

Aplikasi Mobile untuk Analisis Sentimen pada Google Play

Lutfi Budi Ilmawan*¹, Edi Winarko²

¹Progdi S2/S3 Ilmu Komputer, FMIPA UGM, Yogyakarta

²Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta

Email: *luthfy13@live.com, ewinarko@ugm.ac.id

Abstrak

Google dalam application store-nya, Google Play, saat ini telah menyediakan sekitar 1.200.000 aplikasi mobile. Dengan sejumlah aplikasi tersebut membuat pengguna memiliki banyak pilihan. Selain itu, pengembang aplikasi mengalami kesulitan dalam mencari tahu bagaimana meningkatkan kinerja aplikasinya. Dengan adanya permasalahan tersebut, maka dibutuhkan sebuah aplikasi analisis sentimen yang dapat mengolah sejumlah komentar untuk memperoleh informasi.

Sistem yang dibangun memiliki tujuan untuk menentukan polaritas sentimen dari ulasan tekstual aplikasi pada Google Play yang dilakukan dari perangkat mobile. Perangkat mobile memiliki portabilitas yang tinggi dan sebagian dari perangkat tersebut memiliki resource yang terbatas. Hal tersebut diatasi dengan menggunakan arsitektur sistem berbasis client server, di mana server melakukan tugas-tugas yang berat sementara client-nya adalah perangkat mobile yang hanya mengerjakan tugas yang ringan. Dengan solusi tersebut maka Analisis sentimen dapat diaplikasikan pada mobile environment.

Adapun metode klasifikasi yang digunakan adalah Naïve Bayes untuk aplikasi yang dikembangkan dan Support Vector Machine Linier sebagai pembanding. Nilai akurasi dari Naïve Bayes classifier dari aplikasi yang dibangun sebesar 83,87% lebih rendah jika dibandingkan dengan nilai akurasi dari SVM Linier classifier sebesar 89,49%. Adapun penggunaan semantic handling untuk mengatasi sinonim kata dapat mengurangi akurasi classifier.

Kata kunci— analisis sentimen, google play, klasifikasi, naïve bayes, support vector machine

Abstract

Google's Google Play now providing approximately 1.200.000 mobile applications. With these number of applications, it makes the users have many options. In addition, application developers have difficulties in figuring out how to improve their application performance. Because of these problems, it is necessary to make a sentiment analysis applications that can process review comments to get valuable information.

The purpose of this system is determining the polarity of sentiments from applications' textual reviews on Google Play that can be performed on mobile devices. The mobile device has high portability and the majority of these devices have limited resource. That problem can be solved by using a client server based system architecture, where the server performs training and classification tasks while clients is a mobile device that perform some of sentiment analysis task. With this solution, the sentiment analysis can be applied to the mobile environment.

The classification method that used are Naïve Bayes for developed application and Linear Support Vector Machine that is used for comparing. Naïve Bayes classifier's accuracy is 83.87%. The result is lower than the accuracy value of Linear SVM classifier that reach 89.49%. The use of semantic handling can reduce the accuracy of the classifier.

Keywords—sentiment analysis, google play, classification, naïve bayes, support vector machine

1. PENDAHULUAN

Saat ini telah tersedia sekitar 1.200.000 aplikasi mobile pada Gogle Play. Dengan sejumlah aplikasi tersebut membuat pengguna memiliki banyak pilihan. Untuk mengatasi masalah tersebut, pertama-tama mereka harus mengetahui fungsi dari aplikasi yang akan di-install, bagaimana pendapat konsumen lain terhadap aplikasi tersebut dan kemudian memutuskan membeli atau tidak. Biasanya, ulasan pengguna berisi dua bagian, yaitu nilai rating dan komentar secara tekstual. Nilai rating menunjukkan evaluasi keseluruhan pengalaman pengguna menggunakan skala numerik, namun komentar tekstual mampu bercerita lebih mendalam. Setelah beberapa bulan, mungkin ada lebih dari sepuluh ribu komentar tekstual dari aplikasi baru yang diluncurkan di market. Hal ini sangat menantang untuk pengguna untuk membaca semua komentar tersebut untuk membuat keputusan. Selain itu, pengembang aplikasi mengalami kesulitan dalam mencari tahu bagaimana meningkatkan kinerja aplikasi berdasarkan dari ribuan komentar tekstual. Dengan adanya permasalahan tersebut, maka dibutuhkan sebuah aplikasi analisis sentimen yang dapat mengolah sejumlah komentar untuk memperoleh informasi, informasi ini akan memiliki nilai tambah sebagai pengetahuan akan sesuatu. Analisis sentimen adalah teknik yang sesuai untuk menganalisis ulasan tekstual konsumen tentang produk dan layanan. Namun aplikasi untuk analisa sentimen untuk ulasan tekstual aplikasi pada Google Play yang dikembangkan masih belum banyak. Tugas dasar dalam analisis sentimen adalah mengklasifikasikan polaritas dari teks yang berisi tentang pendapat dan emosi.

Metode yang digunakan untuk mengklasifikasikan ulasan tekstual aplikasi adalah metode klasifikasi *Naïve Bayes*. Algoritmanya sederhana namun memiliki kecepatan dan akurasi yang tinggi[1]. Pada penelitian yang dilakukan oleh [2], *Naïve Bayes* dapat bekerja dengan baik bahkan dengan adanya kehadiran dari fitur yang memiliki dependensi yang kuat pada *dataset*. Data komentar yang diambil dari Google Play untuk dijadikan *data training* memiliki jumlah yang sedikit karena sulitnya mendapatkan komentar yang sesuai. Dengan permasalahan ini, metode klasifikasi *Naïve Bayes* sangat sesuai apabila digunakan sebab *Naïve Bayes Classifier* masih mampu bekerja dengan baik dengan ukuran *data training* yang kecil [3][2]. Selain dari itu, algoritma *Naïve Bayes* yang sederhana dan kecepatannya yang tinggi dalam proses pelatihan dan klasifikasi [4], membuat algoritma ini menarik untuk digunakan sebagai salah satu metode klasifikasi. Dengan karakteristik tersebut di atas, maka metode klasifikasi *Naïve Bayes* sesuai jika digunakan pada penelitian ini. Sebagai pembanding akan digunakan metode SVM (*Support Vector Machine*). Kotsiantis [4], memperlihatkan pada penelitiannya bahwa akurasi dari metode SVM lebih besar jika dibandingkan dengan metode klasifikasi yang lainnya, memiliki kecepatan klasifikasi yang tinggi, dan tolenransi yang tinggi terhadap attribut yang tidak relevan.

2. METODOLOGI PENELITIAN

Berikut adalah uraian dari tahapan dan metode yang digunakan dalam penelitian :

2.1 Studi Kepustakaan

Pengumpulan bahan referensi, seperti jurnal penelitian, prosiding, tesis, buku-buku teori dan sumber-sumber lain termasuk informasi yang diperoleh melalui internet.

2.2 Analisis dan Perancangan Sistem

Proses yang dilakukan dalam analisa dan perancangan sistem terbagi dalam beberapa bentuk diantaranya adalah sebagai berikut:

2.2.1 Analisis Sistem

Sistem yang dibangun pada penelitian ini merupakan aplikasi *mobile* yang mampu

menentukan sentimen aplikasi pada Google Play melalui ulasan komentar yang diberikan oleh *user*. Dalam penelitian ini, pendekatan yang digunakan merupakan pendekatan *supervised learning*. Metode klasifikasi yang digunakan adalah metode *Multinomial Naïve Bayes*.

Proses evaluasi kinerja *classifier* menggunakan pendekatan *k-fold cross-validation*. Hasil klasifikasi yang diprediksi benar dan tidak benar oleh model klasifikasi ditabulasikan pada sebuah tabel *confusion matrix*. Informasi dalam *confusion matrix* diperlukan untuk menentukan kinerja model klasifikasi. Hasil dari *confusion matrix* digunakan untuk menentukan akurasi dari *classifier* dengan *performance metric accuracy*.

Sistem yang dibangun merupakan sistem yang diaplikasikan pada *mobile environment*. Jeong dan Kim [5], menyebutkan dalam penelitiannya bahwa perangkat *mobile* memiliki *resource* yang terbatas. Proses *training* dan klasifikasi yang berat menjadi masalah jika diaplikasikan untuk perangkat *mobile* yang memiliki spesifikasi yang rendah antara lain seperti kapasitas *storage* yang kecil dan *processor* yang lambat. Untuk mengatasi masalah tersebut, sistem yang dibangun diharapkan memiliki karakteristik sebagai berikut:

1. Sistem yang dibangun menggunakan arsitektur dengan basis *client server*, di mana *server* menyediakan *web service* untuk berkomunikasi dengan *client* berupa perangkat *mobile*.
2. Aplikasi *mobile* yang dibangun dapat mengambil komentar-komentar aplikasi pada Google Play selanjutnya melakukan tahap *preprocessing*.
3. Aplikasi *mobile* yang dibangun dapat melakukan visualisasi hasil analisa sentimen dari *web service* ke *user* melalui *interface*-nya.
4. *Web service* yang disediakan oleh *server* dapat melakukan proses *training data* dan klasifikasi.
5. Sistem yang digunakan dapat mengukur akurasi dari sebuah *classifier* dengan menggunakan metode *k fold cross validation*.

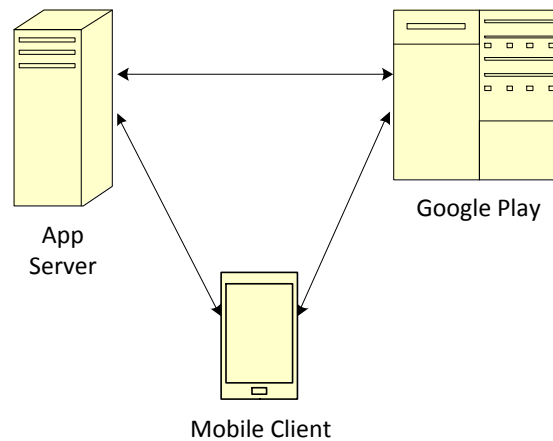
2.2.2 Perancangan Sistem Mobile Sentiment Analysis

Sistem yang dibangun memiliki tujuan untuk menentukan polaritas sentimen dari ulasan tekstual aplikasi pada Google Play yang dilakukan dari perangkat *mobile*. Dalam penelitian ini, komentar diklasifikasikan menjadi tiga buah polaritas sentimen, yaitu sentimen positif, negatif dan *crash*. Komentar diambil berdasarkan dari *query* berupa nama aplikasi, hasil *query* untuk *data training* dan hasil *query* oleh *user* melalui tahap *preprocessing*. Data hasil *preprocessing* untuk *data training* yang telah diberi label digunakan untuk proses *training*. Adapun metode klasifikasi yang digunakan adalah *Multinomial Naïve Bayes*. Model klasifikasi ini kemudian digunakan untuk menentukan sentimen data komentar dari hasil *query* dari *user*.

Adapun perancangan sistem terdiri dari:

1. Perancangan arsitektur sistem

Gambaran umum dari arsitektur sistem yang dirancang dapat dilihat pada Gambar 1. Terdapat tiga komponen, komponen tersebut terdiri dari *data provider*, *app server*, dan *mobile client*. *App server* melakukan *request* untuk mendapatkan komentar dari Google Play kemudian Google Play mengembalikan data ke *app server* berupa komentar-komentar yang akan dijadikan sebagai *data training*. *Data training* digunakan untuk membangun model klasifikasi melalui proses *training*. Adapun *mobile client* melakukan *request komentar* ke Google Play. Setelah *mobile client* mendapatkan komentar tersebut selanjutnya akan melalui tahap *preprocessing* kemudian dikirimkan ke *app server* untuk diklasifikasikan, data yang dikembalikan dari *app server* ke *mobile client* merupakan hasil klasifikasi. *Data provider* merupakan sumber penyedia data yang nantinya akan diproses, dalam hal ini adalah Google Play Store. Dengan arsitektur sesuai pada Gambar 1, diharapkan dapat meminimalisir penggunaan *resource* perangkat *mobile* agar nantinya sistem ini juga dapat digunakan pada perangkat *mobile* dengan spesifikasi perangkat keras yang rendah.



Gambar 1 Arsitektur sistem *mobile sentiment analysis*

2. Komponen dan proses pada perangkat lunak

Pada bagian ini akan dijelaskan tentang komponen dan proses dari *app server* dan *mobile client*.

- *App Server*

Komponen pada *app server* terdiri dari *training data* dan *classification service*. *Training data* berhubungan dengan Google Play sebagai *data provider*, dimulai dengan *request* komentar untuk dijadikan sebagai *data training* ke Google Play. Setelah komentar diambil, komentar tersebut melalui tahap *preprocessing*. Setelah melalui tahap tersebut, maka selanjutnya melalui proses *training data* untuk membangun model klasifikasi. Adapun metode klasifikasi yang digunakan adalah metode klasifikasi *Multinomial Naïve Bayes*. Komponen *classification service* merupakan *web service* untuk berkomunikasi dengan *mobile client*. *Mobile client* melakukan *request* ke *classification service* dengan *entity* data berupa hasil *preprocessing* komentar. Hasil *preprocessing* yang diterima oleh *classification service* kemudian dipecah dan dicocokkan pada model klasifikasi untuk proses penentuan nilai probabilitas kondisionalnya. Selanjutnya proses klasifikasi dilakukan. Adapun data yang dikembalikan ke *mobile client* berupa hasil klasifikasi.

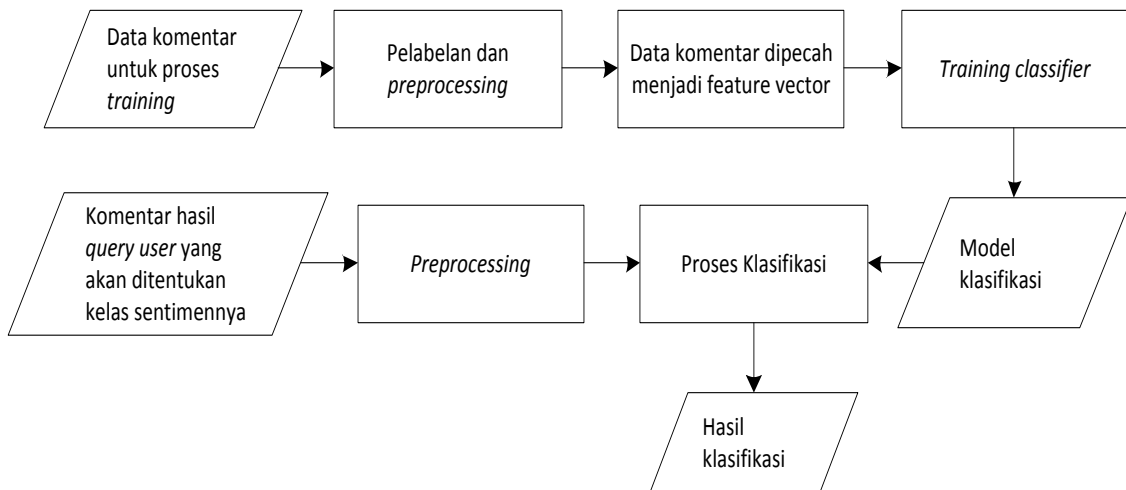
- *Mobile Client*

Mobile client merupakan aplikasi yang terdapat pada *perangkat mobile*. Tidak semua *perangkat mobile* mampu melakukan keseluruhan proses dari analisis sentimen, maka dari itu hanya sebagian dari proses tersebut dilakukan pada *perangkat mobile* yakni *preprocessing*. *Preprocessing* merupakan operasi yang tidak berat untuk dilakukan oleh *perangkat mobile*[6]. Pertama-tama *mobile client* melakukan *request* komentar ke Google Play berdasarkan *query* yang dimasukkan user berupa nama aplikasi. Komentar yang didapatkan kemudian melalui tahap *preprocessing* kemudian hasil *preprocessing* tersebut dijadikan *entity* untuk dikirimkan ke *classification service* yang terdapat pada *app server*. Hasil klasifikasi dikembalikan ke *mobile client* kemudian divisualisasikan ke user.

3. Perancangan proses analisis sentimen

Gambaran umum dari proses analisis sentimen menggunakan metode klasifikasi *Multinomial Naïve Bayes* dari sistem yang dibangun dapat dilihat pada Gambar 2. Data komentar diambil dari Google play untuk membangun *data training*, data yang diambil hanya yang memiliki nilai *rating* 1 dan nilai *rating* 5, data tersebut kemudian melalui proses pelabelan data. Selanjutnya melalui tahap *preprocessing*. Setelah melalui tahap pelabelan dan *preprocessing*, komentar-komentar tersebut siap untuk dijadikan sebagai *data training*, komentar-komentar kemudian dipecah menjadi *feature list*. Dari *feature list* ini akan dihasilkan sebuah daftar kata unik untuk dijadikan sebagai *vocabulary*.

Selanjutnya proses *training classifier* dapat dilakukan untuk membangun model klasifikasi berupa nilai probabilitas kondisional tiap fitur untuk tiap kelas. Model klasifikasi kemudian digunakan untuk melakukan proses klasifikasi terhadap komentar-komentar yang merupakan hasil dari *query user* yang terlebih dahulu melalui tahap *preprocessing*.



Gambar 2 Gambaran umum proses analisis sentiment

- Proses pelabelan

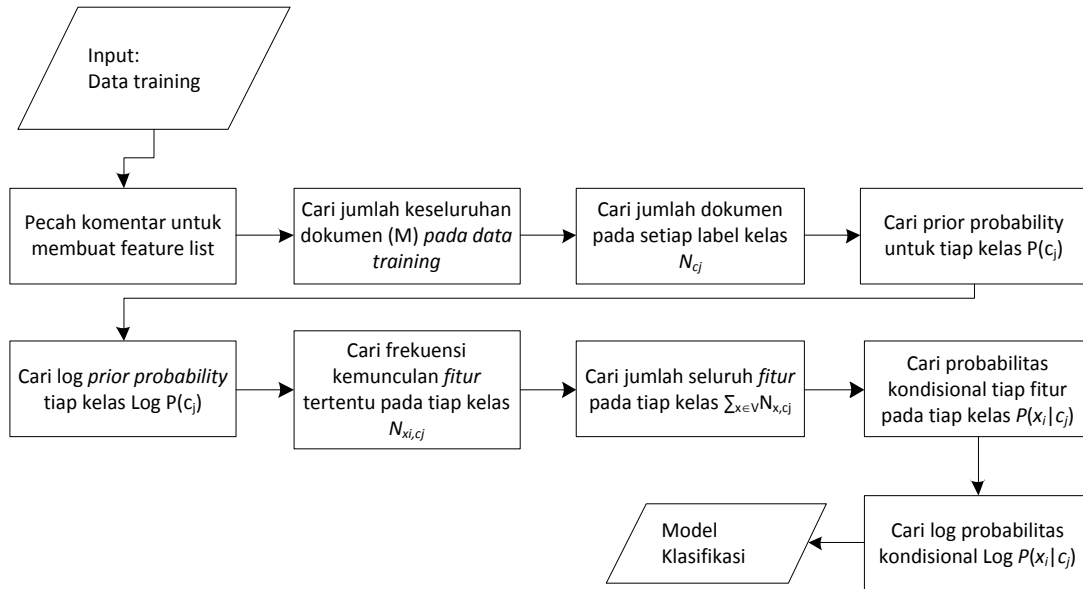
Proses pelabelan dilakukan dengan memanfaatkan nilai rating yang terdapat pada komentar yang diambil dari Google Play. Komentar yang digunakan untuk dijadikan sebagai data training hanya komentar yang memiliki nilai rating 1 dan nilai rating 5. Adapun langkah-langkah pada proses pelabelan sebagai berikut :

- Komentar yang memiliki nilai *rating* selain 1 dan 5 akan dihapus dari *database*.
- Seluruh komentar yang memiliki nilai *rating* 5 diberikan label kelas positif.
- Proses berikutnya menghapus komentar yang memiliki nilai rating 5 dan mengandung *smiley* negatif. Contoh dari *smiley* negatif tersebut adalah “ :(”, “ :-(”, “ :(”, dan “ ;-(”.
- Komentar yang memiliki nilai *rating* 5 dan mengandung kata-kata tertentu yang sering digunakan pada komentar yang memiliki sentimen *crash* antara lain *crash, error, bug, freeze, stuck, invalid, problem, not load, force close, not open, not start, not work, not respond* dan lain sebagainya akan dihapus. Namun jika komentar tersebut mengandung kata seperti di sebutkan di atas dan kemudian dinegasasikan menjadi *not crash, no problem, no bug*, dan lain sebagainya, maka komentar tersebut tidak akan dihapus. Hal ini dilakukan, sebab sebagian kecil komentar dengan rating 5, berisi komentar dengan sentimen negatif ataupun *crash*.
- Selanjutnya komentar yang memiliki nilai *rating* 1 dan mengandung kata-kata tertentu yang sering digunakan pada komentar yang memiliki sentimen *crash* akan dilabeli dengan sentimen *crash*.
- Proses berikutnya memberikan label kelas negatif pada komentar yang belum memiliki kelas sentimen.
- Proses terakhir, komentar dengan sentimen negatif yang mengandung kata *fix, cant* dan *nice* dihapus. Kata-kata tersebut biasanya digunakan untuk komentar dengan sentimen *crash* atau *postif*.

- Proses *preprocessing*
Adapun tahapan dalam melakukan preprocessing meliputi:
 - *Casefolding*, yakni mengubah seluruh karakter huruf pada komentar menjadi huruf kecil.
 - *Punctuation and stopwords elimination*. *Punctuation elimination* yakni menghapus seluruh karakter yang tidak termasuk huruf alphabet lalu mengganti karakter tersebut dengan spasi. Sedangkan *stopword elimination* menghapus kata-kata umum seperti *the, am, is, are, do, does, did*, dan lain sebagainya. Pada penelitiannya [7], mengatakan bahwa pada sistem modern information retrieval, indexing yang efektif dapat didapat dengan pengeliminasian kata-kata umum. Kata-kata umum dan tanda baca dapat mempengaruhi efektivitas pada *information retrieval* karena memiliki frekuensi yang sangat tinggi dan cenderung mengurangi dampak dari perbedaan frekuensi antara kata-kata yang kurang umum, sehingga nantinya akan mempengaruhi proses *training*.
 - *Negation Tag*, yakni sebuah tag “*NOT_*” ditambahkan pada setiap kata di antara kata negasi dan kata yang mengikutinya, lalu menghapus kata negasinya. Akurasi *classifier* dalam analisis sentimen dipengaruhi pada pendeteksian negasi menurut penelitian yang telah dilakukan sebelumnya [8]. Menurut [9], bahwa fitur dengan kata negasi merupakan fitur yang penting dalam analisis sentimen. Kata negasi digunakan untuk membedakan antara informasi faktual dan non faktual pada proses ekstraksi informasi untuk analisis sentimen [8]. Sebagai contoh, “*not good*” sama halnya dengan mengatakan “*bad*”. Dengan mengadaptasikan teknik yang digunakan oleh [10], sebuah tag “*NOT_*” ditambahkan pada setiap kata di antara kata negasi dan kata yang mengikutinya, lalu menghapus kata negasinya. Contohnya seperti “*I do not like this movie*” menjadi “*I do NOT_like this movie*”. Jika fitur yang digunakan hanya unigram, maka kata *like* akan memiliki polaritas dengan sentimen positif, namun jika ditambahkan tag negasi menjadi *NOT_like* maka polaritas sentimennya tentu akan berubah dan akan mempengaruhi seluruh kalimatnya.
- Perancangan *Naïve Bayes Classifier*
Proses *trainingclassifier* dapat dilihat pada Gambar 3 dimulai dengan mengambil data *training* berupa komentar-komentar yang telah diberi label kelas dan telah melalui tahap *preprocessing*. Selanjutnya memecah komentar-komentar tersebut per kata untuk menghasilkan *feature list*. Proses selanjutnya mencari jumlah keseluruhan dokumen M pada *data training* dan jumlah dokumen pada masing-masing label kelas N_{c_j} .

$$P(c_j) = \frac{N_{c_j}}{M} \quad (1)$$

Setelah jumlah keseluruhan dokumen didapatkan dan jumlah dokumen pada masing-masing label kelas, maka proses menghitung nilai *prior probability* $P(c_j)$ tiap kelas dapat dilakukan sesuai dengan persamaan (1) yakni membagi nilai N_{c_j} dengan M . Selanjutnya melakukan operasi log terhadap nilai *prior probability* tersebut menjadi $\log P(c_j)$



Gambar 3 Flowchat proses training classifier

Proses selanjutnya yakni menghitung nilai frekuensi kemunculan fitur pada tiap kelas N_{x_i, c_j} . Selanjutnya menghitung jumlah frekuensi kemunculan seluruh fitur tiap kelasnya $\sum_{x \in V} N_{x, c_j}$. Proses selanjutnya menghitung probabilitas kondisional $P(x_i | c_j)$ fitur x_i pada kelas c_j . Penentuan nilai $P(x_i | c_j)$ menggunakan *laplace estimate* sesuai dengan persamaan (2).

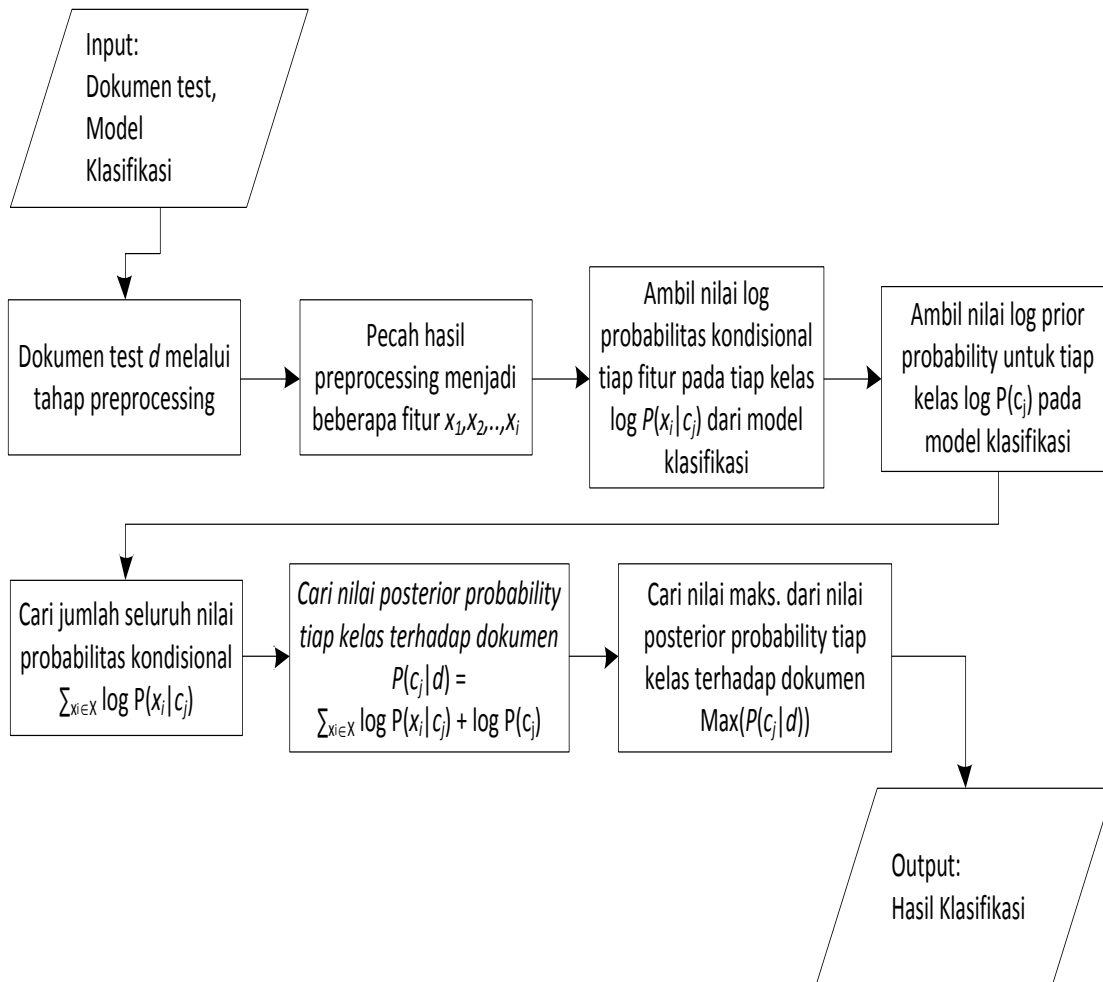
$$P(x_i | c_j) = \frac{N_{x_i, c_j} + 1}{\sum_{x \in V} N_{x, c_j} + 1} = \frac{N_{x_i, c_j} + 1}{(\sum_{x \in V} N_{x, c_j}) + B} \quad (2)$$

Di mana B merupakan jumlah kata unik pada vocabulary V . Selanjutnya melakukan operasi log terhadap nilai probabilitas kondisional $P(x_i | c_j)$ menjadi $\log P(x_i | c_j)$. Model klasifikasi yang dihasilkan berupa nilai log *prior probability* tiap kelas $\log P(c_j)$ dan nilai log probabilitas kondisional tiap fitur pada tiap kelas $\log P(x_i | c_j)$.

Proses klasifikasi data komentar sesuai dengan persamaan (3), yakni mencari nilai maksimum dari *posterior probability* tiap kelas terhadap sebuah dokumen $P(c_j | d)$.

$$c_{map} = \arg \max_{c_j \in C} \log P(c_j) + \sum_{x_i \in X} \log P(x_i | c_j) \quad (3)$$

Proses klasifikasi dimulai dengan melakukan *preprocessing* terhadap dokumen *test* yang akan diklasifikasikan kemudian memecah hasil preprocessing menjadi beberapa fitur. Selanjutnya mencari nilai log probabilitas kondisional dari tiap fitur dari dokumen dengan mengambil nilai log probabilitas kondisional yang terdapat pada model klasifikasi. Selanjutnya menambahkan seluruh nilai log probabilitas kondisional $\sum_{x_i \in X} \log P(x_i | c_j)$ dengan nilai log prior probability $P(c_j)$ untuk menghasilkan nilai posterior probability suatu kelas terhadap sebuah dokumen $P(c_j | d)$. Selanjutnya mencari nilai maksimum dari tiap nilai $P(c_j | d)$ tersebut. Adapun *flowchart* untuk proses klasifikasi dapat dilihat pada Gambar 4.



Gambar 4 Flowchart proses klasifikasi

4. Perancangan proses pengujian akurasi classifier

Pengujian untuk evaluasi *classifier* menggunakan metode *k-fold cross-validation* dengan nilai $k=3$. Data untuk pengujian diambil dari data awal yang digunakan sebagai *data training*. Jumlah data yang diambil, sama untuk setiap kelas sentimennya. Data kemudian dibagi menjadi tiga partisi yang terdiri dari komentar-komentar yang jumlahnya sama dari setiap label kelas. Salah satu partisi digunakan untuk proses testing sementara dua partisi lainnya digunakan untuk proses training. Prosedur ini diulangi sebanyak 3 kali sedemikian sehingga setiap partisi digunakan sebagai data testing tepat satu kali. Total error ditentukan dengan menjumlahkan error yang didapatkan selama proses evaluasi. Hasil evaluasi kemudian ditabulasikan pada sebuah confusion matrix, selanjutnya menghitung akurasi classifier dengan menggunakan persamaan (4).

$$\text{Akurasi} = \frac{\text{Jumlah klasifikasi benar}}{\text{Jumlah Data Uji}} \times 100\% \quad (4)$$

Metode pembandingan yang digunakan pada penelitian ini adalah metode *Support Vector Machine* (SVM) linier, untuk implementasinya menggunakan library LibSVM [11] Selanjutnya pada proses pengujian akurasi classifier ditambahkan juga proses *semantic handling* dengan menggunakan database Wordnet [12] untuk mengatasi permasalahan dari model *bag of word*. SVM linier *classifier* hanya digunakan untuk proses pengujian akurasi, tidak digunakan untuk implementasi pada *mobile client*. Adapun pengujian akurasi *classifier* diimplementasikan dalam bentuk *desktop application*.

2.3 Implementasi

Pada penelitian ini, pembuatan Aplikasi *Mobile* untuk Analisis Sentimen pada Google Play diimplementasikan dengan bahasa pemrograman java, android SDK dan python. Bahasa pemrograman java dan python digunakan untuk server sedangkan untuk pembuatan aplikasi mobile digunakan android SDK. Aplikasi yang dibuat menggunakan arsitektur *client server*. Aplikasi *mobile* yang dibuat memerlukan sebuah *server* sebab proses komputasi yang dilakukan cukup berat, yaitu untuk proses *training* dan klasifikasi. Proses *training* dan klasifikasi pada *server* menggunakan metode klasifikasi *Multinomial Naïve Bayes*, sementara dalam pembuatan web *service*-nya menggunakan library Flask untuk python. Program untuk menjalankan *script* python tersebut menggunakan aplikasi java yang tersimpan pada *server*. Aplikasi java tersebut juga bertugas untuk melakukan proses *training data*.

3. HASIL DAN PEMBAHASAN

Data komentar untuk data training diambil dari beberapa aplikasi yang terdapat pada Google Play. Jumlah data yang digunakan 9298 komentar, komentar positif sebanyak 3104 komentar, komentar negatif sebanyak 3092 komentar, dan komentar crash sebanyak 3102 komentar untuk membangun data training. Adapun data yang digunakan untuk proses pengujian akurasi classifier diambil dari data training dengan total 9000 komentar yang terdiri dari 3000 komentar dengan sentimen positif, 3000 komentar untuk sentimen negatif, dan 3000 komentar untuk sentimen crash.

Evaluasi *classifier* menggunakan metode *3-folds cross-validation*. Selama proses pengujian digunakan 3 partisi data yang berukuran sama, salah satu dari partisi dipilih untuk *testing*, sedangkan dua partisi lainnya digunakan untuk *training*. Prosedur ini diulangi sebanyak 3 kali sedemikian sehingga setiap partisi digunakan sebagai *datatesting* tepat satu kali. Total *error* ditentukan dengan menjumlahkan *error* yang didapatkan selama proses pengujian. Data untuk pengujian diambil dari tabel *data_awal* dengan mengambil 3000 komentar dengan label kelas positif, 3000 komentar dengan label kelas negatif, dan 3000 komentar dengan label kelas *crash*, jadi total data yang digunakan sebanyak 9000 data. Dalam satu partisi terdapat 3000 komentar, yakni 1000 komentar dengan sentimen positif, 1000 komentar dengan sentimen negatif, dan 1000 komentar dengan sentimen *crash*. Hasil akurasi *classifier* dihitung berdasarkan persamaan 4.

Pengujian akurasi *classifier* dilakukan dengan menggunakan fitur yang berbeda-beda sehingga menghasilkan *data set* yang berbeda untuk hasil *preprocessing*-nya. Adapun fitur yang digunakan untuk proses pengujian terdiri dari fitur *unigram*, *unigram* dengan *semantic handling*, *negation tag*, dan *negation tag* dengan *semantic handling*. Berikut sampel dari sebagian komentar yang digunakan sebagai data testing serta hasil *preprocessing*-nya.

Tabel 1 Komentar untuk data testing

No.	Komentar	Fitur	Hasil <i>Preprocessing</i>	Hasil Klasifikasi
1.	<i>Fun This game is very addictive, I've won big in the past and the game froze up on me. I wrote a message to playport and didn't get a response! Some of the pictures on here are gross. I get messages from people that are disgusting!</i>	<i>Unigram</i>	<i>fun game addictive ive won big past game froze wrote message playport didnt get response pictures gross get messages people disgusting</i>	Negatif
		<i>Unigram dengan semantic handling</i>	<i>fun game addictive ive South Korean won large past game freeze write message playport didnt get response picture gross get message people disgust</i>	Negatif
		<i>Negation tag</i>	<i>fun game addictive ive won big past game froze wrote message playport NOT_get response pictures gross get messages people disgusting</i>	Positif
		<i>Negation tag dengan semantic handling</i>	<i>fun game addictive ive South Korean won large past game freeze write message playport NOT_get response picture gross get message people disgust</i>	Negatif
2.	<i>Really? I have had issues with this app in the past. Now the website is bugging. I have success logging in, but now the app has the number 4 and nothing else</i>	<i>Unigram</i>	<i>really have had issues app past website bugging have success logging app has number</i>	Negatif
		<i>Unigram dengan semantic handling</i>	<i>truly rich person have issue app past web site tease rich person success logging app hour angle number</i>	Negatif
		<i>Negation tag</i>	<i>really have had issues app past website bugging have success logging app has number</i>	Crash
		<i>Negation tag dengan semantic handling</i>	<i>truly rich person have issue app past web site tease rich person success logging app hour angle number</i>	Negatif

Selain hasil *preprocessing*, Tabel 1 juga menunjukkan hasil klasifikasi dari *Naïve Bayes classifier*. Hasil klasifikasi yang didapatkan berbeda-beda, hal ini disebabkan oleh penggunaan fitur. Berikut perbandingan hasil akurasi *classifier* berdasarkan metode klasifikasi dan fitur yang digunakan seperti terlihat pada Tabel 2.

Tabel 2 Hasil akurasi classifier

No.	Fitur	Hasil Akurasi	
		<i>Naïve Bayes</i>	SVM Linier
1.	<i>Unigram</i>	82,82%	86,63%
2.	<i>Unigram dengan semantic handling</i>	82,44%	85,82%
3.	<i>Negation tag</i>	83,87%	89,49%
4.	<i>Negation tag dengan semantic handling</i>	83,84%	88,37%

4. KESIMPULAN

Berdasarkan dari hasil penelitian dan pembahasan yang dilakukan maka diperoleh kesimpulan sebagai berikut:

1. Analisis sentimen dapat diaplikasikan pada *mobile environment* dengan menerapkan arsitektur sistem berbasis *client server*, di mana *server* bertugas melakukan proses *training* dan klasifikasi sementara *client* yang berupa perangkat *mobile* hanya melakukan tugas yang ringan seperti proses *preprocessing*, *parsing* dan kirim komentar.
2. Nilai akurasi dari *Naïve Bayes classifier* dari aplikasi yang dibangun sebesar 83,87% lebih rendah jika dibandingkan dengan nilai akurasi dari *SVM Linier classifier* sebesar 89,49%.
3. Berdasarkan penggunaan fitur, nilai akurasi paling tinggi didapatkan dengan menggunakan fitur *negation tag* tanpa *semantic handling*.
4. Penggunaan *semantic handling* untuk mengatasi sinonim kata dapat mengurangi akurasi *classifier*.

5. SARAN

Untuk pengembangan penelitian lebih lanjut, diberikan saran sebagai berikut:

1. Sistem yang dibuat hanya dapat digunakan untuk menentukan sentimen dari komentar berbahasa inggris. Diharapkan pada penelitian selanjutnya, agar sistem yang dikembangkan dapat digunakan untuk berbagai macam bahasa.
2. Pada penelitian ini, fitur yang digunakan hanya unigram dan *negation tag*. Sistem yang dibuat dapat dikembangkan dengan menggunakan beberapa fitur lain seperti *n-grams* dan *POS tag* yang diharapkan dapat menambah akurasi *classifier*.
3. Android Market API yang digunakan merupakan *un-official* API untuk google play store dan memiliki banyak keterbatasan. Peneliti selanjutnya diharapkan dapat mengembangkan API untuk google play agar keterbatasan seperti nama *google user* tidak bisa ditampilkan, pengambilan komentar yang cukup lama, dan keterbatasan-keterbatasan lainnya bisa diatasi agar dapat digunakan pada penelitian selanjutnya

DAFTAR PUSTAKA

- [1] Rish, I., 2001, An Empirical Study of The Naive Bayes Classifier, *In Proceedings of IJCAI-01 workshop on Empirical Methods in AI*, New York.
- [2] Domingos, P. dan Pazzani, M.. 1997. On the Optimality of the Simple Bayesian Classifier under Zero-One Loss, *Machine Learning*, 29, 103-130.
- [3] Kohavi, R., 1996, Scaling Up The Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid, *In KDD-96 Proceedings*, AAAI, California.
- [4] Kotsiantis, S.B., 2007, Supervised Machine Learning: A Review of Classification Techniques, *Informatica*, No. 31, Hal. 249-68.
- [5] Jeong, K. dan Kim, W., 2012, Implementation of Mobile Application about Poly Context USN Data Mining and Display. *International Journal of Multimedia and Ubiquitous Engineering*, Vol. 7, No. 3, Hal. 19-26.
- [6] Talia, D. dan Trunfio, P., 2008, Mobile Data Mining on Small Devices Through Web Services, *Lab Report*, DEIS, University of Calabria, Italy.

-
- [7] Zou, F., Wang, F.L., Deng, X. dan Han, S., 2006, Automatic Identification of Chinese Stop Words. *Advances in Natural Language Proccesing Research in Computing Science*, Vol. 18, Hal. 151-162.
- [8] Lee, H.Y. dan Renganathan, H., 2011, Chinese Sentiment Analysis Using Maximum Entropy, *In Proceedings of the Workshop on Sentiment Analysis where AI meets Psychology*, IJCNLP, Chiang Mai, Thailand, November 13.
- [9] Li, S., Zhang, H., Xu, W., Chen, G., dan Guo, J., 2010, Exploiting Combined Multi-level Model for Document Sentiment Analysis, *In International Conference on Pattern Recognition*, Istanbul, August 23-26.
- [10] Das, S.R. dan Chen, M.Y., 2007, Yahoo! for Amazon: Sentiment Extraction from Small Talk on the Web, *Management Science*, Vol. 53, No. 9, Hal. 1375-1388.
- [11] Chang, C.-C. dan Lin, C.-J., 2011, LIBSVM : A Library for Support Vector Machines, *ACM Transactions on Intelligent Systems and Technology*, Vol. 2, No. 3, Hal. 1-27.
- [12] Miller, G.A., 1995, WordNet: A Lexical Database for English, *Communications of the ACM*, Vol. 38, No. 11, Hal. 39-41.