Aspect Based Sentiment Analysis pada Layanan Umpan Balik Universitas dengan Menggunakan Metode Naïve Bayes dan Latent Semantic Analysis

Gunawan Setiawan, Henry Novianus Palit, Endang Setyati
Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto 121 – 131 Surabaya 60236
Telp. (031) – 2983455, Fax. (031) – 8417658
E-Mail: css.gunawansetiawan@gmail.com, hnpalit@petra.ac.id, endang@stts.edu

ABSTRAK

Universitas Kristen Petra menerapkan sistem pengisian kuesioner *online* yang bertujuan untuk mendapatkan umpan balik terhadap layanan yang telah ada. Di mana untuk saat ini untuk memproses hasilnya masih dilakukan secara *manual*. Namun hal ini tidak efektif karena akan memerlukan waktu dan tenaga, sehingga dikembangkanlah sebuah program yang membantu meringankan pekerjaan manusia dengan menggunakan Python dan menerima input berupa *file excel*. Metode yang digunakan adalah *Naive Bayes* dan *Latent Semantic Analysis* untuk membantu pengklasifikasian sentimen agar dapat ditentukan topik, *sentiment score*, dan tindakannya.

Kata Kunci: *ABSA, SA, aspect based sentiment analysis, sentiment analysis, NLP, sentiment.*

ABSTRACT

Petra Christian University implements a system of filling out online questionnaires that aims to get feedback on existing services. Where for now to process the results is still done manually. But this is not effective because it will require time and effort, so a program was developed that helped ease human work by using Python and receive input in the form of excel files. The method used is Naive Bayes and Latent Semantic Analysis to help classify sentiments so that topics, sentiment scores, and actions can be determined.

Keywords: *ABSA, SA, aspect based sentiment analysis, sentiment analysis, NLP, sentiment.*

1. PENDAHULUAN

Aspect based Sentiment Analysis (ABSA) merupakan subarea dari *opinion* mining yang memungkinkan untuk mendapatkan informasi tentang aspek yang dimaksud oleh pengguna melalui *mining review*[1] . Sebagai contoh, dari "wifi di gedung p lambat" berusaha untuk memberikan pendapat dengan target topik (aspek) "Internet". Sedangkan Analisis Sentimen merupakan salah satu tipe dari *Natural Language Processing* (NLP) untuk mengetahui *mood* publik terhadap suatu produk atau topik tertentu [7]. ABSA dapat bermanfaat untuk mengetahui topik (aspek) yang dimaksud seseorang dalam pendapatnya. Selain itu dapat diketahui pula aspek tersebut bersifat positif, netral, atau negatif.

Universitas Kristen Petra menerapkan sistem pengisian kuesioner *online* yang bertujuan untuk mendapatkan umpan balik terhadap layanan yang telah ada. Di mana untuk saat ini untuk memproses hasilnya masih dilakukan secara *manual*. Namun hal ini tidak efektif karena akan memerlukan waktu dan tenaga untuk membaca satu per satu umpan balik dari publik baru kemudian menggolongkan sentimennya. Dengan demikian dirasa perlu untuk membuat aplikasi yang dapat memproses umpan balik dari publik sehingga didapatkan hasil sentimennya.

2. DASAR TEORI

2.1 Sastrawi *Stemmer*

Stemming adalah suatu proses yang digunakan untuk mengambil kata dasar dari sebuah kata dengan cara menghapus awalan dan akhiran pada kata tersebut. Perlu dilakukan *stemming* agar kata yang memiliki bentuk berbeda namun maknanya sama tetap dianggap 1 buah kata. Contoh:

Kata 1 : Tolong parkiran ANTA diperbesar

Kata 2 : Tolong parkiran ANTA dibesarkan

Sastrawi Python merupakan library python yang dapat digunakan untuk mendapatkan kata dasar dari kata yang kita inputkan. Algoritma yang digunakan oleh *library* ini adalah algoritma Nazief dan Andriani, di mana algoritma ini merupakan salah satu algoritma yang cukup populer untuk melakukan *stemming* kata dalam Bahasa Indonesia. Sastrawi Python sendiri bergantung pada kamus kata dasar dari www.kateglo.com [6]

2.2 NLTK

NLTK merupakan sebuah *platform* untuk membuat program Python yang berurusan dengan bahasa manusia . Menyediakan lebih dari 50 *corpora* dan *lexical resource* seperti WordNet, beserta dengan *library* pemrosesan teks yang sesuai untuk klasifikasi, tokenisasi, *stemming*, *tagging*, *parsing*, dan *semantic reasoning*. Hal ini membuat NLTK sangat cocok dan banyak digunakan sebagai *tools* untuk membuat program NLP.

2.3 POS Tagger

Awalnya POS Tagger akan melakukan tokenisasi terhadap teks menggunakan kamus Bahasa Indonesia. Kemudian akan diproses menggunakan aturan yang sudah didefinisikan untuk menentukan kelas kata yang tepat. Jika tidak ada kelas yang sesuai, kata tersebut akan dibiarkan menjadi ambigu. *Library* ini memiliki variasi *tagset* sebanyak 23 jenis yang dapat dilihat pada Tabel 1.

2.4 *Naïve Bayes*

Sesuai dengan namanya, *Naive Bayes* mendasarkan perhitungannya berdasarkan Teorema *Bayes* dengan asumsi independen (*naive*) yang tinggi. Jadi tidak mempedulikan keterkaitan antara masing-masing kata. Walaupun demikian, hingga sekarang *Naive Bayes* masih sering dijumpai dalam pengklasifikasian teks seperti *spam*, topik, dan bahkan analisis sentimen. Persamaan 1 adalah cara perhitungan *Naive Bayes*[3]

$$P(S|W) = \frac{P(W|S)P(S)}{P(W)} \quad (1)$$

Namun apabila ada 1 kata w yang tidak ada dalam *training*, maka $P(w|S)$ akan menjadi 0. Untuk itu, perlu dilakukan koreksi dengan cara *Laplacian Smoothing* agar terhindar hal tersebut. Pada Persamaan 2 ditunjukkan bagaimana cara untuk melakukan *Laplacian Smoothing*[5]

$$P(w_i|Sj) = \frac{count\ (w_i Sj)+1}{count\ (Sj)+V+1} \quad (2)$$

Proses penentuannya sendiri dengan menghitung peluang yang paling besar yang dapat dituliskan dalam Persamaan 3 sebagai berikut:

$$s = \underset{S_j \in S}{\text{Arg max}}\, P(w_1, w_2, .., w_n|S_j)P(S_j) \quad (3)$$

Kekurangan dari *Naive Bayes* adalah asumsi yang terlalu *naive*, walaupun demikian *Naive Bayes* terbukti dapat menyelesaikan masalah-masalah yang kompleks.

Tabel 1. POS Tag

No	Tag	Deskripsi
1	CC	*Coordinating conjuction*
2	CD	*Cardinal number*
3	OD	*Ordinal number*
4	DT	*Determiner*

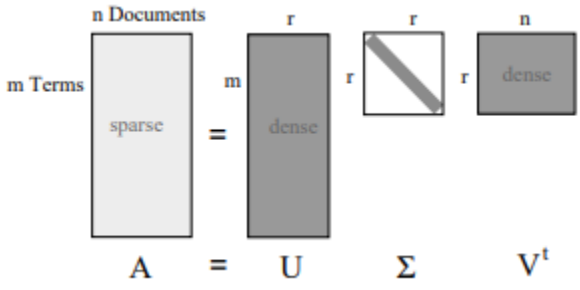
2.5 *Latent Semantic Analysis*

2.5.1 Singular Value Decomposition

Pada sub bab ini akan dijelaskan mengenai teori dari *Latent Semantic Analysis*, bagaimana melakukan pemrosesan menggunakan *Singular Value Decomposition*, dan bagaimana mencari kemiripan dengan menggunakan *Cosine Similarity*.

Singular Value Decomposition (SVD) merupakan salah satu teknik reduksi dimensi yang bermanfaat untuk memperkecil nilai kompleksitas dalam pemrosesan *term-document* matriks. Dengan menggunakan SVD sebuah matriks akan diurai menjadi 3 matriks pembentuknya seperti pada Gambar 1 , yaitu [8]:

1. Matriks Ortogonal U
2. Matriks diagonal S
3. *Transpose* dari matriks ortogonal V



Gambar 2. Matriks SVD

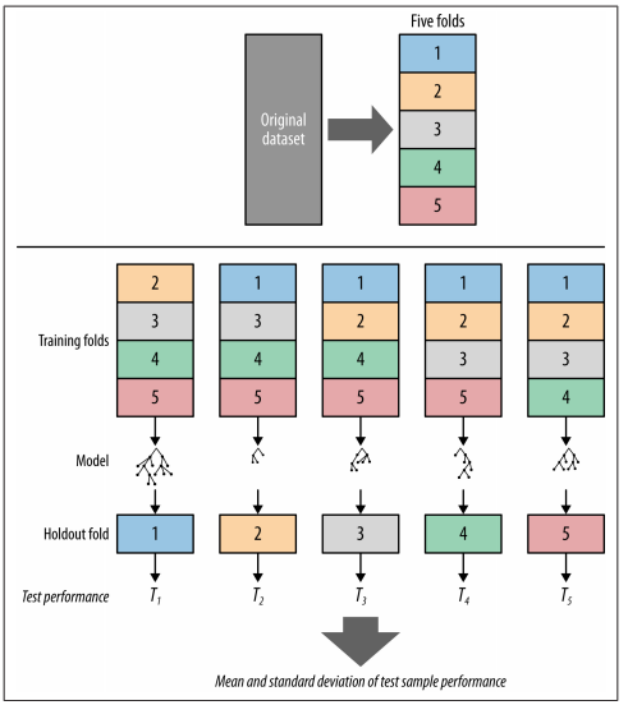
2.5.2 Cosine Similarity

Menggunakan teori vektor seperti pada umumnya, di mana untuk menentukan jarak sudut dari 2 buah vektor dapat dihitung dari hasil *dot product* kedua vektor. Pada Persamaan 4 menunjukkan bagaimana semakin besar nilai dari dot product menandakan 2 vektor/kalimat semakin mirip.[2]

$$\cos\theta = \frac{\vec{A}\cdot\vec{B}}{|\vec{A}||B|} \quad (4)$$

2.6 Cross Validation

Merupakan metode yang memecah data utuh menjadi k bagian data, dimana k merupakan nilai yang ditentukan oleh pengguna. bagian dari data ini akan dimodelkan sedangkan sisanya akan dijadikan data untuk melakukan testing.



Gambar 3. Contoh Cross Validation Error! Reference source not found.

Pada Gambar 3, data dibagi menjadi 5 bagian. Untuk setiap bagian akan dijadikan sebagai *data testing*. Kemudian setiap bagian dari data tersebut akan menghasilkan nilai performa / akurasi [4]

3. DESAIN SISTEM

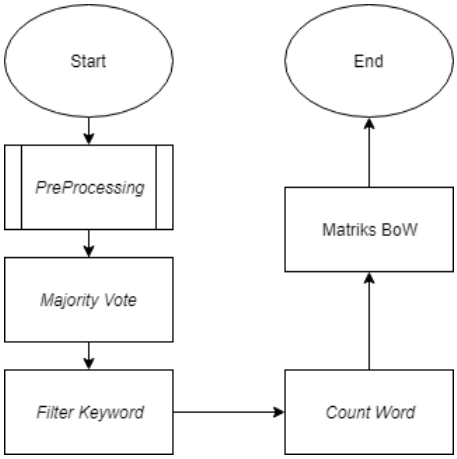
3.1 Proses Membuat Matriks BoW

Setelah data telah dilakukan *labeling* oleh beberapa *user*. Akan ditentukan *majority vote*-nya untuk masing-masing topik serta masing-masing sentimen. Kemudian diambil beberapa kata yang telah ter-*filter* berdasarkan *keyword* pada proses pelabelan data sebagai *bag of words*. Bag of words sendiri merupakan sekumpulan kata yang telah terdeteksi dengan masing-masing frekuensinya untuk tiap kelas yang disediakan. Dalam hal ini untuk topik, jumlah kelasnya ada 5 macam yaitu: internet, toilet, kantin, parkir, dan lain-lain. Pada tahap ini hasil akan disimpan ke dalam file dengan format "model_nb.txt. Gambar 4 menunjukkan proses pembuatan matriks BoW

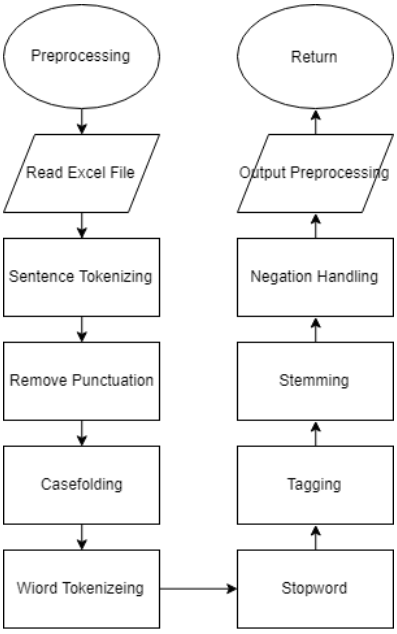
3.2 Proses *Preprocessing*

Gambar 5 menunjukkan proses untuk *preprocessing* data kuesioner yang akan dicari sentimennya. Pertama *user* akan diminta memilih sebuah *file* excel yang berisikan komentar kuesioner. Langkah berikutnya adalah dilakukan *sentence tokenizer*, di mana komentar

akan dipisah-pisahkan berdasarkan tanda ".". Kemudian dilakukan pembuangan tanda baca. Proses selanjutnya adalah mengubah semua kalimat menjadi huruf kecil. Setelah itu dilakukan *word tokenizer*, yaitu memisah-misahkan kata berdasarkan spasi.



Gambar 4 Membuat matriks BoW

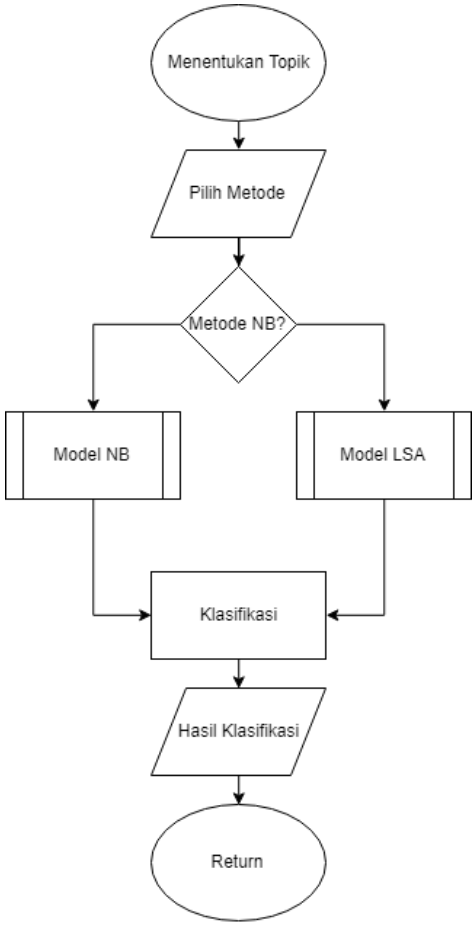


Gambar 5. Proses *Preprocessing*

3.3 Proses Klasifikasi

Ada 2 pilihan metode untuk menentukan topik, yaitu *Naive Bayes* dan *Latent Semantic Analysis*. Setelah *user* menentukan metode, maka data yang telah di-*preprocessing* sebelumnya akan diproses sesuai dengan model metodenya masing-masing sesuai dengan Gambar 6.

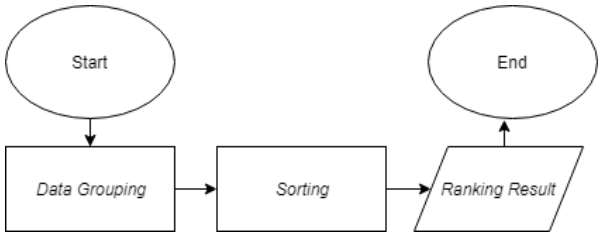
Jika menggunakan metode *Naive Bayes* maka klasifikasinya akan ditambahkan *Laplacian Smoothing*. Sedangkan jika menggunakan model *Latent Semantic Analysis*, akan dilakukan perhitungan *Cosine Similarity*. Nilai *Cosine Similarity* yang paling besar akan dianggap sebagai topiknya. Setelah itu klasifikasi ditampilkan di halaman *web*. Nantinya *user* akan bisa menentukan apakah topik telah sesuai atau tidak. Jika kurang sesuai *user* bisa mengubahnya secara *manual*. Pada tahapan ini data akan diupdate ke dalam *database*



Gambar 6. Proses Klasifikasi

3.4 Proses Ranking

Gambar 7 menunjukkan bagaimana alur kerja untuk me-*ranking* sentimen. Setelah sentimen telah ditentukan topik dan *sentiment score*-nya maka sentimen telah dapat dikelompokkan. Pengelompokkan didasarkan pada topik dan tindakan yang sama. Setelah itu dilakukan pengurutan. Pada tahap ini data *ranking* akan dimasukkan ke ke dalam format excel.



Gambar 7. Proses Ranking

4. IMPLEMENTASI SISTEM

Implementasi pengkodean sistem, menggunakan Bahasa pemrograman Python dengan versi 3.0. *Framework* yang digunakan untuk sistem ini adalah Flask *Micro-Framework.* Adapun beberapa *library* yang mendukung sistem ini adalah: nltk, Matplotlib, Scikit-learn, Flask, Sastrawi, Scipy.

Karena program dijalankan lewat web maka ada beberapa akses *HTTP Method*, antara lain: GET (Menerima data dalam bentuk *array* atau *object*) dan POST (Memasukan data, INSERT).

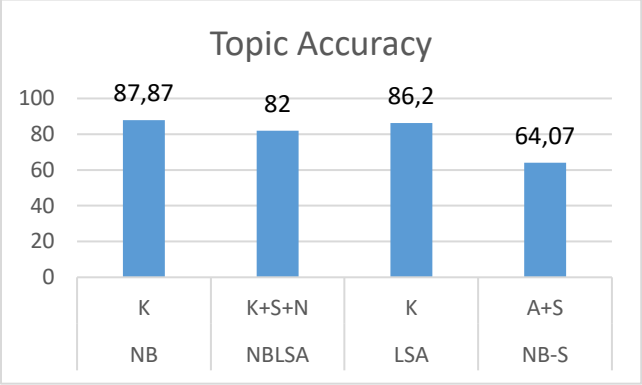
5. ANALISA DAN PENGUJIAN

5.1 Spesifikasi Pengujian

Pengujian dilakukan dengan menggunakan laptop dengan spesifikasi sebagai berikut: RAM: 12 GB, DDR3, HDD: 1 TB, CPU: Core i7, OS Windows 10.

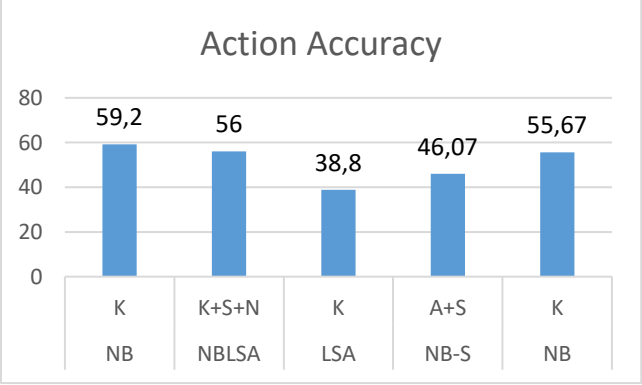
5.2 Pengujian Akurasi

Pengujian akurasi dilakukan dengan *n-fold cross validation*. Nilai n yang digunakan bervariasi yaitu 3, 4, 5, 6, dan 10. Kemudian diambil nilai yang terbaik. Kombinasi klasifikasi topik, sentimen, dan juga tindakan dengan menggunakan *keyword* saja menghasilkan akurasi yang paling baik. Yaitu 87.87 % untuk penentuan topik, 59.2 % untuk penentuan tindakan, 55.67 % untuk penentuan sentimen, dan untuk akurasi program sebesar 29.87 %. Untuk hasil akurasi dapat dilihat pada Gambar 8. Terlihat bahwa metode *Naive Bayes* dan menggunakan fitur *keywords* saja merupakan kombinasi yang terbaik dalam hal akurasi untuk melakukan analisis topik

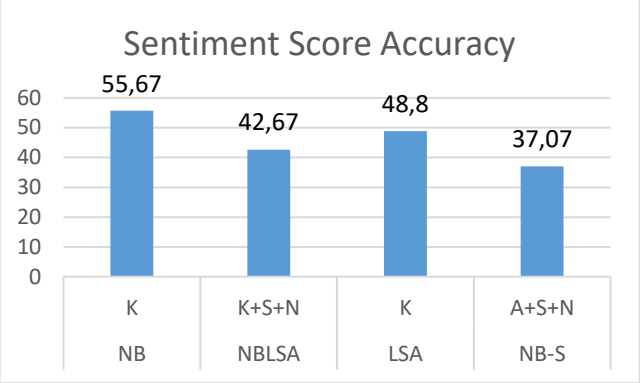


Gambar 8. Akurasi Topik

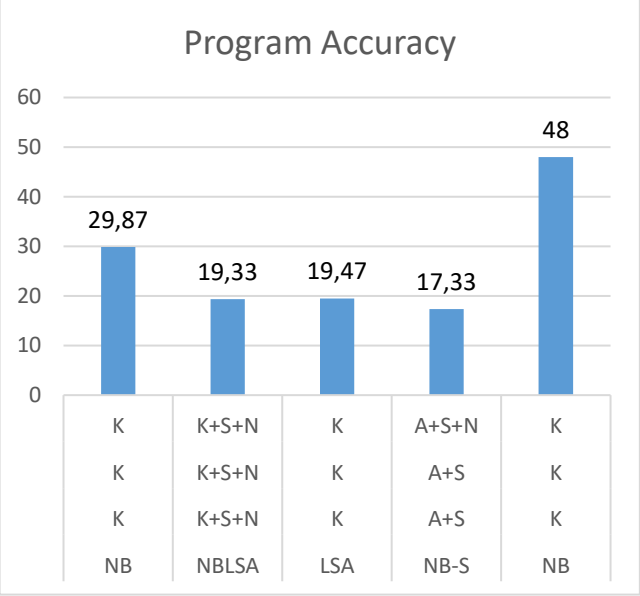
Pada Gambar 9. Terlihat bahwa metode *Naive Bayes* dan menggunakan fitur *keywords* saja merupakan kombinasi yang terbaik dalam hal akurasi untuk melakukan analisis action



Gambar 9. Akurasi Tindakan



Gambar 10. Akurasi Sentimen



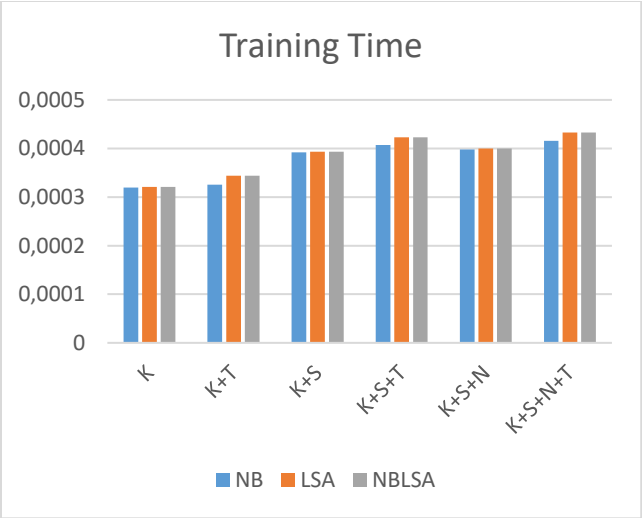
Gambar 11. Akurasi Program

5.3 Pengujian Waktu *Training*

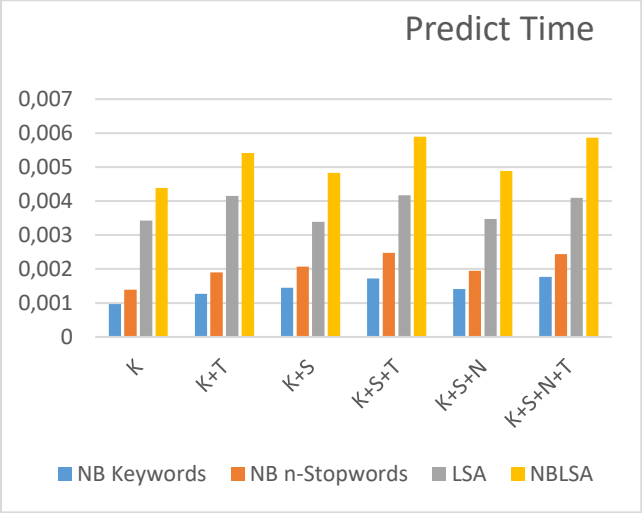
Pengujian waktu *training* dilakukan mulai dari pembacaan label sampai membentuk matriks *BoW*. Isi matriks bervariasi tergantung fitur-fitur yang digunakan. Hal ini dilakukan sebanyak 100 kali dengan jumlah data 1500. Hasilnya kemudian dirata-rata. Lama proses *training Naive Bayes* dapat dilihat pada Gambar 12. Terlihat apabila menggunakan fitur *stem* menyebabkan proses *training* lebih lama, karena pada *library stemmer* akan dicocokkan pada *dictionary* terlebih dahulu, untuk kemudian dicari imbuhannya dan diubah menjadi kata dasarnya saja

5.4 Pengujian Waktu *Testing*

Pengujian waktu *testing* dilakukan mulai dari pembacaan komentar sampai menentukan hasil klasifikasi. Hal ini dilakukan sebanyak 100 kali dengan jumlah data 1500. Hasilnya kemudian dirata-rata. Pada Gambar 13. dapat dilihat perbandingan waktu testing untuk masing-masing metode. Terlihat bahwa durasi *LSA* selalu lebih lama dibandingkan 2 metode yang lain. Hal ini disebabkan karena *LSA* melakukan proses untuk semua list keyword dalam matriksnya.



Gambar 12. Waktu *Training*



Gambar 13. Waktu *Testing*

6. KESIMPULAN

Setelah dilakukan perancangan sistem, pengimplementasian, dan pengujian terhadap aplikasi yang telah dibuat, dapat ditarik kesimpulan sebagai berikut:

- Metode *Naive Bayes* dan *LSA* dapat digunakan untuk melakukan klasifikasi topik, sentimen, tindakan.
- Metode *Naive Bayes* dan menggunakan fitur *keywords* saja merupakan kombinasi yang terbaik dalam hal akurasi untuk melakukan analisis sentimen yaitu 87.87 % untuk penentuan topik, 59.2 % untuk penentuan tindakan, 55.67% untuk penentuan sentimen, dan untuk akurasi program sebesar 29.87 %.
- Metode *Naive Bayes* dan menggunakan fitur *keywords* saja merupakan kombinasi yang terbaik dalam hal waktu *testing* untuk melakukan analisis sentimen yaitu sebesar 0.00009626 detik. Sedangkan untuk LSA adalah 0.0034251
- Metode *Naive Bayes* dan menggunakan fitur *keywords* saja merupakan kombinasi yang terbaik dalam hal waktu *training* untuk melakukan analisis sentimen yaitu sebesar 0.000319 detik. Sedangkan waktur *training* tercepat dari *LSA* adalah 0.000321 detik

Saran untuk pengembangan kedepannnya adalah:

- Dapat menambahkan jumlah *training* data. Misalnya dengan menyebarkan kuesioner kepada mahasiswa
- Meminta bantuan tenaga ahli untuk proses pelabelan data
- Menambahkan jumlah topik, sehingga tidak hanya 5 topik saja yang menyebabkan terlalu banyak komentar masuk ke dalam topik “lain”
- Mencoba menggunakan metode Jaringan Saraf Tiruan seperti *Convolute Neural Network* atau *Recurrent Neural Network*
- Memperbaiki *interface* agar lebih menarik dan *user friendly*
- Melakukan optimasi waktu pada saat melakukan *stemming*
- Mencoba melakukan *tagging* dengan tetangga yang lebih banyak
- Melengkapi *summary*. Misalnya objek, lokasi, atribut, dll
- *Sentiment Score* menggunakan menerapkan ilmu *Fuzzy*

7. DAFTAR PUSTAKA

[1] Anand, D., & Naorem, D. 2016. Semi-supervised Aspect Based Sentiment Analysis for Movies using Review Filtering. *7th International Conference on Intelligent Humant Computer Interaction*, (hal. 86-93).

[2] Bhattacharjee, S., Das, A., Bhattacharya, U., Parui, S. K., & Roy, S. 2015. Sentiment Analysis Using Cosine Similarity Measure. *International Journal of Computer Applications*, (hal. 17-19).

[3] Dey, L., Chakraborty, S., Biswas, A., Bose, B., & Tiwari, S. 2016. Sentiment Analysis of Review Datasets Using Naive Bayes and K-NN Classifier.

[4] Provost, F dan Fawcett, T. 2013. 11 Januari 2018. Data Science for Business. O’Reilly

[5] Saif, H., He, Y., & Alani, H. 2012. Semantic Sentiment Analysis of Twitter. *The Semantic Web - ISWC 2012*, (hal. 508-524).

[6] *Sastrawi*. 2017. Diambil kembali dari https://github.com/sastrawi/sastrawi

[7] Vinodhini, G., & Chandrasekaran, R. 2012. Sentiment Analysis and Opinion Mining: A Survey. *International Journal of Advanced Research in Computer Science and Software Engineering*, (hal. 282-293).

[8] Wang, L., & Wan, Y. 2011. Sentiment Classification of Documents Based on Latent Semantic Analyisis. *Advanced Research on Computer Education, Simulation and Modeling*, (hal. 356-361).