Jurnal Sains, Teknologi dan Industri, Vol. 12, No. 2, Juni 2015, pp. 286 -291
ISSN 1693-2390 print/ISSN 2407-0939 online

Pemodelan Pengguna berdasarkan Klasifikasi SMS Menggunakan Support Vector Machine pada Perangkat Bergerak Android

Muhammad Fikry[1], Yusra[2]
[1,2] Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, UIN Sultan Syarif Kasim Riau
Jl. HR. Soebrantas No. 155 Simpang Baru, Panam, Pekanbaru, 28293
Email: mfikry1980@yahoo.com, usera84@yahoo.com

(Received: 8 April 2015; Revised: 4 Juni 2015; Accepted: 25 Juni 2015)

ABSTRAK

Perangkat bergerak, seperti *smartphone* dan *tablet*, memiliki peranan penting dalam kehidupan sehari-hari penggunanya. Berdasarkan interaksi pengguna perangkat bergerak dapat dibuat suatu model yang merepresentasikan pengguna tersebut. Dalam penelitian ini, dilakukan pemodelan pengguna perangkat bergerak dengan implementasi *proof-of-concept*berupa rancang bangun aplikasi Android yang dapat digunakan untuk mengklasifikasikanSMSpada perangkat bergerak dengan menggunakan SVM (Support Vector Machine).Model klasifikasi SVM dibangun dengan menggunakan 640 SMS sebagai data latih dengan kernel gaussian RBF, serta pemilihan *feature* dengan metode DF. Dari hasil pengujian terhadap 160 SMS sebagai data uji, diperoleh akurasi untuk topik Pribadi sebesar 88.75%, diikuti topik Pekerjaan sebesar 5%.

Kata Kunci: klasifikasi topik, pemodelan pengguna, perangkat bergerak, Support Vector Machine

ABSTRACT

Mobile devices, such as smartphones and tablets, has vital role in the everyday life of its users. Based on its user interaction, a model that represents the user can be created.In this study, we conducted mobile users modeling and implement a proof-of-concept Android application which can be used to classify SMSon mobile device by using SVM (Support Vector Machine).SVM classification model is built by using Gaussian RBF as kernel, and document frequency (DF) as feature selection method. From the results, we obtained an accuracy of 88.75% for Personal topic, and an accuracy of 5%. for Work topic.

Keywords: *mobile device, Support Vector Machine, topic classification, user modeling*

Corresponding Author
Muhammad Fikry
Program Studi Teknik Informatika, Fakultas Sains dan Teknologi,
Universitas Islam Negeri Sultan Syarif Kasim Riau,
Email: mfikry1980@yahoo.com

Pendahuluan

Perangkat bergerak (*mobile device*) memiliki fitur personal, portabel, selalu dibawa, penggunaannya yang mudah dan cepat, serta selalu terkoneksi dengan jaringan. Penggunaan perangkat bergerak tidak hanya untuk menelpon ataupun mengirim pesan, namun penggunaannya telah lebih luas misalnya mengakses situs *web* (termasuk jejaring sosial) dengan menggunakan *web browser*, serta meningkatkan produktifitas penggunanya melalui berbagai perangkat lunak aplikasi bergerak (*mobile application software*, disingkat *mobile app*) yang tersedia di sejumlah *application distribution platform*.

Pada dasarnya, perangkat bergerak digunakan secara perorangan sehingga aplikasi-aplikasi yang terinstalasi di dalamnya hanya digunakan oleh seorang pengguna. Dengan demikian, berdasarkan interaksi pengguna terhadap perangkat lunak sistem operasi maupun aplikasi dapat dibuatkan suatu model yang merepresentasikan pengguna tersebut. Selanjutnya, aplikasi tersebut dapat secara dinamis menyesuaikan (misalnya antar muka dan fungsionalitasnya) dengan model pengguna tersebut melalui penggunaan algoritma *machine learning*, sebagaimana yang dilakukan oleh Bozkir dan Sezer [1], Figura [2], Gerber, et.al. [3], Oliveira et.al. [6], Park et.al.[7], Tsang dan Clarke [9], serta Sahs dan Khan [8].

Algoritma *machine learning* digunakan ketika tidak diketahui algoritma untuk menyelesaikan suatu pekerjaan tertentu, namun tersedia data yang cukup untuk keperluan pembelajaran. Telah banyak algoritma yang dikembangkan untuk *machine learning*, namun masih terdapat tantangan dalam pengimplementasiannya di perangkat bergerak, agar aplikasi dapat lebih cerdas misalnya dalam memberikan rekomendasi kepada penggunanya, sehingga pengguna dapat menggunakan

perangkat bergerak secara lebih efektif dan efisien. Hal ini dikarenakan perangkat bergerak memiliki keterbatasan pada karakteristik fisik, seperti ukuran layar, *memory*, dan kemampuan memproses, serta keterbatasan daya baterai.Sistem operasi bergerak (*mobile operating system*) yang sudah ada, seperti Google Android belum menyediakan dukungan baik pemodelan pengguna maupun layanan *machine learning*. Untuk *machine learning* pada perangkat bergerak, terdapat sejumlah proyek *open source* yaitu android-reasoning-util, ml4android, dan Weka untuk Android. Proyek-proyek tersebut berada dalam kondisi belum selesai dan/atau terlalu kompleks untuk digunakan pada aplikasi perangkat bergerak.Hal tersebut melatarbelakangi kebutuhan untuk mewujudkan dukungan (pemodelan pengguna dan layanan *machine learning*) pada perangkat bergerak.Dalam penelitian ini,dirancang bangun suatu layanan *machine learning*, serta dilakukan implementasi *proof-of-concept*. Klasifikasi dilakukan terhadap teks pada Short Message Service (SMS)yang dikirimkan oleh pengguna perangkat bergerak dengan menggunakan SVM (Support Vector Machine).

Pemodelan Pengguna

Model pengguna (dikenal juga sebagai profil pengguna, persona atau *archetype*) digunakan oleh *developer*perangkat lunak untuk keperluan personalisasi dan peningkatan kegunaan dari produk dan layanan yang dibangunnya. Model pengguna merupakan representasi eksplisit dari seorang pengguna.

Tiga langkah utama dalam proses memodelkan pengguna, yaitu :

1. Mengumpulkan data tentang pengguna.
2. Menganalisis data untuk membangun model pengguna.
3. Menggunakan model pengguna untuk beradaptasi.

Terdapat beberapa pola model pengguna. Model pengguna statismerupakan model pengguna yang paling dasar. Setelah data dikumpulkan, biasanya tidak berubah lagi.Model pengguna dinamis memungkinkan representasi pengguna yang lebih *up to date* dengan mengikuti interaksi pengguna terhadap sistem danperubahan preferensi pengguna. Model pengguna berbasis stereotipdidasarkan pada statistik demografi. Berdasarkan informasi yang dikumpulkan, pengguna diklasifikasikan ke dalam suatu stereotipe umum. Model pengguna yang sangat adaptifmencoba untuk merepresentasikan satu pengguna tertentu, dan oleh karena itu sangat adaptif terhadap sistem. Berbeda dengan model pengguna berbasis stereotip, model ini bertujuan untuk menemukan solusi yang spesifik untuk setiap pengguna. Model ini mengumpulkan banyak informasi pertama kali.

Informasi mengenai pengguna dapat dikumpulkan dengan beberapa cara.Sistem dapat menanyakan fakta tertentu selagi berinteraksi (pertama kali) dengan sistem. (Johnson dan Taatgen, [5]).Sistem juga dapat mempelajari preferensi pengguna dengan mengamati dan menafsirkan interaksi mereka dengan sistem. Dengan menggunakan pendekatan *hybrid*,sistem meminta umpan balik eksplisit dan mengubah model pengguna dengan pembelajaran adaptif.

Setelah sistem mengumpulkan informasi mengenai seorang pengguna, sistem dapat mulai beradaptasi dengan kebutuhan pengguna. Informasi dan fungsi dapat disajikan sesuai dengan minat pengguna, menampilkan hanya informasi yang relevan, menyembunyikan informasi yang tidak dibutuhkan pengguna, menyarankan apa yang dilakukan selanjutnya dan lain lain. Dengan demikian, pada suatu sistem adaptif, adaptasi dinamis untuk pengguna secara otomatis dilakukan oleh sistem itu sendiri berdasarkan model pengguna yang dibangun.

Machine Learning

Supervised Learning adalah paradigma *machine learning* untuk memperoleh informasi hubungan input-output dari suatu sistem yang didasarkan pada satu set pasangan input-output dari sampel *training*. Tujuan dari *supervised learning* adalah untuk membangun suatu sistem cerdas yang dapat belajar memetakan input dan output, dan dapat memprediksi output sistem yang diberikan input baru.

Selama proses *supervised learning*, *training input* x_i diumpankan ke *learning system* dan *learning system* menghasilkan $\tilde{y}_i$. $\tilde{y}_i$kemudian dibandingkan dengan label kebenaran y_i dengan sebuah arbitrator yang menghitung antara keduanya. Perbedaan (*error signal*)kemudian dikirim ke *learning system* untuk menyesuaikan parameter *learner*. Tujuan dari *learning process* adalah untuk mendapatkan kumpulan parameter *learning system* yang optimal yang dapat meminimalkan perbedaan antara $\tilde{y}_i$dan $y_i$untuk semua i yaitu meminimalkan kesalahan total selama seluruh rangkaian *training* data. Algoritma *supervised learning* menganalisa data training dan menghasilkan fungsi kesimpulan, dimana fungsi kesimpulan tersebut harus memprediksi nilai output yang benar untuk setiap input objek yang valid.

Guna menyelesaikan permasalahan supervised learning, dilakukan hal sebagai berikut:

1. Tentukan jenis *training examples*.
2. Mengumpulkan *training set*.
3. Tentukan representasi dari *learned function*.
4. Tentukan struktur *learned function* dan algoritma yang sesuai.
5. Lakukan desain.

6. Evaluasi akurasi *learned function*.

Pattern Recognition merupakan salah satu bidang dalam komputer sains, yang memetakan suatu data ke dalam konsep tertentu yang telah didefenisikan sebelumnya. Konsep tertentu ini disebut *class* dan *category*. SVM merupakan salah satu metode *pattern recognition* dan *learning machine* yang bekerja atas prinsip Structural Risk Minimization (SRM) dengan tujuan menemukan *hyperplane* terbaik yang memisahkan dua buah *class* pada *input space*.

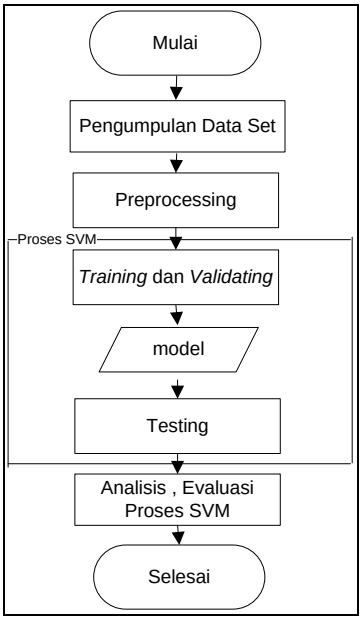
SVM berusaha menemukan *hyperplane* yang terbaik pada *input space*. Prinsip dasar SVM adalah *linear classifier*. Berdasarkan konsep SVM yang berusaha mencari *hyperplane-hyperplane* terbaik pada *input space*, maka problem klasifikasi dapat diterjemahkan dengan usaha menemukan garis (*hyperplane*) yang memisahkan antara kedua kelompok tersebut. Sehingga muncul berbagai alternatif garis pemisah (*discrimination boundaries*).

Dalam pengklasifikasian teks, sering kali melibatkan *tokenization. Tokenization* menurut Herbrich dan Graepel [4] adalah membagi *sequence* karakter ke dalam sebuah teks sebagai *word boundary*. *Tokenization* disebut juga sebagai *word segmentation*. Tujuan *stemming* adalah mengurangi variasi kata yang memiliki kata dasar yang sama.

Penentuan*feature* merupakan tugas yang paling penting dalam klasifikasi teks. Ekstraksi *feature* adalah tugas memilih *term* (kata/istilah) yang akan digunakan dalam *training* set.Menentukan *feature* dapat mengurangi peningkatan ukuran *term* pada proses *training*, dan disamping itu dapat mengurangi *noise* dengan menghapus *feature* yang tidak relevan, sehingga dapat meningkatkan akurasi klasifikasi.Terdapat beberapa pendekatan dalam proses ekstraksi *feature* yang relevan, yaitu : *Document Frequency* (DF), *Inverse Document Frequency* (IDF), *Information Gain* (IG), *Mutual Information* (MI), dan *Chi Square*.

Metode Penelitian

Tahapan-tahapan yang dilakukan dalam penelitian ini diperlihatkan pada Gambar 1.



Gambar 1. Metodologi penelitian

Tahapan-tahapan tersebut dapat dijelaskan sebagai berikut :

1. Pengumpulan *dataset*

   Pada tahap ini, dilakukan pengumpulan *dataset* teks yang dimasukkan oleh pengguna perangkat bergerak pada aplikasi yang ada pada perangkat bergeraknya. Pada tahap ini juga dilakukan rancang bangun perangkat lunak (layanan *machine learning* dan program pendukungnya) untuk mendukung pelaksanaan tahapan-tahapan selanjutnya.

2. *Preprocessing*

   Pada tahap ini, dilakukan proses *tokenization* dan *stemming*, proses ekstraksi dan pembobotan *feature*.

3. Proses SVM

   Pada tahap ini, melibatkan proses SVM, berupa*training* dan *testing*.

4. Analisis dan evaluasi proses SVM

   Pada tahap ini, dilakukananalisis terhadap hasil proses SVM yang dilakukan.

Hasil dan Pembahasan

Pengklasifikasian topik dilakukan terhadap konten SMS (Short Message Service) yang dikirim oleh pengguna perangkat bergerak.Dengan demikian, *dataset* diperoleh dari konten SMS terkirim.

Spesifikasi *dataset* dapat dilihat pada Tabel 1.

Jurnal Sains, Teknologi dan Industri, Vol. 12, No. 2, Juni 2015, pp. 286 -291
ISSN 1693-2390 print/ISSN 2407-0939 online

Tabel 1. Dataset

Jumlah dataset	800 SMS
Jumlah data latih (80% dari dataset)	640 SMS
Jumlah data uji (20% dari data set)	160 SMS

Dataset diklasifikasikan berdasarkan topik, yaitu pekerjaan (SMS yang dikirim untuk urusan pekerjaan) dan pribadi (SMS yang dikirim untuk urusan pribadi/keluarga). Proses pelabelan topik pada *dataset* dilakukan secara manual dengan komposisi berimbang untuk masing-masing topik sebanyak 400 label.

Keseluruhan *dataset* yang telah dilabel selanjutnya dipartisi menjadi dua, 80% diambil sebagai data latih dan 20% diambil sebagai data uji, sebagaimana terlihat pada Tabel 2.

Tabel 2. Data latih dan uji

Klasifikasi	Data Latih	Data Uji
Urusan Pekerjaan	320	80
Urusan Pribadi	320	80

Data latih dan data uji telah melaluiproses *tokenization* (*word segmentation*) dan *stemming*(penentuan kata dasar yang mengacu pada kata dasar).Ekstraksi *feature*menggunakan metode DF (*Document Frequency*) dengan *threshold* nilai minimum. Berdasarkan metode ekstraksi *feature*tersebut, dilihat urutan peringkat kata berdasarkan frekuensi kemunculan kata, dan dipilih yang memenuhi *threshold* nilai minimum.

Pengklasifikasian topik dengan menggunakan SVM dilakukan terhadap data uji dengan input model hasil *training* terhadap keseluruhan data latih.Untuk mendukung hal tersebut, selain telah dibangun program untuk ekstraksi dan pembobotan *feature*, juga dibangun program yang melibatkan SVM pada proses *training* untuk menghasilkan model pembelajaran, dan proses *testing* untuk mengklasifikasikan data uji.

Dalam penelitian ini,digunakan *library* OpenCV4Android, yang memiliki fungsi SVM dan mendukung penggunaan kernel gaussian RBF dengan melewatkan parameter C dan gamma. Untuk mendapatkan pasangan parameter C dan gamma terbaik, OpenCV4Android menyediakan fungsi *train_auto*dengan menetapkan nilai *n-fold cross validation*.Keluaran yang dihasilkan oleh adalah klasifikasi topik, yaitu pekerjaan dan pribadi.

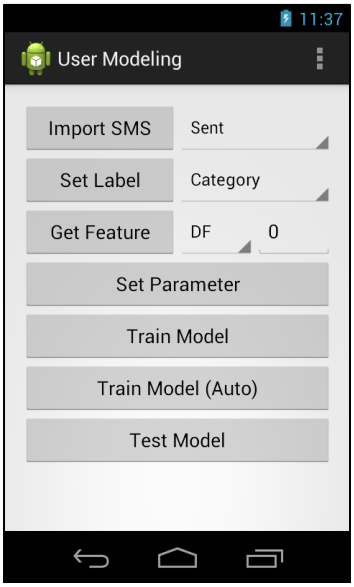
Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang telah dirancang bangun, sebagai berikut:

1. Pengguna dapat mengimpor SMS ke dalam basis data.
2. Pengguna dapat melakukan pelabelan secara manual terhadap SMS yang sudah diimpor.
3. Pengguna dapat melakukan pemilihan *feature*.
4. Pengguna dapat memilih parameter untuk *training* dan *testing* dengan menggunakan SVM.
5. Pengguna dapat melakukan *training*.
6. Pengguna dapat melakukan *testing*.

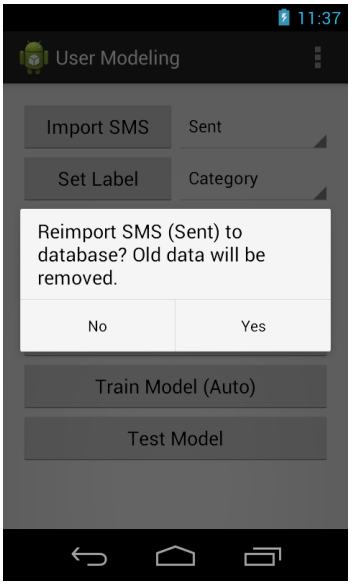
Hasil Implementasi

Pada Gambar 2, tampilan utama menampilkan seluruh fitur yang dimiliki oleh aplikasi.



Gambar 2. Tampilan utama

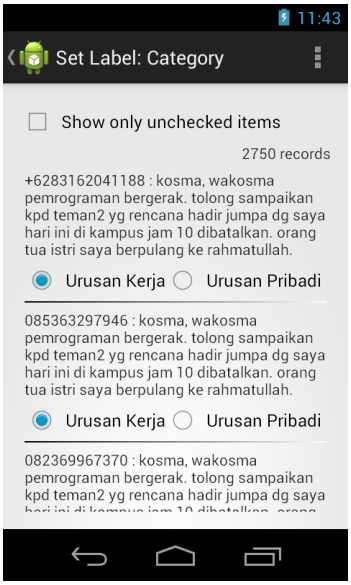
Pada tampilan utama, pengguna dapat mengklik tombol Import SMS untuk menampilkan dialog konfirmasi sebagaimana diperlihatkan pada Gambar 3 untuk memasukkan SMS yang telah dikirim ke dalam *database*. Gambar 4 memperlihatkan hasil impor SMS.



Gambar 3. Tampilan impor SMS

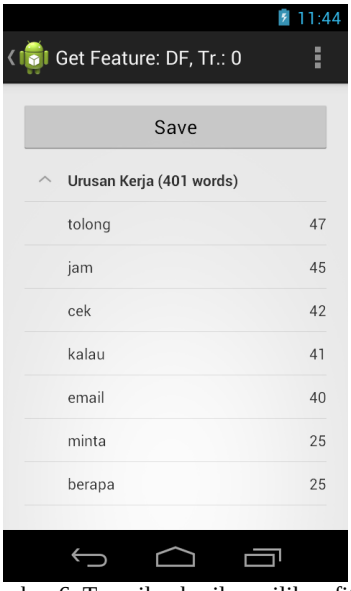
Gambar 4. Tampilan hasil impor SMS

Pengguna dapat melakukan pelabelan terhadap hasil impor SMS yang sudah melalui *preprocessing* sebagaimana diperlihatkan pada Gambar 5.



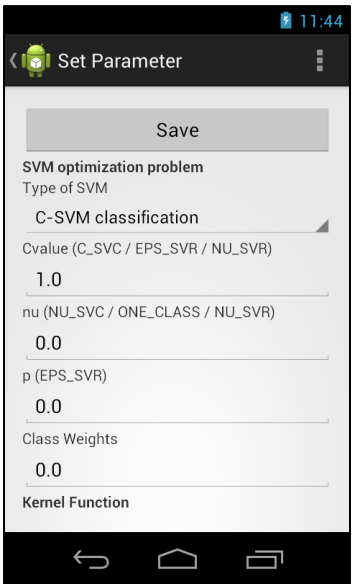
Gambar 5. Tampilan pelabelan SMS

Selanjutnya, pengguna dapat melakukan pemilihan *feature* dengan menggunakan metode DF dan menentukan nilai *threshold*. Gambar 6 memperlihatkan tampilan hasil pemilihan fitur.



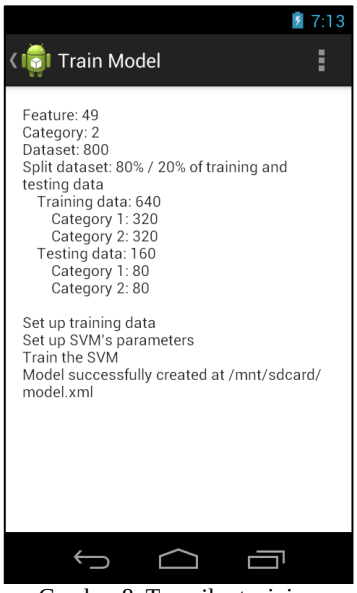
Gambar 6. Tampilan hasil pemilihan fitur

Sebelum melakukan *training* dan *testing*, pengguna dapat menentukan parameter SVM pada Gambar 7.

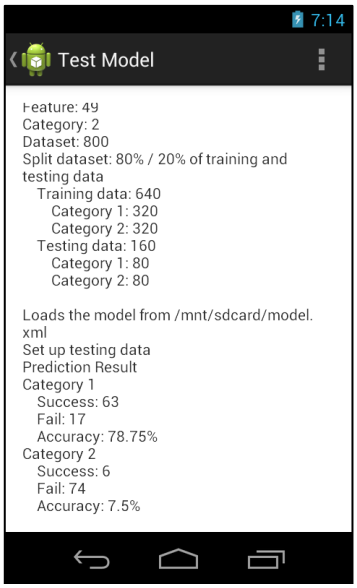


Gambar 7. Tampilan penentuan parameter SVM

Akhirnya, Pengguna dapat melakukan *training* pada Gambar 8 dan *testing* pada Gambar 9.



Gambar 8. Tampilan*training*



Gambar 9. Tampilan *testing*

Kesimpulan

Aplikasi bergerak berbasis Android yang dirancang bangun telah dapat mengklasifikasikan SMS yang dikirim oleh pengguna ke dalam 2 (dua) klasifikasi topik, yaitu Pekerjaan dan Pribadi.Model klasifikasi SVM dibangun dengan menggunakan 640 SMS sebagai data latih dengan kernel gaussian RBF, serta pemilihan feature dengan metode DF. Dari hasil pengujian terhadap 160 SMS sebagai data uji, diperoleh akurasi untuk topik Pribadi sebesar 88.75%, diikuti topik Pekerjaan sebesar 5%.

Untuk penelitian lanjutan,aplikasi dapat menggunakan tipe SVM dan tipe kernel yang berbeda untuk meningkatkan hasil akurasi, ataupun menggunakan metode *machine learning* lainnya.Selain itu, untuk memungkinkan model pengguna yang dimiliki oleh suatu aplikasi dapat digunakan oleh aplikasi lainnya ataupun oleh sistem operasidapat digunakan *framework* Xposed.*Framework* tersebut memungkinkan aplikasi yang sudah terinstal (baik memiliki kode sumbernya ataupun tidak) untuk meng-*updatedataset*, memperbaharui model dan menggunakan model yang ada.

Daftar Pustaka

[1] Bozkir, A.S.,dan Sezer, E., Mobile Mind: A Fully Mobile Platform Based Machine Learning Application, Hacettepe University, Computer Engineering Department, Ankara, Turkey, 2011.

[2] Figura, J., Machine Learning for Google Android, Thesis. Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University in Prague, 2012.

[3] Gerber, S., Fry, M., Kay, J., Kummerfeld, B., Pink, G., dan Wasinger, R., PersonisJ: Mobile, Client-Side User Modelling, UMAP, LNCS 6075, pp. 111–122, Springer-Verlag Berlin Heidelberg 2010.

[4] Herbrich, R., dan Graepel, T., *Handbook of Natural Language Processing*. 2010.

[5] Johnson, A., dan Taatgen, N., User Modeling, *Handbook of Human Factors in Web Design*, Lawrence Erlbaum Associates, pp. 424–439, 2005.

[6] Oliveira, R., Karatzoglou, A., Concejero, P., Armenta, A., dan Oliver, N., Towards a Psychographic User Model From Mobile Phone Usage, Vancouver, BC, Canada, 2011.

[7] Park, M.H., Hong, J.H., dan Cho, S.B., Location-Based Recommendation System Using Bayesian User's Preference Model in Mobile Devices, J. Indulska et al. (Eds.): UIC 2007, LNCS 4611, pp. 1130–1139, Springer-Verlag, Berlin Heidelberg, 2007.

[8] Sahs, J., dan Khan, L., A Machine Learning Approach to Android Malware Detection, *Intelligence and Security Informatics Conference (EISIC)*, 2012.

Tsang, S.L., dan Clarke S., Mining User Models for Effective Adaptation of Context-aware Applications, *International Journal of Security and its Applications Vol. 2*, No. 1, January, 2008.

