifier.predict(X_test))
scores_svm = cross_val_score(SVM_Text_Classifier(), X_test, y_test, cv=10)
```

Gbr. 16 kode implementasi akurasi *classifier*

B. *Pengujian dan Hasil Pengujian*

Untuk melakukan prediksi polaritas komentar mahasiswa, digunakan data yang telah melalui tahap *preprocessing* dan *feature extraction*, kemudian dibagi menjadi dua bagian, satu bagian untuk pelatihan (*train set*) dan satu bagian lainnya digunakan untuk pengujian (*test set*). Data yang disiapkan dilakukan *preprocessing* dengan tujuan untuk menghasilkan fitur prediksi berupa matriks frekuensi komentar dengan akurasi terbaik yang akan digunakan untuk proses perhitungan pelatihan untuk prediksi. Untuk mendapatkan matriks frekuensi relatif komentar dengan akurasi terbaik, sistem melakukan pemilihan fitur (*feature selection*) dengan menghasilkan data *unigram* kata dengan pembobotan *Term Frequency-Inverse Document Frequency* (TF-IDF). Pada pengujian *confusion matrix* dilakukan uji coba dengan merubah perbandingan antara *test set* dan *train set.*

```
[[ 50   1   3]
 [  2  12  28]
 [  8   2 225]]
              precision    recall  f1-score   support

         0.0       0.83      0.93      0.88        54
         1.0       0.80      0.29      0.42        42
         2.0       0.88      0.96      0.92       235

    accuracy                           0.87       331
   macro avg       0.84      0.72      0.74       331
weighted avg       0.86      0.87      0.85       331

Akurasi Naive Bayes Classifier:  0.8670694864048338
```

Gbr. 17 pengujian dengan *Naïve Bayes*

Pengujian *confusion matrix* dengan metode *Naïve Bayes* menghasilkan akurasi 86.7%, dengan *precision* tertinggi 88% untuk polaritas Positif, dan *recall*terendah 29% untuk polaritas Negatif.

```
[[ 44   6   0]
 [  3  44   2]
 [ 16  23 193]]
              precision    recall  f1-score   support

         0.0       0.70      0.88      0.78        50
         1.0       0.60      0.90      0.72        49
         2.0       0.99      0.83      0.90       232

    accuracy                           0.85       331
   macro avg       0.76      0.87      0.80       331
weighted avg       0.89      0.85      0.86       331

Akurasi KNN Classifier:  0.8489425981873112
```

Gbr. 18 pengujian dengan K-NN

Pengujian *confusion matrix* dengan metode *K-Neares Neighbor* menghasilkan akurasi 84.8% dengan *precision* tertinggi unutk polaritas Positif 99%, dan *recall* terendah 83% untuk polaritas Positif.

```
[[ 49   1   4]
 [  0  38  11]
 [  5   5 218]]
              precision    recall  f1-score   support

         0.0       0.91      0.91      0.91        54
         1.0       0.86      0.78      0.82        49
         2.0       0.94      0.96      0.95       228

    accuracy                           0.92       331
   macro avg       0.90      0.88      0.89       331
weighted avg       0.92      0.92      0.92       331

Akurasi SVM Classifier:  0.9214501510574018
```

Gbr. 19 pengujian dengan SVM

Pengujian *confusion matrix* dengan metode *Support Vector Machine* menghasilkan akurasi 92.14%, dengan *precision* tertinggi 95% untuk polaritas Positif, dan *recall* terendah 78% untuk polaritas Negatif.

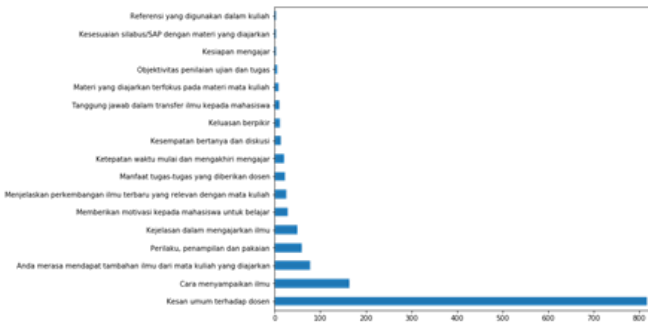
Perbandingan *akurasi* menggunakan 10 *fold cross validation*untuk mengevaluasi kinerja *classifier* dapat dilihat pada table di bawah ini:

TABEL I. PERBANDINGAN HASIL UJI

Test:Train (%) 10 Fold Cross Validation	Naïve Bayes Acuracy %	KNN Acuracy %	SVM Acuracy %
Cross Validation-1	86.56	73.88	94.77
Cross Validation-2	90.29	80.59	91.79
Cross Validation-3	84.21	78.94	85.71
Cross Validation-4	86.46	88.72	90.22
Cross Validation-5	90.22	81.95	88.72
Cross Validation-6	86.36	81.81	84.09
Cross Validation-7	80.15	75.57	87.02
Cross Validation-8	85.49	83.96	88.54
Cross Validation-9	93.89	83.96	95.41
Cross Validation-10	87.78	83.96	88.54

Dari hasi perbandingan uji coba pada Tabel I dapat disimpulkan bahwa perbandingan *cross validation* data test dan data train yang menghasilkan nilai terbaik adalah metode *Support Vector Machine* dengan tingkat akurasi 95.41%.

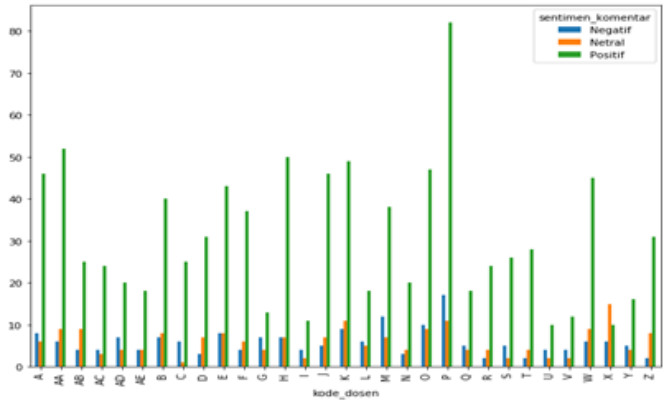
Dengan kemampuan sistem untuk melakukan klasifikasi polaritas komentar mahasiswa, dimungkinkan analisis komentar terhadap indikator yang ada di angket, seperti Gbr. 20.



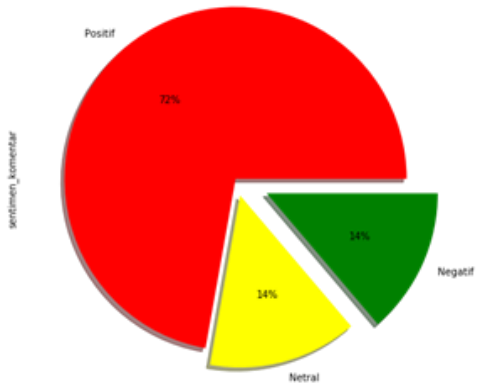
Gbr. 20 distribusi komentar terhadap indicator

Berdasarkan distribusi komentar mahasiswa terhadap indikator penilaian, komentar mahasiswa banyak di indikator penilaian kesan umum terhadap dosen.

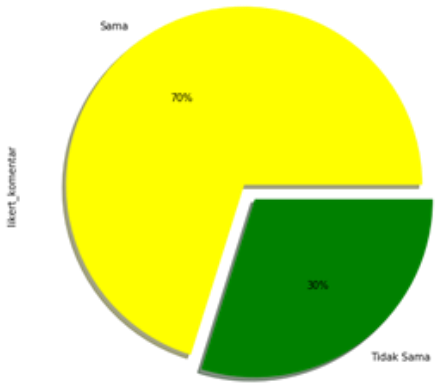
Dengan kemampuan sistem untuk melakukan klasifikasi polaritas komentar mahasiswa, dimungkinkan analisis komentar terhadap indikator yang ada di angket, seperti Gbr. 21, Gbr. 22, dan Gbr. 23:



Gbr. 21 polaritas sentiment komentar terhadap dosen



Gbr. 22 persen polaritas sentiment komentar terhadap dosen



Gbr.23 polaritas sentiment komentar vs skor numerik

Berdasarkan perbandingan di atas, dapat dilihat perbandingan kesesuaian antara skor numerik (*likert score)* dengan polaritas sentimen komentar, yaitu dimungkinkan mendapat skor numerik yang berbeda dengan polaritas sentimen komentar. Mendapat skor numerik 4-5 (Baik – Sangat Baik) namun polaritas sentimen komentar negative.

V. KESIMPULAN

Beberapa kesimpulan yang dapat diperoleh dari penelitian yang telah dilakukan adalah sebagai berikut:

1) Dengan *Term Frequency-Inverse Document Frequency* (TF-IDF) *feature extraction*, serta 10 *fold cross validation*, akurasi tertinggi di hasilkan dari metode *Support Vector Machine* yaitu 95.41%.
2) Akurasi yang dihasilkan sistem pada proses klasifikasi polaritas komentar mahasiswa menggunaan *Term Frequency-Inverse Document Frequency* (TF-IDF) sebagai *feature extraction* yang diterapkan pada 3 *classifier* yang berbeda dari *supervised model* rata-rata memiliki akurasi di atas 80%, Mendekati hasil penelitian serupa yaitu *Word Base feature extraction* dengan metode *K-Nearest Neighbor* sebesar 84.32%.

REFERENSI

[1] Kusrini. *Sistem Pakar, Teori dan Aplikasi*. Yogyakarta: Andi Offset. 2006.
[2] Sarkar, Dipanjan. *Text Analytics with Python*. New York: Apress. 2016.
[3] K. Ganesan, "Text Mining, Analytics & More: All About Stop Words for Text Mining and Information Retrieval," [Online]. Available: http://textanalytics101.rxnlp.com/2014/10/all-about-stop-words-for-text-mining.html.
[4] X. Huang, dan Q. Wu, "Micro-blog commercial word extraction based on improved TF-IDF algorithm," IEEE International Conference of IEEE Region 10 (TENCON 2013). 2013.
[5] E. Rasywir, dan A. Purwarianti, "Eksperimen pada Sistem Klasifikasi Berita Hoax Berbahasa Indonesia Berbasis Pembelajaran Mesin," Jurnal Cybermatika 3 (2). 2015.
[6] J. Han, M. Kamber, dan J. Pei, *Data Mining: Concept and Techniques*, Thrid Edition. Waltham: Morgan Kaufmann Publishers. 2011.
[7] M. F. Nur, dan D. D. Santika, "Analisis Sentimen pada Dokumen Berbahasa Indonesia dengan Pendekatan Support Vector Machine," Konferensi Nasional Sistem dan Informasi, KNS&I11-002, Nov. 2011

Sigit Wijanarko lahir di Sintang pada tanggal 23 Juni 1980. Lulus S1 Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Pembangunan Nasional – Jakarta pada tahun 2004. Lulus Magister Ilmu Komputer Program Pascasarjana Universitas Budi Luhur dengan konsentrasi Rekayasa Komputasi Terapan pada tahun 2019. Saat ini aktif sebagai Dosen tetap di STMIK Antar Bangsa dan praktisi IT di perusahaan swasta.