SMS *Filtering* Menggunakan *Naive Bayes Classifier* dan *FP-Growth Algorithm Frequent Itemset*

Dea Delvia Arifin[1], Shaufiah, ST.,MT.[2], M.Arif Bijaksana, Ir.,MTECH[3]

[1,2,3]Fakultas Informatika – Universitas Telkom, Bandung
[1]delvia.dea@gmail.com, [2]shaufiah@telkomuniversity.ac.id, [3]arifbijaksana@gmail.com

Abstrak

SMS (*Short Message Service*) masih menjadi pilihan utama sebagai media komunikasi walaupun sekarang ponsel semakin berkembang dengan berbagai media komunikasi aplikasi *messenger.* Seiring dengan berkembangnya berbagai media komunikasi lain, beberapa operator di beberapa negara menurunkan tarif SMS untuk tetap menarik minat pengguna ponsel. Namun penurunan tarif ini menyebabkan meningkatnya SMS *spam*, karena dimanfaatkan oleh beberapa pihak sebagai salah satu alternatif untuk iklan hingga penipuan. Hal itu menjadi permasalahan penting karena dapat mengganggu dan merugikan pengguna.

***Naive Bayes* dianggap sebagai salah satu *learning algorithm* yang sangat efektif dan penting untuk *machine learning* dalam *information retrieval. Naive Bayes* terbukti memiliki kinerja yang baik dalam klasifikasi teks dan deteksi SMS *spam* [2,10] dengan menunjukan akurasi yang tinggi. Dengan dikolaborasikan algoritma yang mampu menentukan *frequent itemset* dengan baik maka mampu menghasilkan tingkat akurasi yang lebih baik [2]. Karena tidak hanya setiap kata yang dianggap *mutually independent,* tetapi juga kata yang *frequent* sebagai kata yang *single, independent* dan *mutually exclusive* [2], sehingga mampu meningkatkan nilai peluang dan menyebabkan sistem lebih tepat dalam klasifikasi. Dalam hal ini digunakan *FP-Growth* untuk *mining frequent pattern* yang memiliki performansi yang baik dan efisien karena tidak membutuhkan pembangkitan kandidat *frequent* [4]. Hasil penelitian penggunan kolaborasi antara Naive Bayes dan FP-Growth menghasilkan akurasi rata-rata terbesar sebesar 98, 506% dan lebih unggul 0,025% dari metode tanpa melibatkan FP-Growth untuk dataset SMS Spam Collection v.1, serta meningkatkan nilai *precision* sehingga hasil klasifikasi lebih akurat.**

Kata Kunci: ekstraksi kata kunci, KEA, social media

Abstract

SMS (Short Message Service) is still the primary choice as a communication media although nowadays mobile phone is growing with a variety of communication media messenger application. Along with the development of various other communication media, some carriers in some countries decrease the SMS rates to keep attracting the mobile phone users. However, this tariff reduction led to increased SMS spam, as used by some people as an alternative to advertisement even fraud. It becomes an important issue because it can annoy and harm to the user.

Naive Bayes is considered as one of the learning algorithm which is very effective and important for machine learning in information retrieval. Naive Bayes proved to have a good performance in the classification of text and SMS spam detection [2, 10] by showing high accuracy. With the collaboration algorithm that is able to determine frequent itemset well then able to produce a better accuracy rate [2]. **Because not only considered each and every word as independent and mutually exclusive but also frequent words as a single, independent and mutually exclusive [2], so it increase probabilty value and lead system more accurate to classification.** ***In this case, FP-growth is used for mining frequent pattern that has good performance and efficient because it does not require frequent candidate generation [4]. Results of using collaboration of Naive Bayes and FP-Growth showed the highest average accuracy is 98, 506% and 0,025% better than without using FP-Growth for dataset SMS Spam Collection v.1, and improve the precision value so the classicication results more accurate.***

Keywords: keyword extraction, KEA, social media

1. Pendahuluan

SMS adalah sebuah media komunikasi berbentuk teks yang mengijinkan pengguna ponsel untuk saling berbagi teks pendek (biasanya kurang dari 160 karakter 7-bit) [10]. Seiring dengan penggunaannya yang semakin meluas dan popularitasnya sebagai media komunikasi

terpenting, banyak pihak yang memanfaatkan hal tersebut untuk kepentingan komersil seperti sebagai media iklan bahkan penipuan. Menurunnya tarif SMS menjadi salah satu penyebab juga semakin meningkatnya SMS *spam*, seperti di Cina tarif untuk SMS kurang dari $0.001 [10]. Jumlah SMS *junk* atau *SMS spam* semakin bertambah setiap harinya dan berdasarkan *Korea Information Security* (KISA), jumlah SMS *junk* ini melebihi email *spam*. Sebagai contohnya, pengguna ponsel di US mendapatkan 1,1 milyar SMS *spam* dan pengguna Cina juga menerima 8.29 milyar SMS *spam* dalam seminggu [13].

Salah satu solusi yang dapat dilakukan terhadap permasalahan diatas adalah melakukan *filtering* SMS dengan klasifikasi teks. Beberapa teknik yang populer untuk klasifikasi teks diantaranya *decision trees, Naive Bayes, rule induction, neural network, nearest neighbors,* dan *Support Vector Machine.* Namun dalam klasifikasi SMS ini berbeda dengan klasifikasi pada teks dokumen biasa atau email dikarenakan teks pada SMS sangat pendek (maksimal 160 7-bit karakter), banyak terdapat teks yang disingkat, dan cenderung tidak formal [10]. SMS yang terlalu pendek ini menimbulkan pertanyaan lain "apakah fitur yang digunakan cukup untuk membedakan antara SMS *spam* dengan *non-spam*?". Bahkan kini jenis SMS semakin bervariasi, sehingga dibutuhkan teknik lain untuk penambahan fitur yang dapat membedakan antara SMS *spam* dengan *non-spam.* Namun, dari setiap variasi SMS yang ada tetap memiliki pola yang serupa khususnya untuk sms *spam*, hal tersebut dapat menjadi landasan untuk digunakan teknik dengan melibatkan kemunculan kata-kata yang muncul bersamaan sebagai fitur tambahan untuk membedakan sms *spam* dan *non-spam*.

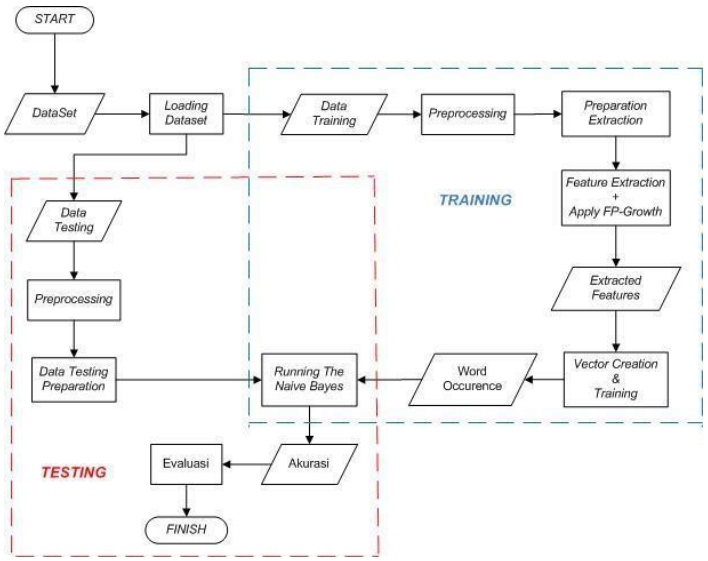
Dalam penelitian ini digunakan kolaborasi dua buah metode yaitu *Naive Bayes Classifier* dan *FP-Growth Algorithm Frequent Itemset*. *Naive Bayes* dianggap sebagai salah satu *learning algorithm* yang sangat efektif dan penting untuk *machine learning* dalam *information retrieval.* Selain itu berdasarkan paper yang diacu [2] menyatakan bahwa dengan menerapkan *minimum support* yang ditentukan pengguna, dapat meningkatkan akurasi dibandingkan dengan hanya menggunakan *Naive Bayes* saja. Karena dengan *minimum support* didapatkan *frequent itemset* sebagai fitur tambahan, sehingga tidak hanya setiap kata yang dianggap *mutually independent*, tetapi juga kata yang *frequent* sebagai kata yang *single, independent* dan *mutually exclusive* [2], maka mampu meningkatkan nilai peluang dan menyebabkan sistem lebih tepat dalam klasifikasi. Dalam paper acuan digunakan *Apriori Algorithm* dalam mendapatkan *frequent itemset*, namun pada penelitian ini dipilih menggunakan *FP-Growth Algorithm* yang memiliki kemampuan lebih baik dibandingkan *Apriori Algorithm* [4]. *FP-Growth* merupakan algoritma *mining frequent pattern* yang memiliki performansi yang baik dan efisien karena tidak membutuhkan pembangkitan kandidat *frequent pattern* [4].

2. Desain Sistem

Sistem yang dibangun secara umum terdiri dari 2 tahapan, yaitu proses *training* dan *testing* :

2.1 Proses

Proses *training* adalah proses pelatihan data untuk membentuk model klasifikasi dan proses *testing* adalah proses untuk menguji hasil klasifikasi berdasarkan model yang telah diperoleh.



Gambar 2-1. gambaran umum sistem

2.1.1 Preprocessing

Preprocessing pada data *training* dan *testing* dilakukan secara terpisah untuk memudahkan proses selanjutnya. *Preprocessing* dilakukan di tahap awal sebelum proses *training* (menggunakan data *training*) dan sebelum proses *testing* (menggunakan data *testing*). Tahapan yang dilakukan pada *preprocessing* adalah sebagai berikut :

1. *Case Folding* dan hapus karakter

Seluruh teks diubah menjadi huruf kecil untuk menyeragamkan data serta menghapus karakter selain huruf, angka dan menghapus tanda baca.

2. *Tokenisasi*

Sebelum dilakukan proses selanjutnya (proses 3 hingga proses 6), dilakukan tokenisasi terlebih dahulu untuk memecah string kalimat menjadi token atau satu kata individu, sehingga memudahkan dalam proses pencarian token.

3. *Handle Slang Words*

Pada dataset, banyak terdapat kata yang tidak formal atau istilah lain disebut dengan kata *slang*. Untuk menangani kata tersebut, maka dibuat kamus yang berisi kata-kata slang dilengkapi dengan arti kata sebenarnya. Daftar kata untuk kamus *slang* diambil dari *"Slang Dictionary-Text Slang & Internet Slang Words"* pada situs http://www.noslang.com/dictionary/.

4. *Stopword Removal*

Untuk menghapus kata-kata yang termasuk ke dalam stopwords, digunakan teknik pencocokan kata dengan kamus yang berisi daftar kata *stopwords* yang diambil dari situs http://www.ranks.nl/stopwords.

5. *Stemming*

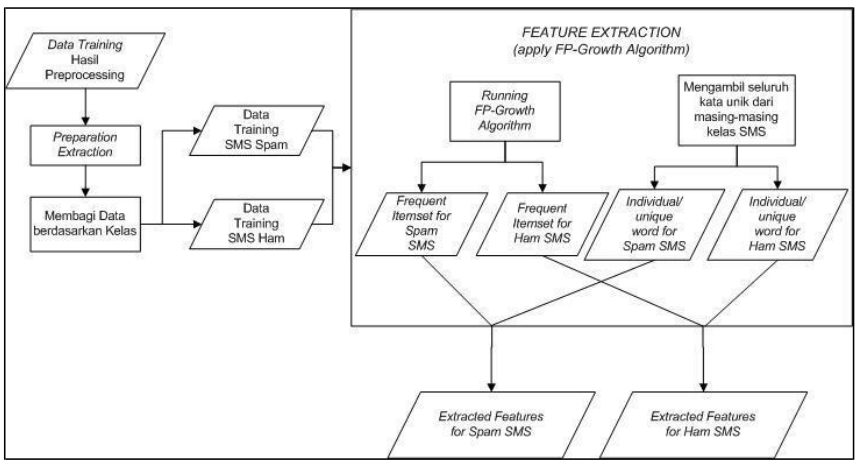
Pada dataset banyak terdapat kata yang memiliki imbuhan, sehingga dibutuhkan proses *stemming* untuk mengembalikan kata-kata tersebut ke dalam bentuk akarnya. Hal tersebut dimaksudkan agar dapat mengurangi variasi kata yang seharusnya memiliki arti sama namun memiliki bentuk imbuhan yang berbeda. Pada proses ini digunakan *Snowball Tartarus library* yang menerapkan algoritma *Porter Stemmer*, diambil dari situs *snowball.tartarus.org.*

6. *Handle Number*

Proses ini hanya menangani karakter angka berupa nomor telepon. Hal ini dilakukan, karena berdasarkan pengamatan, nomor telepon banyak muncul pada dataset SMS khususnya yang termasuk kelas spam, maka nomor telepon bisa menjadi fitur unik untuk klasifikasi teks SMS. Sehingga dibuat ketentuan untuk karakter angka dalam satu token dengan panjang ≥ 7 (panjang standar minimum nomor telepon), satu token karakter yang terdiri dari angka tersebut diubah dengan string "phonenumber" untuk menyamakan seluruh data nomor telepon dari kumpulan karakter angka menjadi satu kata yang sama. Apabila menemui karakter angka dalam satu token, namun panjang tidak memenuhi, maka karakter angka dalam satu token tersebut akan dihapus.

2.1.2 Feature Extraction

Pada *data training* dilakukan *feature extraction* dengan melibatkan algoritma *FP-Growth* untuk mendapatkan *frequent itemset.*



Gambar 2-2. proses feature extraction

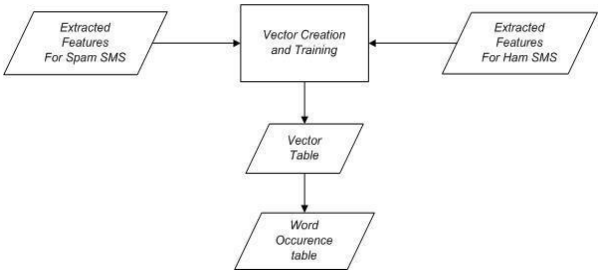
Penjelasan Gambar 2-2 adalah sebagai berikut.

1. Sebelum diproses, data isi SMS dalam format kata tersebut diubah ke dalam format angka melalui proses *preparation extraction*. Kemudian data dipisah ke dalam masing-masing kelas, sehingga didapatkan dua buah *file* inputan untuk proses selanjutnya.,

2. Kemudian dilakukan *running* FP-Growth dengan *minimum support* yang ditentukan pada pengujian,
3. Hasil proses FP-Growth berupa *frequent itemset* yang kemudian menjadi fitur baru untuk masing-masing kelas dalam proses klasifikasi,
4. Fitur baru tersebut digabung dengan fitur individu dari masing-masing kelas.

2.1.3 *Vector Creation and Training*

Pada proses ini dilakukan perhitungan untuk masing-masing kata yang telah diekstrak pada masing-masing kelas. Untuk mempermudah perhitungan, dibuat *vector table* lalu diubah ke bentuk *word occurrence table.*

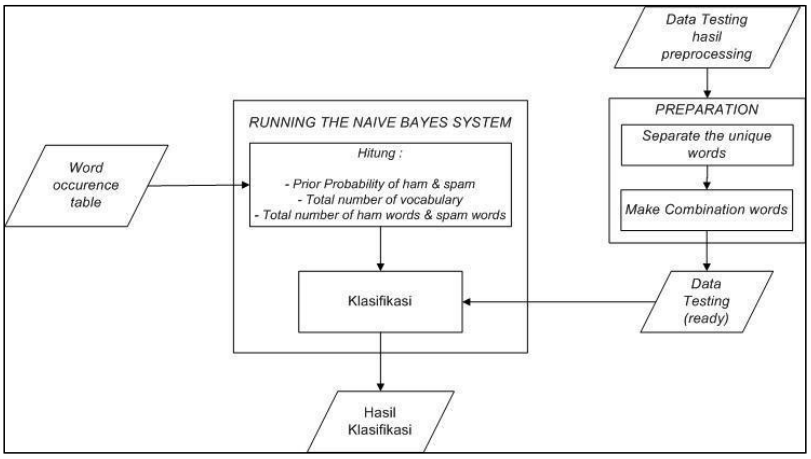


Gambar 2-3. proses vector creation and training

Vector table merupakan tabel yang berisi kemunculan fitur atau kata pada masing-masing kalimat sms. Sedangkan *word occurrence table* merupakan tabel yang berisi jumlah kemunculan seluruh kata yang terdapat pada masing-masing kelas.

2.1.3 Running The Naive Bayes System

Pada tahap ini, mulai dilakukan proses klasifikasi dengan Algoritma *Naive Bayes.* Kata-kata yang sudah dihitung pada *word occurence table* selanjutnya dilakukan perhitungan total dan *prior probability* terhadap masing-masing kelas (*spam* dan *ham*). Kemudian memasukkan data testing yang telah dilakukan proses *preparation* untuk dilakukan klasifikasi. Pada tahap klasifikasi, diterapkan perhitungan dengan *laplace estimator* atau *laplace smoothing* untuk menghindari nilai probabilitas 0. Gambar 2-4 merupakan gambaran proses klasifikasi dengan Naive Bayes.



Gambar 2-4.running naive bayes

2.1.3 Evaluasi (Testing)

Melalui proses evaluasi dapat diketahui apakah model yang telah didapatkan sudah layak untuk diimplementasikan atau tidak dengan menghitung nilai akurasinya. Maksud model disini ialah hasil yang didapat setelah melakukan proses *training* yaitu berupa nilai-nilai seperti *prior probability*, jumlah *vocabulary* yang didapat dan jumlah kata yang didapat pada masing-masing kelas. Jika hasil nilai akurasinya mencapai nilai yang tinggi maka model tersebut layak untuk digunakan dalam proses klasifikasi SMS baru.

Cara untuk mengukur kinerja dari suatu klasifikasi teks secara efektif terhadap suatu term yaitu dengan mengukur *recall (r)* dan *precision (p). Precision* adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Sedangkan *recall* adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi.

$$p = \frac{tp}{tp+fp} \qquad r = \frac{tp}{tp+fn}$$

$$F\ measure = \frac{2\ x\ p\ x\ r}{p+r}$$

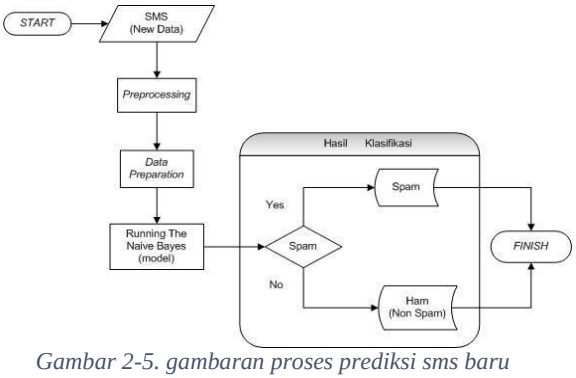
True positive dapat diartikan sebagai pesan yang *spam* yang dianggap sebagai pesan yang *spam*, *false positive* adalah pesan yang sah (*ham)* dianggap sebagai pesan *spam*, dan *false negative* merupakan pesan *spam* yang dianggap sebagai pesan *ham.*

Setelah kita mendapatkan definisi dari setiap *precision* dan *recall* maka kita dapat menghitung akurasinya. Akurasi didefinisikan

sebagai tingkat kedekatan antara nilai prediksi dengan nilai aktual, dirumuskan sebagai berikut :

$$Akurasi = \frac{tp+tn}{tp+tn+fp+fn}$$

Setelah didapatkan akurasi yang baik berdasarkan proses *training* dan *testing*, maka selanjutnya model (hasil perhitungan dari *word occurrence*) dapat digunakan untuk proses prediksi klasifikasi SMS baru. Pada proses prediksi ini data SMS baru tetap melewati tahap *preprocessing* dan *preparation* seperti yang diterapkan pada *data testing* untuk kemudian dilakukan klasifikasi seperti pada sub bab 2.1.3 untuk menentukan SMS tersebut termasuk *spam* atau *ham*.



Gambar 2-5. gambaran proses prediksi sms baru

3. Pengujian

3.1 Dataset

Untuk mengidentifikasi performansi dari sistem SMS *filtering* yang dibangun pada tugas akhir ini, digunakan dataset yang berasal dari SMS Corpus : *SMS Spam Collection v.1* dengan SMS berjumlah 5.574 SMS yang terdiri dari 4.827 SMS *ham* dan 747 SMS *spam*, serta dari *SMS Spam Corpus v.0.1 Big* dengan SMS berjumlah 1324 yang terdiri dari 1002 SMS *ham* dan 322 SMS *spam*.

3.2 Proses Pengujian

Pada pengujian ini dilakukan 6 skenario proses pengujian :

1. Pengujian Naive Bayes tanpa FP-growth menggunakan dataset *SMS Spam Corpus v.0.1 Big*
2. Pengujian Naive Bayes dengan FP-growth menggunakan dataset *SMS Spam Corpus v.0.1 Big*
3. Pengujian Naive Bayes tanpa FP-growth menggunakan dataset *SMS Spam Collection v.1*
4. Pengujian Naive Bayes dengan FP-growth menggunakan dataset *SMS Spam Collection v.1*
5. Pengujian Naive Bayes tanpa FP-growth menggunakan kedua dataset
6. Pengujian Naive Bayes dengan FP-growth menggunakan kedua dataset
7. Pengujian Naive Bayes dengan FP-growth berdasarkan karakteristik dataset

3.3 Hasil Pengujian

Berdasarkan proses pengujian ke-1 hingga ke-6 dapat dilakukan analisis terhadap penerapan *minimum support* yang optimal dan analisis perbandingan terhadap pengunaan Naive Bayes saja dengan penggunaan kolaborasi Naive Bayes dan FP-Growth.

Untuk proses pengujian ke-7 dapat dilakukan analisa terhadap karakteristik data yang cocok untuk diterapkan dengan metode FP-Growth.

3.3.1 Analisis *minimum support* pada FP-Growth

Penerapan nilai *minimum support* pada setiap dataset memiliki hasil yang berbeda. Dari hasil pengujian yang didapat, dianalisis hasil *minimum support* yang optimal pada masing- masing dataset, berikut disajikan gambar 3-1 untuk mempermudah dalam proses analisis.

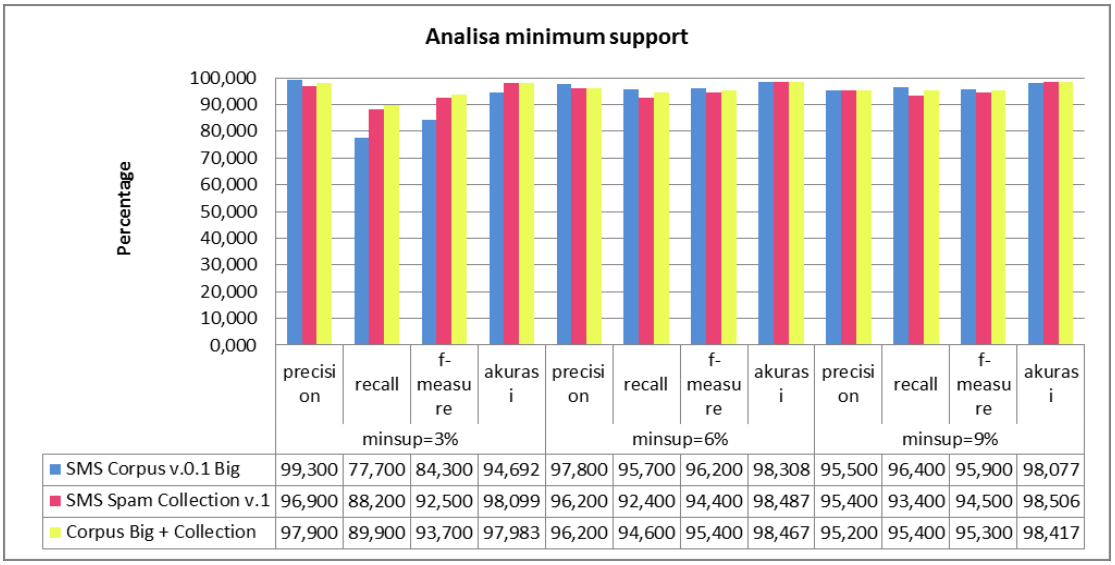
Dari gambar 3-1 terlihat pada SMS Corpus v.0.1 Big menghasilkan nilai *precision* tertinggi saat *minimum support 3%*. Hal tersebut dikarenakan banyaknya fitur baru yang dihasilkan khususnya untuk kelas *spam*, sehingga tingkat ketepatan sistem dalam memberikan jawaban terhadap informasi yang diminta penggunan menjadi lebih tinggi. Namun berbanding terbalik dengan hasil *recall*-nya yang menghasilkan nilai paling rendah. Karena berdasarkan pada tabel 4-5 saat menerapkan *minimum support* 3% untuk data SMS Corpus v.0.1 Big, dihasilkan fitur yang terlalu banyak pada sms *spam* hingga mencapai ribuan. Berdasarkan rumus peluang kata (rumus 3.1), bahwa jumlah kemunculan kata pada suatu kelas berbanding terbalik dengan hasil besar peluang kata pada kelas tersebut, jadi jika jumlah kemunculan kata dalam suatu kelas semakin besar maka peluang kata tersebut masuk ke kelas itu semakin kecil. Selain itu komposisi jumlah data *spam* lebih kecil kurang lebih 1/3 dari data *ham*, sehingga nilai *prior probability*-nya jelas lebih kecil, sehingga berdasarkan rumus (2.13) akan dihasilkan nilai *posterior probability* untuk diklasifikasikan k kelas *spam* pun menjadi lebih kecil, hal tersebut menyebabkan banyak sms *spam* yang salah terklasifikasi. Pada SMS Corpus v.01 Big, *minimum support* paling optimal diperoleh saat sebesar 6% dengan hasil akurasi 98,308%.

Pada penggunaan dataset *SMS Spam Collection* selalu menghasilkan nilai akurasi yang lebih tinggi dibandingkan dengan penggunaan dataset yang lainnya. Hal tersebut menunjukan bahwa ketiga parameter *minimum support* tersebut cocok digunakan pada dataset ini. Pada dataset

Spam Collection ini akurasi tertinggi didapat saat *minimum support* 9% yaitu sebesar 98,506%.

Pada penggunaan gabungan kedua dataset dihasilkan akurasi yang hampir sama dengan penggunaan dataset *SMS Spam Collection*, hal tersebut bisa disebabkan karena jumlah dataset yang digunakan pun memiliki kuantitas yang hampir sama sehingga saat diterapkan ketiga parameter *minimum support* tersebut tidak memiliki perbedaan yang signifikan. Pada dataset ini dihasilkan akurasi tertinggi saat *minimum support* 6% dengan akurasi sebesar 98,467%.

Kesimpulan dari pengujian dengan pengunaan ketiga dataset tersebut adalah kuantitas dan kualitas dataset berpengaruh dalam menghasilkan nilai yang baik. Kuantitas pengunaan dataset berbanding terbalik dengan kuantitas nilai parameter *minimum* support. Untuk data dengan kuantitas kecil seperti pada data SMS Corpus Big v.0.1 sebaiknya tidak menggunakan nilai *minimum support* yang terlalu kecil juga, karena akan menghasilkan fitur baru yang terlalu banyak dan tidak optimal. Untuk data yang sangat besar seperti gabungan kedua dataset tersebut, maka dibutuhkan nilai parameter yang semakin kecil agar didapatkan fitur baru yang optimal. Karena jika *minimum support* terlalu besar tidak akan menghasilkan fitur baru sama sekali dan hal itu tidak berdampak terhadap kenaikan nilai akurasi.

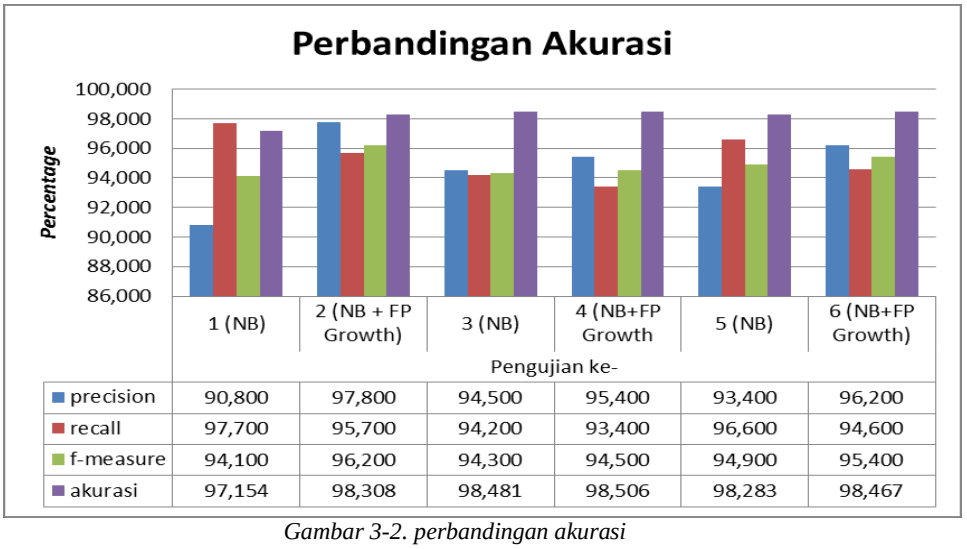


	minsup=3%				minsup=6%				minsup=9%			
	precision	recall	f-measure	akurasi	precision	recall	f-measure	akurasi	precision	recall	f-measure	akurasi
SMS Corpus v.0.1 Big	99,300	77,700	84,300	94,692	97,800	95,700	96,200	98,308	95,500	96,400	95,900	98,077
SMS Spam Collection v.1	96,900	88,200	92,500	98,099	96,200	92,400	94,400	98,487	95,400	93,400	94,500	98,506
Corpus Big + Collection	97,900	89,900	93,700	97,983	96,200	94,600	95,400	98,467	95,200	95,400	95,300	98,417

Gambar 3-1. analisa minimum support

3.3.2 Analisis perbandingan kedua metode

Nilai akurasi dari hasil pengujian pada kedua metode (dengan FP-Growth dan tanpa FP-Growth) akan dibandingkan dengan menggunakan nilai evaluasi rata-rata. Untuk metode dengan melibatkan FP-Growth, akan digunakan hasil nilai akurasi terbaik berdasarkan pengujian pada sub bab 3.3.1.

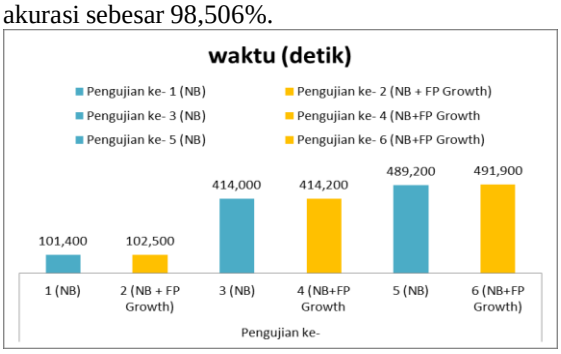


	1 (NB)	2 (NB + FP Growth)	3 (NB)	4 (NB+FP Growth	5 (NB)	6 (NB+FP Growth)
precision	90,800	97,800	94,500	95,400	93,400	96,200
recall	97,700	95,700	94,200	93,400	96,600	94,600
f-measure	94,100	96,200	94,300	94,500	94,900	95,400
akurasi	97,154	98,308	98,481	98,506	98,283	98,467

Gambar 3-2. perbandingan akurasi

Berdasarkan hasil pengujian yang ditampilkan pada gambar 3-2 terbukti bahwa dengan menerapkan metode FP-Growth untuk dikolaborasikan dengan Naive Bayes selalu menghasilkan nilai *f-measure* dan akurasi yang lebih tinggi, yang berarti sistem lebih tepat dalam melakukan klasifikasi. Selain itu, FP-Growth mampu meningkatkan nilai *precision* secara signifikan. Dengan demikian sistem semakin tepat dalam memberikan jawaban terhadap informasi yang diminta oleh pengguna. Berbanding terbalik dengan nilai *recall* lebih kecil, hal itu wajar karena perbandingan komposisi dokumen yang digunakan lebih besar *ham*. Namun, hal tersebut memiliki keuntungan lain karena jika terdapat sms yang memiliki fitur yang tidak diketahui sebelumnya berdasarkan *training*, maka kelas tersebut akan cenderung terklasifikasi menjadi *ham*, hal tersebut dapat melindungi sms tersebut tersaring menjadi *spam* jika ternyata sms tersebut merupakan sms penting.

Jadi berdasarkan pengujian, masing-masing pengunaan dataset menghasilkan akurasi yang lebih unggul dengan menerapkan FP-Growth, untuk SMS Corpus v.0.1 Big menaikan akurasi sebesar 1,154%, untuk SMS Spam Collection menaikan akurasi sebesar 0,025% dan untuk pengunaan kedua dataset menaikan akurasi sebesar 0,184%. Sedangkan akurasi tertinggi diperoleh pada dataset SMS Spam Collection v.1 dengan akurasi sebesar 98,506%.



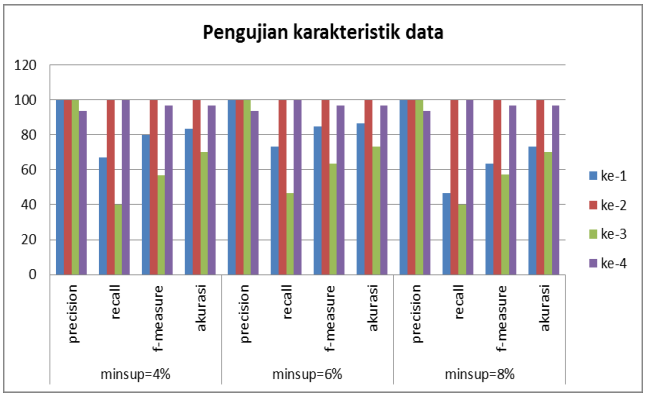
Gambar 3-3. perbandingan waktu

Dari segi waktu, dengan melibatkan algoritma FP-Growth jelas lebih lama bila dibandingkan dengan tanpa melibatkan FP-Growth. Hal ini dikarenakan FP-Growth membutuhkan waktu untuk membentuk fitur-fitur baru, sehingga fitur menjadi lebih banyak dan proses pengklasifikasian lebih lama karena data testing harus mencocokan fitur lebih banyak terhadap fitur pada data training.

Untuk penggunaan dataset SMS Spam Corpus v.0.1 Big memiliki selisih waktu 1,1 detik, penggunaan dataset SMS Spam Collection memmiliki selisih waktu 0,2 detik dan penggunaan gabungan kedua dataset memiliki selisih waktu 2,7 detik.

3.3.3 Analisis penggunaan data berdasarkan karakteristik

Pada pengujian karakteristik data ini digunakan dua buah tipe data. Tipe I adalah data yang memiliki pola serupa di setiap SMSnya, sedangkan untuk data tipe II adalah data yang memiliki pola lebih bervariasi di setiap SMSnya. Data yang digunakan hanya mengambil 100 SMS untuk *training* dengan 70 *ham* : 30 *spam* dan mengambil 30 SMS untuk *testing* dengan 15 *ham* : 15 *spam*. Hasil pengujian digambarkan pada gambar 3-4 berikut :



Gambar 3-4. hasil pengujian berdasarkan karakteristik data

Keterangan :
Pengujian ke-1 → Pola data training serupa setiap sms dan pola data testing serupa setiap sms
Pengujian ke-2 → Pola data training bervariasi setiap sms dan pola data testing serupa setiap sms
Pengujian ke-3 → Pola data training serupa setiap sms dan pola data testing bervariasi setiap sms
Pengujian ke-4 → Pola data training bervariasi setiap sms dan pola data testing serupa setiap sms

Berdasarkan hasil yang diperoleh, dapat dilihat bahwa pada pengujian ke-1 dan ke-3 dengan menggunakan data training yang mirip di setiap SMSnya menghasilkan fitur baru yang terlalu banyak. Hal tersebut mengakibatkan nilai *recall* menjadi sangat kecil dan menghasilkan nilai akurasi yang tidak terlalu tinggi. Sebaliknya saat digunakan data *training* yang lebih bervariasi, menghasilkan banyak fitur baru yang lebih optimal. Hal tersebut dapat membantu untuk lebih mendeteksi sms-sms dengan pola-pola kalimat yang serupa. Seperti ditunjukan pada pengujian ke-2 dengan penggunaan data *training* yang lebih bervariasi dan data *testing* yang memiliki kemiripan di setiap sms-nya menghasilkan akurasi yang sangat tinggi mencapai 100%.

4. Kesimpulan

Berdasarkan analisa terhadap pengujian yang dilakukan dalam tugas akhir ini, dapat disimpulkan bahwa :

1. Dari kedua metode yang digunakan, performansi kedua metode sama baiknya untuk pengklasifikasian sms dengan akurasi rata-rata diatas 90%. Pengunaan metode kolaborasi Naive Bayes dan FP-Growth lebih unggul dengan rata-rata akurasi untuk masing-masing dataset pada SMS Spam Corpus v.0.1 Big unggul sebesar 1,154%, pada SMS Spam Collection unggul sebesar 0,025% dan gabungan kedua dataset unggul sebesar 0,184%.
2. Akurasi rata-rata terbaik didapat saat digunakan dataset SMS Spam Collection v.1 dengan *minimum support* 9% saat implementasi dengan FP-Growth dengan akurasi sebesar 98,506%.
3. Penerapan *minimum support* membantu menangani masalah fitur yang terbatas dikarenakan jumlah karakter yang terbatas pada SMS, sehingga dihasilkan fitur baru yang dapat membedakan antara sms *spam* dan sms *ham.*
4. Penggunaan dataset dengan data *training* yang bervariasi cocok untuk diterapkan menggunakan metode FP-Growth,
5. Pemberian nilai parameter *minimum support* berbanding terbalik dengan kuantitas dataset. Untuk data yang semakin besar digunakan *minsup* yang semakin kecil agar didapat fitur baru yang optimal (jika *minsup* terlalu besar tidak menghasilkan fitur baru), sedangkan untuk dataset yang semakin kecil digunakan *minsup* yang lebih besar (jika *minsup* terlalu kecil didapat fitur baru yang terlalu banyak dan tidak efektif).
6. Dengan menerapkan FP-Growth untuk *feature extraction* dapat meningkatkan nilai *precision*, sehingga sistem lebih tepat dalam memberikan jawaban terhadap informasi yang diminta pengguna untuk klasifikasi sms.

Daftar Pustaka

[1] A.W, E., Mardiani, & Tinaliah. (2013). Penerapan Metode Naive Bayes Untuk Sistem Klasifikasi SMS Pada Smartphone Android.

[2] Ahmed, l. (2014). SMS Clasiification Based on Naive Bayes Classifier and Apriori Algorithm Frequent Itemset. *International Journal of Machine Learning and Computing*, vol.4, 183-187.

[3] Han. (2013). *Data Mining : Concept and Technique 3.* Waltham: Morgan Kaufmann Publishers.

[4] Han, J., & Pei Jian, Y. (2000). Mining Frequent Pattern without Candidate Generation. *ACM SIGMOD Confrence Proceedings.*

[5] Makhtidi, K. (2012). *Sistem Spam Detector Untuk SMS Berbahasa Indonesia pada Smartphone Android.* Bogor: Departemen Ilmu Komputer,Fakultas matematika dan Ilmu Pengetahuan Alam Institut Pertanian Bogor.

[6] Motoda, H., & Liu, H. (t.thn.). Feature Selection, Extraction and Construction.

[7] Porter, M. (1980). An Algorithm for Suffix Stripping. *Program : electronic library and information systems*, 14(3) 130-137.

[8] Raschka, S. (October 14, 2014). Naive Bayes and Text Classification I Introduction and Theory.

[9] Samuel, D. (t.thn.). Penerapan Struktur FP-Tree dan Algoritma FP-Growth dalam Optimasi Penentuan Frequent Itemset.

[10] Shirani-Mehr, H. (t.thn.). SMS Spam Detetection using Machine Learning Approach.

[11] Sunardi, M., & Listiyono, H. (2009). Aplikasi SMS Gateway. *Jurnal Teknologi Informasi DINAMIK*, XIV(1).

[12] *The Porter Stemming Algorithm*. (t.thn.). Dipetik April 23, 2015, dari Snowball Tartarus: http:// snowball.tartarus.org/algorithms/porter/stemmer.html

[13] W.Qian, H. (Sept.2009). Studying of Classifying Junk Message Based on The Data Mining. *Proc.International Conference on Management and Service Science* (hal. 1-4). IEEE Press.