PERANCANGAN *FRAMEWORK MOBILE DEVICE MANAGEMENT* PADA *PLATFORM* ANDROID

KurniaAnggriani
Teknik Informatika Universitas Bengkulu

kurniaanggriani@unib.ac.id

Abstrak: Peningkatan penggunaan perangkat *mobile* dan adanya kebijakan *bring your own device* (BYOD) mengakibatkan keragaman data yang tersimpan diperangkat *mobile*, baik data pribadi maupun data perusahaan. Sehubungan dengan hal tersebut, maka meningkat pula ancaman keamanannya berupa *mobile malware* dan ancaman fisik yaitu kehilangan perangkat serta ancaman yang berasal dari perilaku pengguna. Oleh karena itu, dilakukan penelitian untuk merancang *framework mobile device management* yang merupakan *tools* untuk memonitor, mengontrol dan melindungi perangkat *mobile*. *Framewor kmobile device management* dirancang pada *platform* Android. Metode penelitian yang digunakan adalah kualitatif dengan tahapan analisis domain dan desain *framework.* Hasil dari penelitian adalah kebutuhan fungsional *mobile device management* dan rancangan *framework mobile device management* pada *platform* Android. Kebutuhan fungsional *mobile device management* sejumlah 21 fitur yang didapatkan dari perbandingan 3 *software mobile device management* dimana fitur tersebut dimiliki oleh minimal 2 *software*.
Kata Kunci: *framework,mobile device management,*Android.

Abstract: The increasing number of mobile device usage and the policies of bring your own devices (BYOD) cause the diversity of data stored on mobile devices, not only personal data but also corporate data. Based on that fact, the number of threats are also increase. There are mobile malware and physical threats, such as loss of mobile devices and threats from user behavior. Therefore, the research conducted to develop a mobile device management framework, a tools to monitor, control and protect mobile devices. Qualitative method is used, domain analysis, the framework design and a framework prototype. The results of the study are the functional requirements of mobile device management, and a prototype framework of mobile device management on the Android platform. Functional requirements of mobile device management is 21 features obtained from the comparison of 3 mobile device management software wherein the features possessed by at least 2 software. Mobile device management framework consists of a server and a client. The server consists of a mobile device management, policies, networks and applications. Clients consist of seeing the policy and send a message to the server. Protoype has implemented 12 of the 21 features. The framework of mobile device management is developed on the Android platform. The results of the research are the functional requirements of mobile device management and a prototype framework of mobile device management on the Android platform. Testing of the framework is using black box testing, the benefits of framework testing and expert judgement.
Keywords: framework, mobile device management, Android.

I. PENDAHULUAN

Kebijakan *bring your own device* (BYOD) mengakibatkan penggunaan perangkat *mobile* tidak hanya untuk kepentingan pribadi tetapi juga untuk kepentingan pekerjaan[1]. Berdasarkan survey mengenai keamanan perangkat *mobile,* seiring meningkatnya penggunaan perangkat *mobile* maka meningkat pula ancaman keamanannya[2]. Selain ancaman yang berupa *mobile malware*, ancaman juga dapat berupa ancaman fisik yaitu kehilangan perangkat *mobile* dan ancaman yang berasal dari perilaku pengguna[3].

Mobile device management merupakan *tools* yang digunakan untuk memonitor, mengontrol dan melindung perangkat *mobile*[4]. *Mobile device management* mencakup keamanan perangkat, aplikasi, jaringan dan data[5]. *Framework* merupakan aplikasi *semicomplete* yang dapat digunakan untuk mengembangkan aplikasi pada domain tertentu [6].

II. TINJAUAN PUSTAKA

A. *Framework*

Framework merupakan salah satu teknik guna ulang yang digunakan dalam proses pengembangan aplikasi. *Framework* menyediakan bagian untuk diadaptasi oleh pengembang aplikasi sedangkan bagian lainnya merupakan bagian yang sudah tetap dan tidak akan dimodifikasi lagi. Untuk menjadikan *framework* menjadi sebuah aplikasi yang lengkap, pengembang aplikasi harus melakukan adaptasi terhadap *framework* tersebut. Adaptasi dilakukan untuk menambah fungsionalitas yang dibutuhkan kedalam *framework.*

Berikut beberapa definisi *framework* menurut literature yang ada:

1. *Framework* didefinisikan sebagai sebuah aplikasi *semicomplete* yang digunakan kembali dan dapat dimodifikasi untuk menghasilkan aplikasi tertentu[7].
2. *Framework* adalah kerangka sebuah aplikasi yang dapat dimodifikasi dan disesuaikan oleh pengembang aplikasi[6].
3. *Framework* adalah sekumpulan kelas yang saling bekerjasama yang membentuk desain yang dapat digunakan kembali untuk kelas perangkat lunak tertentu[8].

Framework dikategorikan sebagai abstraksi arsitektur. Empat aspek abstraksi arsitektur yang berhubungan dengan *framework* sebagai berikut[6]:

1. Struktur

Framework tersusun dari *class* yang berkolaborasi dan bertanggung jawab secara spesifik

2. Fungsionalitas

Dalam rangka menyediakan fitur domain yang diperlukan, *framework* mewujudkan fungsi umum aplikasi didomainnya. Selanjutnya, agar dapat digunakan dipengembangan aplikasi, *framework* harus memiliki fungsi adaptasi. Sebuah contoh adalah menyediakan *hotspot* yang dapat dikonfigurasi.

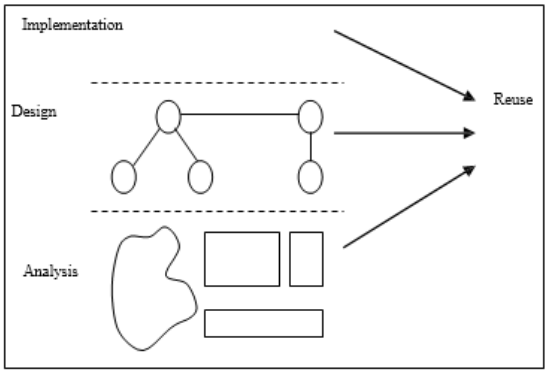
3. Abstraksi

Framework menggambarkan abstraksi dari struktur dan fungsionalitas didomain sebagai generalisasi struktur dan fungsionalitas. Generalisasi struktur dan fungsionalitas dicakup dalam seluruh kesatuan.

4. Penggunaan ulang

Framework mengungkapkan kesamaan antara sekelompok aplikasi dan dapat digunakan kembali ketika mengembangkan aplikasi dalam kelompok ini.

Dari sudut pandang abstraksi arsitektur, *framework* menyediakan fungsi penggunaan ulang pada tiga level yaitu analisis, desain dan implementasi seperti terlihat pada Gambar 1.



Gambar 1. Penggunaan ulang *framework* pada analisis, desain dan implementasi

B. Ancaman Pada Perangkat Mobile

Perangkat *mobile* harus mendukung tujuan keamanan. Tujuan dasar yang utama dari keamanan untuk perangkat *mobile* adalah sebagai berikut:

1. *Confidentiality*, menjamin bahwa data yang tersimpan diperangkat *mobile* tidak dapat diakses oleh pihak yang tidak berwenang.
2. *Integrity*, mendeteksi perubahan yang disengaja atau tidak disengaja pada data yang tersimpan.
3. *Availability*, menjamin bahwa pengguna dapat mengakses sumber daya menggunakan perangkat *mobile* ketika dibutuhkan.
4. Untuk mencapai tujuan tersebut, perangkat *mobile* harus diamankan dari beragam ancaman dan kelemahan. Berikut adalah ancaman dan kelemahan perangkat *mobile*[9]:
    1. Kurangnya control terhadap keamanan fisik.
    2. Penggunaan perangkat mobile yang tidak terpercaya.
    3. Penggunaan jaringan yang tidak terpercaya.
    4. Penggunaan aplikasi yang tidak terpercaya.
    5. Interaksi dengan sistem lain.
    6. Penggunaan konten yang tidak terpercaya.
    7. Penggunaan layanan lokasi.

C. Mobile Device Management

Mobile device management adalah serangkaian manajemen mobilitas dan *tools* keamanan. *Mobile device management* dapat dikembangkan dengan konfigurasi berbasis lokasi atau sebagai *hosted service*[4]. *Mobile device management* tidak hanya mengatur data yang tersimpan di perangkat *mobile* tetapi juga *hardware* seperti kamera dan *port* USB dari perangkat *mobile*[5]. *Mobile device management* mengacu kepada *frameworks* atau solusi yang mengontrol, memonitor dan mengatur penggunaan perangkat *mobile* diperusahaan atau penyedia layanan[10].

III. ANALISIS DOMAIN

Analisis domain dilakukan dengan mengeksplorasi *software mobile device management* yaitu Aegis Safer, Blackberry Enterprise dan MaaS360 [11,12,13]. *Software mobile device management* yang digunakan dapat diimplementasikan di *platform* Android, iOS dan Blackberry. Setelah dieksplorasi, dilakukan perbandingan fitur yang dimiliki ketiga *software mobile device management* tersebut. Hasil dari perbandingan fitur akan menjadi kebutuhan fungsional *framework mobile device management*. Kebutuhan fungsional setidaknya didukung oleh minimal dua *software.* Tabe1. merupakan kebutuhan fungsional dari *framework mobile device management.*

Tabel 1. Kebutuhan Fungsional *Framework Mobile Device Management*

No	KebutuhanFungsional
1	Admin MDM dapat mendaftarkan perangkat *mobile* ke *software Mobile device management*
2	Admin MDM dapat melakukan otentikasi perangkat
3	Admin MDM dapat mengatur penggunaan kamera (*disable/enable*).
4	Admin MDM dapat mengatur penggunaan *microphone* (*disable/enable*).
5	Admin MDM dapat mengatur penggunaan *SD card* seperti penggantian *SD card*.
6	Admin MDM dapat mengatur penggunaan SIM seperti penggantian SIM.
7	Admin MDM dapat mengatur keluar (*disable/enable*)
8	Admin MDM dapat mengunci layar perangkat secara jarak jauh
9	Admin MDM dapat *reset* perangkat
10	Admin MDM dapat mengatur penggunaan *GPS*
11	Admin MDM dapat mengatur penggunaan *WiFi*
12	Admin MDM dapat mengatur penggunaan *bluetooth*

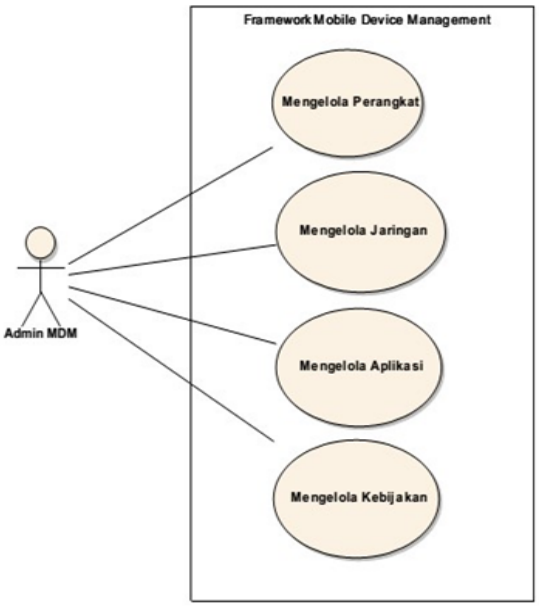
13	Admin MDM dapat mengatur penggunaan *tethering*
14	Admin MDM dapat membuat *blacklist* aplikasi
15	Admin MDM dapat mengirimkan informasi *blacklist* aplikasi ke *Client*
16	Admin MDM dapat membuat *whitelist* aplikasi
17	Admin MDM dapat mengirimkan informasi *whitelist* aplikasi ke *Client*
18	Admin MDM dapat membuat kebijakan yang sesuai dengan organisasi
19	Admin MDM dapat mengatur kebijakan yang berlaku
20	Pengguna dapat melihat kebijakan yang berlaku pada perangkatnya
21	Pengguna dapat mengirim pesan kepada server

IV. DESAIN *FRAMEWORK*

Desain framework terdiri dari pemodelan fungsionalitas, arsitektur *framework*, identifikasi kelas, perancangan kelas dan identifikasi *hotspot.*

A. Pemodelan Fungsionalitas

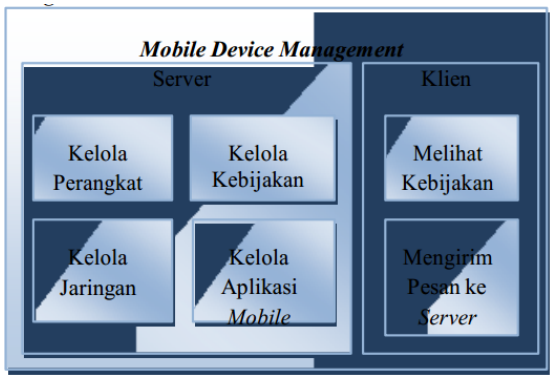
Pemodelan fungsionalitas digambarkan dengan diagram *use case.* Diagram *use case* dibuat berdasarkan kebutuhan fungsionalitas *framework* yang telah ditentukan sebelumnya. Penggunaan *use case* dalam pemodelan fungsionalitas dapat memberikan gambaran fungsi-fungsi aplikasi spesifik yang dapat dibuat dengan menggunakan *framework* tersebut. Gambar 2. merupakan salah satu gambar *use case diagram.*



Gambar 2.*Use Case Diagram Framework Mobile Device Management*

B. Arsitektur Framework

Arsitektur *framework* terdiri dari server dan klien. Dari sisi server terdiri dari kelola perangkat *mobile*, kelola aplikasi dan kelola kebijakan. Dari sisi klien terdiri dari melihat kebijakan dan mengirim pesan ke server. Gambar 3. merupakan arsitektur *framework mobile device management.*



Gambar 3. Arsitektur *Framework Mobile Device Management*

C. Identifikasi Kelas

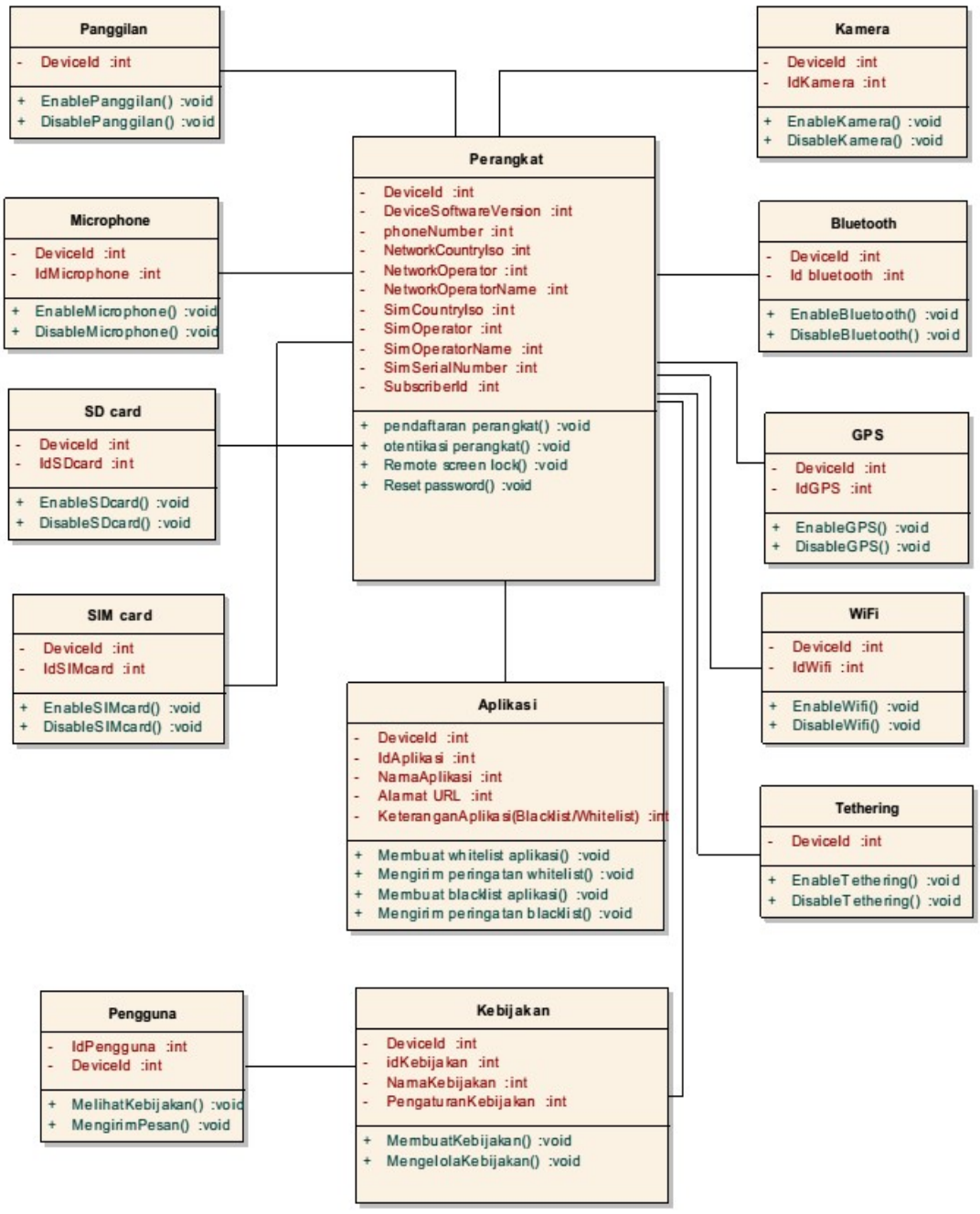
Identifikasi kelas dilakukan setelah melakukan analisis domain. Dengan melakukan analisis terhadap kelas-kelas yang teridentifikasi maka diharapkan akan terlihat relasi antar kelas yang terkait. Berikut ini adalah kelas pada *framework mobile device management:*

1. Kelas Perangkat merupakan kelas yang terdiri dari atribut perangkat *mobile* dan *method* untuk mengelola perangkat *mobile.*
2. Kelas Aplikasi merupakan kelas yang terdiri dari atribut aplikasi dan *method* untuk mengelola aplikasi pada perangkat *mobile*.
3. Kelas Kebijakan merupakan kelas yang terdiri dari atribut kebijakan dan *method* untuk mengelola kebijakan yang akan diterapkan pada perangkat *mobile*.

4. Kelas Kamera merupakan kelas yang terdiri dari atribut kamera dan *method* untuk mengontrol kamera pada perangkat *mobile*.
5. Kelas *Bluetooth* merupakan kelas yang terdiri dari atribut *bluetooth* dan *method* untuk mengontrol bluetooth pada perangkat *mobile*.
6. Kelas GPS merupakan kelas yang terdiri dari atribut GPS dan *method* untuk mengontrol GPS pada perangkat *mobile*.
7. Kelas WiFi merupakan kelas yang terdiri dari atribut WiFi dan *method* untuk mengontrol WiFi pada perangkat *mobile*.
8. Kelas *Microphone* merupakan kelas yang terdiri dari atribut *microphone* dan *method* untuk mengontrol *microphone* pada perangkat *mobile*.
9. Kelas SD *card* merupakan kelas yang terdiri dari atribut SD *card* dan *method* untuk mengontrol SD *card* pada perangkat *mobile*.
10. Kelas SIM *card* merupakan kelas yang terdiri dari atribut SIM *card* dan *method* untuk mengontrol SIM *card* pada perangkat *mobile*.
11. Kelas *Tethering* merupakan kelas yang terdiri dari atribut dan *method* untuk mengontrol *tethering* pada perangkat *mobile*.
12. Kelas Panggilan merupakan kelas yang terdiri dari atribut dan *method* untuk mengontrol panggilan pada perangkat *mobile*.
13. Kelas Pengguna merupakan kelas yang terdiri dari atribut pengguna dan *method* untuk melihat kebijakan dan mengirim pesan pada *server*.

D. Perancangan Kelas

Berdasarkan hasil identifikasi kelas yang telah dilakukan pada sub bab sebelumnya, maka setelah itu kita dapat melakukan perancangan kelas. Perancangan kelas dilakukan dengan menggunakan deskripsi yang terdapat pada proses identifikasi kelas. Pada tahap ini kita akan memodelkan anggota kelas yaitu *properties* dan *method* sesuai dengan deskripsi dan analisis domain sebelumnya. Berikut ini dapat kita lihat relasi antar kelas yang saling berkolaborasi sehingga *framework* dapat bekerja dengan baik.

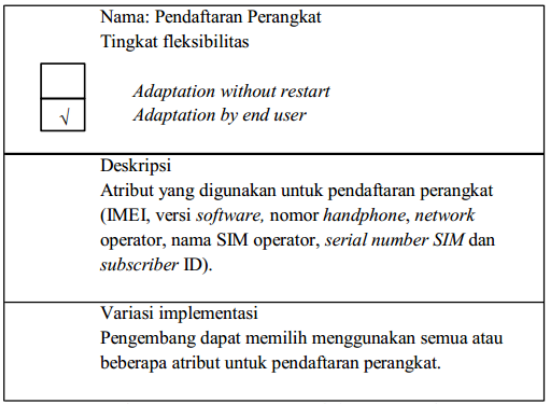


Gambar 4. Diagram *Class Mobile Device Management*

E. Identifikasi Hotspot

Hotspot merupakan bagian dari *framework* yang dapat diubah sesuai dengan keperluan pengembang aplikasi. Dengan melakukan eksplorasi lebih mendalam terhadap fitur-fitur aplikasi pada domain yang sama, dapat diketahui fungsi-fungsi mana saja yang berubah pada tiap aplikasi yang dihasilkan oleh *framework*. Dari hasil eksplorasi tersebut dihasilkan kartu *hot-spot* untuk mengidentifikasi kebutuhan aplikasi yang berubah-ubah.

Hotspot pendaftaran perangkat dengan variasi implementasi dimana pengembang dapat memilih menggunakan semua atau beberapa atribut untuk pendaftaran perangkat. Gambar 5. Merupakan kartu *hotspot* pendaftaran perangkat.



Gambar 5. Kartu *Hotspot* Pendaftaran Perangkat

Hotspot membuat kebijakan dengan variasi implementasi dimana pengembang dapat menambah atau mengurangi kontrol untuk kebijakan sesuai dengan kebutuhan *mobile device management* yang dikembangkan.

Hotspot mengelola aplikasi dengan variasi implementasi dimana pengembang dapat memilih cara mengelola aplikasi, yaitu *Blacklist* atau *Whitelist.*

V. KESIMPULAN

Kesimpulan dari penelitian ini adalah:

1. *Framework mobile device management* telah berhasil dirancang yang terdiri dari klien dan server.
2. Kebutuhan *mobile device management* sejumlah 21 yang didapatkan dari hasil analisis domainya itu eksplorasi dan perbandingan fitur tiga *software mobile device management* yang telah ada.

REFERENSI

[1] Miller, Keithdkk. (2012). "BYOD: Security and Privacy Considerations". IEEE.

[2] Polla, M dkk. (2013)"A Surveyon Security for Mobile Devices". IEEE Communications Survey & Tutorials. Hal. 446-471.

[3] Mohan, Felix. (2013) "Realizingthe Mobile Enterprise: Balancing the Risksand Rewards of Consumer Devices". Report Based on Discussions with the Security for Business Innovation Council.

[4] Tatte, G dan Bamnote, D. (2013). "Mobile Device Management: A Functional Overview". International Journal of Computer Science and Applications. Hal.319-323

[5] Rhee, K dkk. (2012). Security Requirements of a Mobile Device Management System.International Journal of Security and Its Applications. Hal.353-358.

[6] Fayad, M dkk.(1999). *Building Application Frameworks: Object Oriented Foundations of Framework Design*. NewYork: Willey.

[7] R.E,F dan Johnson, B. (1988). *Design Reusable Classes. Journal of Object Programming.* MA: Addison Wesley

[8] Gamma, E dkk. (1995). *Design Patterns: Elements of Reusable Object Oriented Software*. Reading, MA: Addison Wesley.

[9] NIST. (2013). "Guidelines for Managing and Securing Mobile Devices in the Enterprise".

[10] Bergman, N dkk. (2013). *Hacking Exposed: Mobile Security and Solutions*. New York: Mc Graw Hill.

[11] Mark Any. [*Online*]. Tersedia: www.markany.com. Diunduh tanggal 18 Januari 2014.

[12] Blackberry Enterprise 10. [*Online*]. Tersedia: http://us.blackberry.com/business/software/bes-10.html. Diunduh tanggal 18 Januari 2014.

[13] MaaS360.[*Online*].Tersedia: http://www.maas360.com/. Diunduh tanggal 18 Januari 2014.