IMPLEMENTASI ALGORITMA NAIVE BAYES PADA APLIKASI ANTI SMS SPAM

Oleh:
Karimul Makhtidi

Pembimbing
Hari Agung Adrianto, S.Si, M.Si.

PENDAHULUAN

Latar Belakang

Sekarang ini, *Short Message Service* atau SMS merupakan media komunikasi yang paling sering digunakan dalam berkomunikasi. SMS seakan sudah melekat dengan aktivitas keseharian masyarakat. Selain mudah dan cepat, tarif yang semakin murah membuat masyarakat lebih gemar menggunakan SMS dalam berkomunikasi. Tidak hanya untuk berkomunikasi, SMS juga digunakan sebagai layanan publik seperti SMS banking, yaitu layanan yang dapat mempermudah masyarakat dalam melakukan berbagai aktivitas perbankan hanya dengan menggunakan SMS.

Penggunaan SMS yang semakin luas ini dimanfaatkan oleh sebagian orang yang tidak bertanggung jawab untuk melakukan tindak kejahatan, seperti penipuan melalui SMS. Hanya dengan bermodal handphone dan pulsa, penipu dapat melakukan aksinya dengan mudah dan merasa lebih aman karena akan sulit dilacak. Maraknya aksi penipuan dewasa ini membuat masyarakat semakin resah. Apalagi SMS yang mengandung unsur penipuan dan yang tidak mengandung unsur penipuan sulit untuk dibedakan oleh masyarakat awam. Bentuk tindak penipuan melalui SMS ini sangat beragam. Salah satu yang saat ini marak di masyarakat adalah SMS yang berisi permintaan pulsa telepon seluler ke nomor tertentu, dengan mengatasnamakan orangtua, yaitu mama atau papa.

SMS penipuan yang beredar luas di masyarakat sebenarnya memiliki pola tertentu. Hanya saja masyarakat tidak banyak mengetahui sehingga tertipu oleh SMS tersebut. Dengan mengenali pola ini, masyarakat dapat lebih berhati-hati dalam menindaklanjuti SMS yang diterima sehingga kejahatan melalui SMS ini dapat dihindari.

Berbagai upaya telah dilakukan untuk meminimalisir dampak dari maraknya sms penipuan yang menggunakan berbagai macam modus. Salah satunya yang sedang hangat adalah upaya pemerintah menerapkan SMS interkoneksi berbasis biaya antar operator. Namun hasil dari kebijakan ini belum bisa di nilai karena masih relatif baru untuk sebuah kebijakan.

SMS yang memiliki indikasi tindak penipuan ini termasuk ke dalam SMS spam**.** Penelitian Pritasari Palupiningsih (2011) mencoba menganalisis pola SMS yang berindikasi adanya tindak penipuan dan menggunakan pola SMS tersebut sebagai prediksi terhadap suatu SMS, untuk menentukan ada atau tidaknya indikasi tindak penipuan. Penelitian tersebut telah menghasilkan model klasifikasi yang dapat digunakan untuk memprediksi sebuah SMS memiliki indikasi tindak penipuan atau tidak. Prediksi dilakukan dengan menggunakan fitur-fitur yang ada dalam SMS tersebut. Selain itu, melalui penelitiannya juga dibandingkan kinerja algoritme Naive Bayes dan C4.5 ter-hadap dataset yang digunakan. Dihasilkan bahwa kinerja algoritme Naive Bayes lebih baik daripada algoritme C4.5.

Penelitian ini merupakan penelitian lanjutan dari penelitian Pritasari Palupiningsih (2011) dengan mengimplementasikan model klasifikasi yang telah dihasilkan dan juga melanjutkan pengembangan aplikasi Anti sms spam yang dibuat oleh Candra Wangsa S dkk (projek mata kuliah TKI). Model klasifikasi ini diimplementasikan pada *platform mobile* android sehingga memberikan gambaran kepada pengguna mengenai SMS yang

memiliki indikasi tindak penipuan dan yang tidak.

Tujuan Penelitian

Tujuan penelitian ini adalah mengimplementasikan sistem anti SMS spam pada *platform mobile* android.

Ruang Lingkup Penelitian

Ruang lingkup penelitian yang dilakukan adalah menggunakan model klasifikasi untuk data SMS yang beredar di masyarakat pada tahun 2010 – 2012. Indikasi tindak penipuan yang akan dicari polanya adalah SMS yang berisi permintaan pulsa telepon seluler ke nomor tertentu, dengan mengatasnamakan orangtua, yaitu mama atau papa. Kemudian SMS yang berisi penawaran menjadi agen pulsa, undian berhadiah, keikutsertaan seminar dari lembaga tertentu, permintaan transfer sejumlah uang ke rekening tertentu dan SMS yang menunjukkan ketertarikan pada proses jual beli tanah, rumah, atau mobil.

Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan gambaran yang lebih jelas mengenai SMS yang memiliki indikasi tindak penipuan.

TINJAUAN PUSTAKA

Spam SMS

Spam SMS adalah penyalahgunaan dalam pengiriman pesan melalui *Short Message Service* (SMS) untuk menampilkan berita iklan atau keperluan lainnya yang mengakibatkan ketidaknyamanan pengguna ponsel. Contoh dari Spam SMS adalah SMS penipuan, SMS ancaman, SMS promosi, dan lain-lain.

Klasifikasi

Klasifikasi yaitu proses untuk menemukan sekumpulan model atau fungsi yang mendeskripsikan dan membedakan kelas-kelas data dengan tujuan untuk memprediksikan kelas dari objek yang belum diketahui kelasnya (supervised learning) dengan karakteristik tipe data yang bersifat kategorik (Han & Kamber 2001). Proses klasifikasi dibagi menjadi dua fase, yaitu learning dan test (Han & Kamber 2001). Pada fase learning, sebagian data yang telah diketahui kelas datanya (training set) digunakan untuk membentuk model. Selanjutnya pada fase test, model yang sudah terbentuk diuji dengan sebagian data lainnya untuk mengetahui akurasi model tersebut. Jika akurasinya mencukupi maka model tersebut dapat dipakai untuk memprediksi kelas data yang belum diketahui.

Naive Bayes

Naive Bayes adalah metode Bayesian Learning yang paling cepat dan sederhana. Hal ini didasarkan pada teorema Bayes dan hipotesis kebebasan, menghasilkan *classifier* statistik berdasarkan peluang. Ini adalah teknik sederhana, dan harus digunakan sebelum mencoba metode yang lebih kompleks. *Classifier* Naive Bayes adalah algoritma pembelajaran peluang yang berasal dari teori keputusan Bayesian. Peluang sebuah pesan *d* berada di kelas *c*, $P(c|d)$, dihitung sebagai

$$P(c|d)\alpha P(c)\prod_{k=1}^{m} P(t_k|c), \quad (1)$$

dimana $P(t_k|c)$ adalah peluang bersyarat dari fitur t_k yang terjadi dalam pesan kelas c dan $P(c)$ adalah peluang dari pesan sebelumnya yang terjadi di kelas *c*. $P(t_k|c)$ dapat digunakan untuk mengukur berapa banyak bukti kontribusi t_k bahwa *c* adalah kelas yang benar. Dalam klasifikasi email, kelas dari pesan ditentukan dengan menemukan *maximum a posteriori* (MAP) kelas c_{map} paling mungkin yang didefinisikan oleh

$$c_{map} = \arg max_{c\in\{c_1,c_s\}} P(c|d)$$
$$= \arg max_{c\in\{c_1,c_s\}} P(c)\prod_{k=1}^{m} P(t_k|c) \quad (2)$$

Karena Formula 2. melibatkan perkalian banyak peluang bersyarat, satu untuk masing-masing fitur, dapat mengakibatkan perhitungan berada di *underflow floating point*. Dalam prakteknya, perkalian peluang sering dikonversi untuk sebuah tambahan logaritma probabilitas dan, oleh karena itu, untuk memaksimalkan persamaan adalah alternatif yang ditunjukkan oleh

$$c_{map} = \arg max_{c\in\{c_1,c_s\}} [\log P(c) + \sum_{k=1}^{m} \log P(t_k, c)] \quad (3)$$

Semua parameter model, yaitu distribusi peluang kelas dan fitur, dapat diperkirakan dengan frekuensi relatif dari data latih *D*. Perhatikan bahwa ketika kelas dan fitur pesan yang diberikan tidak terjadi bersama-sama di data latih, estimasi probabilitas berbasis frekuensi yang sesuai akan menjadi nol, yang akan membuat sisi kanan Formula

3. tidak terdefinisi. Masalah ini dapat diatasi dengan memasukkan beberapa koreksi seperti *Laplace smoothing* di semua probabilitas perkiraan.

Evaluasi Klasifikasi SMS

Cara untuk mengukur kinerja dari suatu klasifier teks secara efektif terhadap suatu *term* yaitu dengan mengukur *recall (r)* dan *precision (p)*. Perhitungan *recall* dan *precision* diformulasikan berdasarkan tabel 1 berikut ini

Table 1 Confusion Matrix

	Relevant	Not relevant
Retrieved	Tp	Fp
Not retrieved	Fn	Tn

dengan demikian, *recall* dan *precision* dapat didefinisikan sebagai

$$p = \frac{tp}{tp + fp}, \quad r = \frac{tp}{tp + fn}.$$

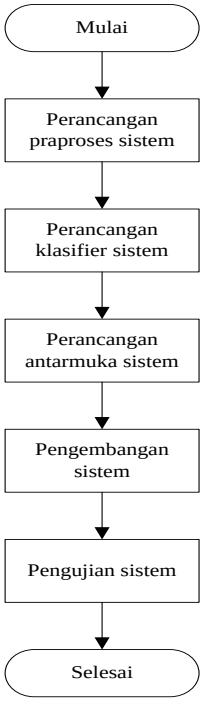
Precision merupakan rasio dokumen yang di-retrieve adalah relevan, sedangkan *recall* merupakan rasio dokumen relevan yang di-retrieve.

METODE PENELITIAN

Penelitian ini akan dilakukan melalui beberapa tahap. Tahapan tersebut disesuaikan dengan metode penelitian yang dapat dilihat pada Gambar 1.

Perancangan Praprosess Sistem

Terdapat tiga langkah yang dilakukan pada tahap praproses, yaitu *tokenizing*, *filtering*, dan *stemming*. Praproses dokumen dilakukan dengan memisahkan mereka menjadi token kata yang berisi huruf (penggunaan angka dan tanda baca tidak termasuk dalam bahasa), menambahkan masing-masing token dengan spasi, dan menghasilkan semua kemungkinan n-gram dengan panjang 1 sampai 5 untuk setiap kata dalam dokumen. Pada lagkah *tokenizing*, dilakukan pemotongan setiap kata yang terdapat dalam SMS. Karakter selain huruf yang terdapat dalam SMS akan dihilangkan. Setiap kata disebut sebagai token. Langkah selanjutnya adalah *filtering*, dimana pada langkah ini dilakukan pembuangan *stopword*. Yang termasuk ke dalam *stopword* adalah yang, di, ke, dari, adalah, dan, atau, dan lain sebagainya. Setelah itu dilakukan langkah *stemming*, yaitu menghilangkan imbuhan atau akhiran yang terdapat pada token. Pada tahap ini akan dibuat rancangan praproses sistem sehingga tahap praproses dapat dilakukan secara otomatis oleh sistem.



Gambar 1 Metodologi Penelitian

Perancangan Klasifier Sistem

Klasifikasi merupakan bagian terpenting dalam sistem ini. Dalam tahap ini akan dibuat sistem klasifikasi menggunakan algoritma Naïve Bayes. Dengan demikian sistem akan mampu menyaring SMS yang masuk dan kemudian memasukkan SMS tersebut ke dalam kelas spam atau tidak.

Perancangan Antarmuka Sistem

Perancangan antarmuka dilakukan dengan merancang tampilan pada platform android dengan kombinasi warna, teks, dan gambar sesuai dengan isi dan tujuan pengembangan sistem. Selain itu dalam tahap ini sistem akan dirancang pula bagian

interaksi manusianya sehingga dapat mudah dalam penggunaanya (*user friendly*).

Pengembangan Sistem

Tahap pengembangan sistem ini dimaksudkan untuk menggabungkan, mengimplementasikan, dan mengintegrasikan tahapan sebelumnya yang sudah dilakukan supaya menjadi satu kesatuan sistem yang siap digunakan. Pemilihan *framework* yang tepat juga dilakukan pada tahap ini, sehingga aplikasi yang dibangun memiliki kerangka dan aturan dalam penulisan kode sehingga kode menjadi terstruktur dan memudahkan untuk dikembangkan pada penelitian selanjutnya.

Pengujian Sistem

Pengujian terhadap sistem dilakukan dengan mengevaluasi *precision* dan *recall* dari kinerja klasifikasi yang dihasilkan. Dengan menggunakan precision dan recall ini, dapat dilakukan akurasi dari hasil klasifikasi sehingga dapat diketahui efektivitas kinerja dari algoritma Naive Bayes.

Lingkungan Implementasi

Lingkungan implementasi yang digunakan adalah sebagai berikut:

Perangkat lunak:

- Microsoft Windows 7 Professional
- Android SDK
- Eclipse

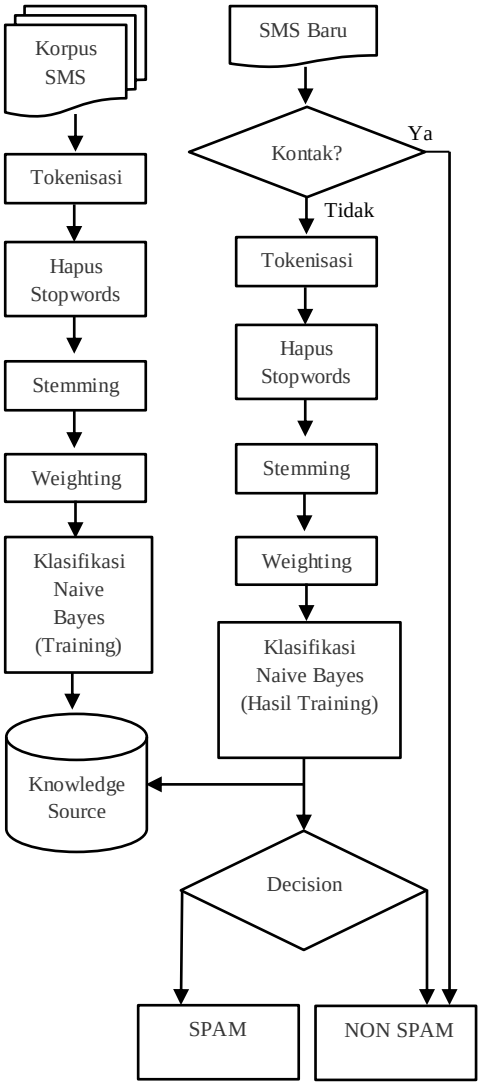
Perangkat keras:

- Processor intel Core i7 @1.73 GHz
- RAM 4 GB
- Hardisk 500GB
- Sony Ericsson Xperia Neo V

HASIL DAN PEMBAHASAN

Deskripsi Singkat Sistem

Sistem Anti SMS Spam ini terbagi menjadi dua alur. Yang pertama adalah alur *offline* dan yang kedua adalah alur *online*. Yang dimaksud alur *offline* adalah tahapan proses klasifikasi yang dilakukan diluar perangkat *mobile* android sedangkan alur *online* itu dilakukan diperangkat *mobile* atau emulator android. Alur kerja sistem dapat dilihat pada Gambar 2 berikut ini



Gambar 2 Alur Sistem

Perancangan Praproses Sistem

Kelas yang dipakai dalam pengklasifikasian SMS ini ada dua macam, yaitu kelas spam dan kelas non spam. Tahapan dalam klasifikasi terdiri dari proses pengolahan data training dan proses pengolahan data testing. Data training yang digunakan sebanyak 148 dokumen SMS dengan sms spam sebanyak 93 dokumen dan sms non spam sebanyak 55 dokumen. Seluruh dokumen sms menggunakan bahasa indonesia, baik yang baku maupun yang tidak baku. Untuk

memproses seluruh dokumen digunakan program dengan menggunakan bahasa pemrograman perl.

Pada tahap praproses data yang pertama kali dilakukan adalah mengubah dokumen sms (korpus) yang telah dikumpulkan ke dalam bentuk dokumen XML. Seluruh dokumen XML ini memiliki elemen root <SMS>, dengan elemen child <CLASS>, <ID>, dan <CONTENT>. Dokumen XML ini kemudian digunakan untuk praproses data yang meliputi 3 tahap, yaitu tokenisasi, hapus *stopword*, dan stemming.

Tokenisasi

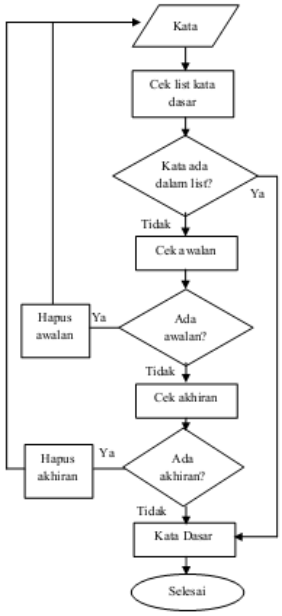
Seluruh korpus masuk ke dalam proses tokenisaasi untuk diambil token atau kata-kata yaitu kata yang dibatasi oleh *whitespace* dan semua karakter kecuali huruf dan angka, seperti: , . ? ! / ' " : ; & ( ) + - *. Selain itu dalam proses tokenisasi ini, kata-kata akan di ubah dalam bentuk *lowercase* untuk menyeragamkan kata.

Hapus Stopword

Penghapusan *stopword* dilakukan pada kata yang berfungsi sebagai kata depan dan singkatan dari kata depan itu sendiri. *Stopword* yang digunakan pada sistem ini dibatasi untuk kata-kata: di/d, ke/k, dari/dr, pada/pd, dan/n, atau/or, yang/yg, dengan/dgn, ini/ni/ne, itu/tu, juga/jg, untuk/utk/u, akan/kan/kn dan tetapi/tapi/tp.

Stemming

Pada tahap stemming dilakukan penghapusan prefiks dan sufiks pada kata yang mengandung imbuhan untuk mendapatkan kata dasar. Penentuan kata dasar juga mengacu kepada kamus kata dasar yang diperoleh dari KBBI (Kamus Besar Bahasa Indonesia) yang jumlah keseluruhannya 28.526 kata. Algoritma stemming yang digunakan seperti terlihat pada Gambar 3.



Gambar 3 Algoritma stemming.

Perancangan Klasifier Sistem

Setelah dilakukan praproses data diperoleh token dari setiap dokumen. Tahap berikutnya adalah pembobotan, pembobotan yang digunakan adalah pembobotan *term frequency* atau *tf*. Pembobotan *tf* digunakan karena pada klasifikasi Naïve Bayes hanya dibutuhkan frekuensi dari kata-kata untuk mencari peluang Bayesnya.

Dalam proses training ini dibutuhkan informasi frekuensi tiap kata unik untuk masing-masing kelas, frekuensi kata di setiap kelas dan jumlah kata unik di semua kelas. Proses training menghasilkan peluang masing-masing kelas yang dihitung dengan menggunakan persamaan (4) dan peluang setiap kata unik menjadi anggota suatu kelas yang dihitung dengan menggunakan persamaan (5).

$$P(v_j) = \frac{|docs_j|}{|contoh|} \quad (4)$$

$$P(w_k | v_j) = \frac{n_k + 1}{n + |kosakata|} \quad (5)$$

dengan P(vj) dan probabilitas kata wk untuk setiap kategori $P(w_k|v_j)$ dihitung pada saat pelatihan. $|docs_j|$ adalah jumlah kata pada kategori j dan |*contoh*| adalah jumlah dokumen yang digunakan dalam pelatihan. Sedangkan n_k adalah jumlah kemunculan kata w_k pada kategori v_j , n adalah jumlah semua kata pada kategori v_j dan |*kosakata*| adalah jumlah kata yang unik (*distinc*) pada semua data latihan.

Secara keseluruhan tahapaan klasifikasi pada tahap training ini dapat dilakukan sebagai berikut :

1. Dari korpus, ambil tiap sms => (isisms, kelassms)
2. Untuk tiap sms:
   - (jumlah sms di korpus)++
   - beri ID setiap sms
   - simpan kelas sms => spam atau bukan
   - melakukan tokenisasi isisms dengan pemisah => ,.?!/:;&%()=-'"
   - simpan tiap pisahan kata yang diperoleh, untuk tiap kata:
     - ubah menjadi huruf kecil
     - jadikan NULL jika merupakan salah satu dari stopword => Regex
     - *stemming*
     - jika panjang string lebih dari 0, maka simpan kata ini (IDsms, kata, frekuensi++) => matrix term frekuensi => termfrekuensi kata k di s => termfrekuensi[s , k]
3. Hitung jumlah seluruh kata dalam korpus:
   - untuk setiap sms s:
     - jika kelas s adalah spam maka (jumlah sms di kelas spam)++
   - untuk setiap kata k:
     - jumlah kata di korpus += jumlah kata k
     - jika kelas s adalah spam:
       - jumlah kata k di kelas spam += termfrekuensi[s,k]
       - jumlah kata k di kelas bukan spam += 0
       - jumlah token di kelas spam += termfrekuensi[s,k]
     - selainnya
       - jumlah kata k di kelas spam += termfrekuensi[s,k]
       - jumlah kata k di kelas bukan spam += 0
4. Hitung peluang setiap kata untuk setiap kelas:
   - untuk setiap kata k:
     - peluang kata k di kelas spam = ((jumlah kata k di kelas spam) + 1) / ((jumlah token di kelas spam) + (jumlah kata di korpus))
     - peluang kata k di kelas bukan spam = ((jumlah kata k di kelas bukan spam) + 1) / ((jumlah kata di korpus) - (jumlah token di kelas spam) + (jumlah kata di korpus))
5. Testing untuk setiap sms di korpus (hitung peluang setiap SMS untuk setiap kelas):
   - untuk setiap sms s:
     - peluang sms s di kelas spam = (jumlah sms di kelas spam) / (jumlah sms di korpus)
     - peluang sms s di kelas bukan spam = 1 - (peluang sms s di kelas spam)
     - untuk setiap kata k:
       - peluang sms s di kelas spam = (peluang sms s di kelas spam) * ((peluang kata k di kelas spam) ^ (termfrekuensi[s,k]))
       - peluang sms s di kelas bukan spam = (peluang sms s di kelas bukan spam) * ((peluang kata k di kelas bukan spam) ^ (termfrekuensi[s,k]))
     - jika (peluang sms s di kelas spam) > (peluang sms s di kelas bukan spam) maka kelas sms adalah SPAM
     - selainnya kelas sms adalah BUKAN SPAM

Pada penelitian ini diperoleh 2432 kata unik beserta peluangnya dengan rincian 1216 untuk kelas spam dan 1216 untuk kelas

non spam. Hasil dari persamaan 4 dan persamaan 5 ini kemudian digunakan untuk melakukan testing dari dokumen yang ada. Dari 55 dokumen yang termasuk ke dalam kelas non spam didapat 55 dokumen yang termasuk dalam kelas non spam sedangkan dari 93 dokumen dari kelas spam didapat 93 dokumen yang termasuk ke dalam kelas spam. Dengan demikian akurasi klasifikasi mencapai 100%.

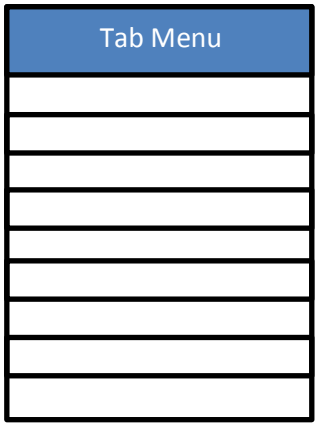
Dari hasil tersebut, peluang dari tiap-tiap kata unik di masing-masing kelas akan digunakan sebagai *knowledge source* dan hasil klasifikasi dari masing-masing sms yang ada di dalam korpus digunakan sebagai pembanding pada tahap implementasi sistem di android.

Perancangan Antarmuka Sistem

Antarmuka aplikasi pada perangkat mobile sangat berbeda dengan antarmuka pada aplikasi desktop maupun web. Antarmuka pada perangkat mobile harus didesain sangat sederhana namun juga tidak tidak mengurangi kenyamanan pengguna. Masing-masing *operating sistem* di perangkat *mobile* memiliki ciri khas yang berbeda, terutama android. Pada penelitian ini, antarmuka sistem mencoba mengikuti guidelines UI dari android dan secara garis besar terdiri atas menu dan konten. Untuk menu pada sistem ini menggunakan *tab menu* sedangkan tampilan konten menggunakan *listview*. Menu yang ada dalam aplikasi ini diantaranya *logs*, *allow*, *block*, dan *options*. Menu *logs* untuk menampilkan sms yang di blok, menu *allow* untuk menampilkan *whitelist number* dan menu *block* untuk menampilkan *blacklist number*. Rancangan tampilan dari sistem ini dapat dilihat pada Gambar 3.

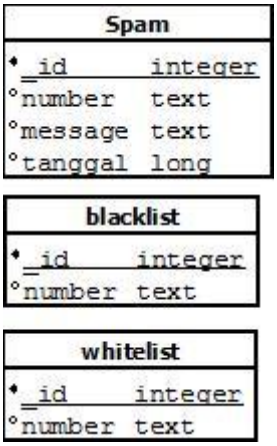
Pengembangan Sistem

Pada tahap pengembangan sistem ini, seluruh tahapan dari mulai tokenisasi, buang stopwords, dan stemming diadopsi dan diimplementasikan menggunakan bahasa pemrograman android. Untuk setiap sms yang masuk akan ditangkap dan dilakukan tahap yang sama sampai didapatkan kata-kata unik. Kemudian kata unik tersebut akan dicocokan dengan *knowledge source* dan dihitung peluangnya untuk masing-masing kelas. Nilai peluang yang lebih besar akan dimasukkan kedalam kelas tersebut. *Knowledge source* ini direpresentasikan dalam bentuk Hashmap sehingga mudah dalam mendapatkan peluang dari suatu kata unik.



Gambar 3 Rancangan Antarmuka Aplikasi

Setiap SMS yang dideteksi sebagai spam akan disimpan dalam database SQLite. Selain tabel spam ada pula tabel blacklist dan whitelist untuk menyimpan nomor yang di blacklist dan nomor whitelist. Struktur dari masing-masing tabel dapat dilihat pada Gambar berikut:



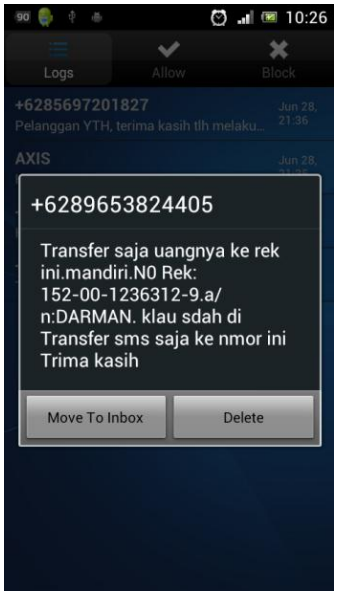
Gambar 4 Struktur Tabel

Dalam implementasinya, aplikasi Anti Sms Spam ini tidak hanya bisa untuk mendeteksi sms spam secara otomatis, namun juga ada beberapa fitur tambahan,

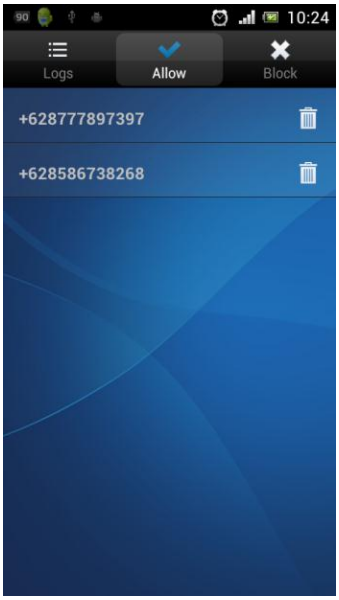
yaitu notifikasi, *review* dan *restore blocked* SMS, menghapus *blocked* SMS, *shake gesture* untuk menghapus seluruh *blocked* SMS, menambahkan dan menghapus *blacklist* dan *whitelist* number.

Berikut hasil tampilan Aplikasi Anti SMS Spam :

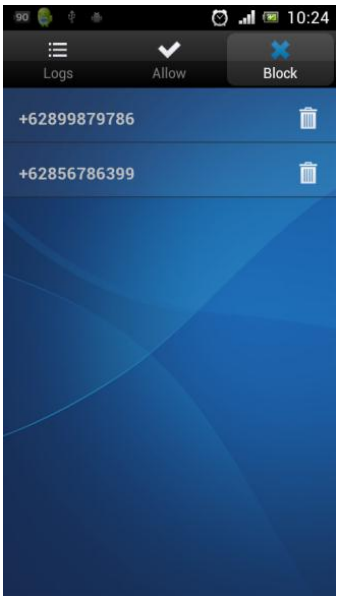
Gambar 5 Tampilan Menu *Logs*



Gambar 6 Tampilan Review *Blocked* SMS



Gambar 7 Tampilan Menu *Allow*



Gambar 8 Tampilan Menu *Block*

Pengujian Sistem

Pada pengujian sistem, dilakukan testing sebanyak 60 SMS. Pengujian ini dilakukan di emulator dan device android karena emulator hanya berguna untuk menguji fungsionalitas, bukan performance di device yang sebenarnya. Pengujian ini dilakukan juga evaluasi kinerja dari klasifikasi sistem. Dari 60 SMS tersebut diperoleh :

	Spam	Non Spam
Spam	30	0
Non Spam	0	30

Nilai akurasi : 60/60 x 100% = 100%
Precision = 30/(30+0) x 100% = 100 %
Recall = 30/(30+0) x 100% = 100 %

Berdasarkan hasil diatas dapat disimpulkan bahwa kinerja klasifikasi pada sistem ini sangat bagus karena akurasi, *precison*, dan *recall*-nya mencapai 100%. Selain itu performance di device android juga menunjukkan kinerja yang cukup bagus dengan waktu relatif singkat (1-2 detik) dapat mengklasifikasikan SMS yang masuk.

KESIMPULAN DAN SARAN

Kesimpulan

Hasil penelitian ini menunjukkan bahwa:

1. Telah diimplementasikan algoritma naive bayes pada sistem sms spam *detection* pada platform *mobile* android.
2. Telah dihasilkan sistem deteksi sms spam pada platform *mobile* android untuk bahasa indonesia baik yang baku maupun tidak baku.
3. Kinerja klasifikasi pada sistem sangat baik dengan akurasi, *precision*, dan *recall* mencapai 100% dan performance di device android yang cukup bagus(1-2 detik).

Saran

Saran untuk penelitian selanjutnya :

1. Dapat menambahkan fitur-fitur baru yang dapat mempermudah pengguna dalam menggunakan aplikasi ini, misalnya menambahkan *blacklist* dari sms atau kontak.
2. Sistem sms spam detection ini dapat di terapkan pada platform mobile yang lain seperti blackberry, windows phone, dan IOS.
3. Menggunakan algortima klasifikasi yang lain seperti SVM atau Hidden Markov Model yang secara teori memiliki kinerja yang lebih bagus dari naive bayes.

DAFTAR PUSTAKA

F. Sebastiani, “Machine learning in automated text categorization,” in ACM Computing Surveys 34, 1, 2002.

G. Cormack, “Feature engineering for mobile (sms) spam filtering,” in In 30th ACM SIGIR Conference on Research and Development on Information Retrieval, 2007.

e. a. Cormack, G. V., “Spam filtering for short messages,” inProceedings of the sixteenth ACM conference on Conference on information and knowledge management, 2007.

G. Odón, D. & Bringas, “Content based sms spam filtering,” in In Proceedings of the 2006 ACM symposium on Document engineering, 2006.

E. Jiang, “Content-based spam email classification using machine-learning algorithms.” John Wiley & Sons, 2010.

I. H. Witten, “Text mining,” inin a Digital Library. International Journal on Digital Libraries archive, 2010.

Y. Yang and J. Pedersen, “A comparative study on feature selection in text categorization.” Morgan Kaufmann Publishers, 1997.

T. Joachims, “Transductive inference for text classification using support vector machines,” in International Conference on Machine Learning (ICML), 1999.