APLIKASI REMOTE *MONITORING SMARTPHONE ANDROID* MENGGUNAKAN LAYANAN *SHORT MESSAGE SERVICE* (SMS)

Hadi Pranoto[1], Eko Budi Setiawan[2]

[1,2] Universitas Komputer Indonesia
Jl. Dipatiukur 112 Bandung
E-mail : hadipranoto96@gmail.com[1], eko@email.unikom.ac.id[2]

ABSTRAK

Android telah memperbarui sistem dari setiap versi yang dilepaskannya. Penambahan *Application Programming Interface* (API) ini dilakukan setiap kali Google merilis sistem operasi Android yang baru. Ketersediaan API untuk aplikasi pihak ketiga memberi kesempatan bagi pengembang untuk dapat memantau smartphone Android. Sama seperti Google Device Manager yang bisa menginstruksikan *smartphone* Android melalui jaringan internet, namun tetap memiliki kekurangan yaitu jika target smartphone dalam keadaan tidak aktif internet. Dalam penelitian ini, penulis memanfaatkan media SMS untuk dapat memproses instruksi dan mengakses sistem API untuk keperluan pemantauan. Hasil dari penelitian ini adalah dengan menggunakan SMS maka pengguna dapat menginstruksikan *smartphone* Android untuk mengambil foto, mendapatkan lokasi terkini, menelepon, menghapus *file smartphone*, mengatur proteksi layar, dan kontak cadangan dengan keandalan pesan yang lebih tinggi. Aplikasi ini bisa berjalan dengan baik di Android *Lollipop* 5.1 (API Level 22) atau di atas karena memiliki cukup API untuk mendukung fungsionalitas sistem.

Kata kunci : Android, *Application Programming Interface*(API), SMS, *Remote monitoring*, GPS, Device Admin.

1. PENDAHULUAN

Android adalah sebuah kumpulan perangkat lunak yang bersifat open source yang mencakup sistem operasi, *middleware*, dan aplikasi kunci bersama dengan satu set Application Programming Interface (API) untuk menulis aplikasi mobile yang dapat membentuk tampilan, rasa, dan fungsi dari sebuah perangkat mobile [1]. Android telah mengalami pembaharuan sistem dari setiap versi yang dirilisnya. Sebagai contoh yaitu pada sistem operasi Android *Marshmallow* (API 23) yang telah memberikan dukungan API untuk pengenalan sidik jari yang memungkinkan pengguna menggunakan sidik jari mereka untuk membuka perangkat, otentikasi Play Store, dan otentikasi aplikasi pihak ketiga. Dengan adanya fitur tersebut akan membuat pengguna lebih "*aware*" terhadap aktivitas-aktivitas yang dilakukan oleh aplikasi didalam sistem Android.

Meskipun pada sistem operasi Android telah tersedia API untuk menjalankan fungsionalitas sebuah perangkat *mobile*, masalah yang sering muncul adalah saat perangkat Android hilang atau tertinggal di suatu tempat. Tindakan yang mungkin dilakukan untuk membantu mencari keberadaan perangkat yang tertinggal di suatu tempat yaitu dengan *monitoring* menggunakan aplikasi yang dapat mengakses API Android seperti Android Device Manager yang disediakan Google. Namun metode tersebut masih memiliki kelemahan dalam melakukan *monitoring* yaitu saat perangkat dalam keadaan tidak aktif paket data internet maka perangkat Android akan lebih sulit dijangkau oleh pengguna.

Berdasarkan hasil pengolahan data kuesioner tentang pola perilaku dalam menggunakan smartphone yang disebar kepada 70 responden, diketahui 18.57% responden sering menggunakan *airplane mode* dan 32.85% responden aktif menggunakan paket data internet saat perangkat digunakan saja. Dari hal tersebut dapat disimpulkan sebagian orang membatasi *smartphone*-nya dalam menerima pesan komunikasi. Padahal untuk melakukan *monitoring* perangkat Android menggunakan aplikasi pihak ketiga dibutuhkan setidaknya koneksi internet. Berbagai penelitian telah dilakukan sebelumnya untuk menangani kelemahan tersebut seperti penelitian yang dilakukan oleh Kumar dan Qadeer [2], Zhang, Hui, Qizhen, dan Kim Tai-hoon [3], Kuppusamy dan Aghila [4], Rohitaksha, Madhu, Nalini, dan Nirupama [5], Punjabi, Pooja, Mantur, dan Sneha [6], Gupta dan Reddy [7], Dave dan Welekar [8], Chandran [9], yaitu dengan menggunakan layanan Short Message Service (SMS) untuk melakukan kendali jarak jauh seperti yang dilakukan oleh Rayarikar, Sanket, dan Pimpale [10], dan Jongseok dan Kim Howon [11]. Namun pada penelitian dilakukan oleh Kumar dan Qadeer [2] dan Zhang, Hui, Qizhen, dan Kim Tai-hoon [3] pesan yang digunakan untuk melakukan *monitoring* belum menggunakan enkripsi sehingga harus mencantumkan *password* dan *keyword* secara

plain text. Kemudian pada penelitian Kuppusamy dan Aghila [4] telah menggunakan proses enkripsi dan dekripsi didalam sistemnya, namun tidak menggunakan server sehingga respon yang dapat diterima pengguna hanya berupa informasi berbasis teks.

Masalah lain juga dapat muncul saat kehilangan *smartphone* yaitu seseorang dapat saja mematikan daya terhadap smartphone pengguna. Hal ini terjadi karena sistem di Android memperbolehkan aksi mematikan daya secara langsung dengan menggunakan tombol fisik *power off*. Apabila *smartphone* telah dalam keadaan mati maka seseorang dapat saja melakukan pergantian kartu SIM atau bahkan *factory reset*. Apabila SIM *Card* diganti maka akan menjadi masalah karena *smartphone* tidak dapat di-*monitoring* baik menggunakan internet maupun melalui SMS.

Dengan demikian, aplikasi yang akan dibangun dalam penelitian ini diharapkan untuk dapat mengatasi keterbatasan internet dalam menyampaikan instruksi *monitoring* sehingga dapat membantu meningkatkan reabilitas penyampaian informasi menggunakan layanan SMS.

2. ISI PENELITIAN

2.1 Komponen Aplikasi Android

Aplikasi Android ditulis dalam bahasa pemrograman Java. Kode Java dikompilasi bersama dengan data *file* yang dibutuhkan oleh aplikasi dimana prosesnya dikemas oleh *tools* yang dinamakan "*apt tools*" kedalam paket Android sehingga menghasilkan *file* dengan ekstensi apk. *File* apk itulah yang kita sebut dengan aplikasi dan nantinya dapat dipasang pada perangkat *mobile* [19]. Terdapat empat jenis komponen pada aplikasi Android yaitu activity, service, broadcast receiver, dan content provider.

Activity akan menyajikan *User Interface* (UI) kepada pengguna sehingga pengguna dapat melakukan interaksi. Sebuah aplikasi Android bisa jadi hanya memiliki satu *activity*, tetapi umumnya aplikasi memiliki banyak *activity* yang bergantung pada tujuan aplikasi dan desain aplikasi itu sendiri. Untuk berpindah dari satu *activity* ke *activity* lain dapat dilakukan menggunakan sebuah *trigger* seperti klik tombol pada tampilan aplikasi.

Service tidak memiliki *Graphic User Interface* (GUI), tetapi *service* berjalan secara background. Sebagai contoh dalam memutar musik, *service* mungkin memutar musik atau mengambil data dari jaringan, tetapi setiap *service* harus berada dalam kelas induknya. Apabila sebuah pemutar musik sedang memutar lagu dari list yang ada, aplikasi akan memiliki dua atau lebih *activity* yang memungkinkan pengguna untuk memutar sambil memilih lagu baru. Untuk menjaga musik tetap berjalan sebuah *activity* dapat menjalankan *service.*

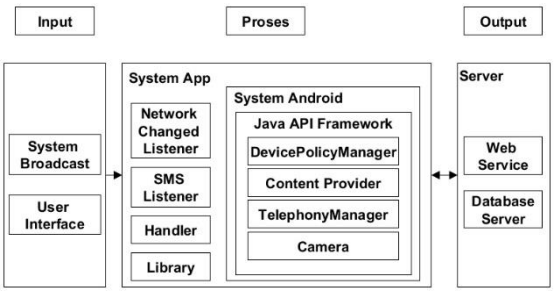
Broadcast receiver berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. Contoh broadcast seperti notifikasi zona waktu telah berubah, baterai lemah, gambar berhasil diambil oleh kamera, dan lain-lain. Aplikasi juga dapat menginisialiasi broadcast misalnya memberikan informasi pada aplikasi lain bahwa ada data yang telah diunduh ke perangkat dan siap untuk digunakan.

Content provider membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam file sistem seperti *database* SQLite. *Content provider* menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*.

Di dalam penelitian ini akan menggunakan seluruh komponen tersebut. *Activity* digunakan penulis untuk berinteraksi dengan pengguna. *Service* digunakan penulis untuk mengeksekusi perintah berdasarkan trigger tertentu. *Broadcast receiver* digunakan untuk mendengarkan SMS masuk dan mendeteksi pergantian SIM *Card*. Content provider digunakan spesifik seperti membaca ID pada SIM *Card.*

2.2 Gambaran Umum Sistem

Gambaran umum dari aplikasi yang akan dibuat dapat dilihat pada Gambar 1.

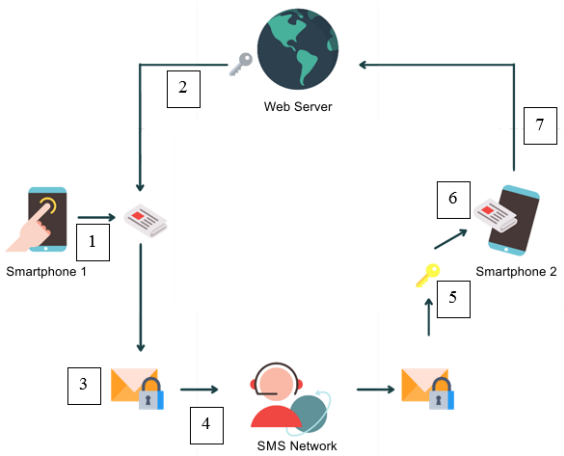


Gambar 1. Gambaran Umum Sistem

Secara umum cara kerja sistem *monitoring smartphone* Android menggunakan SMS dapat dijelaskan sebagai berikut: input berasal dari antarmuka pengguna (*user interface*) atau dari sistem Android yang aktif mem-*broadcast* pesan. Setelah menerima input, sistem aplikasi akan berkomunikasi dengan *Application Programming Interface* (API) pada *smartphone* Android yang bertujuan untuk mengakses sumber daya kamera, kontak, GPS, dan lain-lain untuk memproses input. Setelah berhasil memproses input, apabila menghasilkan data keluaran maka data tersebut akan dikirimkan ke *server.*

2.3 Analisis dan Perancangan Sistem

Sitem yang akan dibangun yaitu aplikasi *remote monitoring smartphone* Android menggunakan layanan *Short Message Service* (SMS). Sistem yang akan dibangun dapat dilihat pada Gambar 2 berikut ini.



Gambar 2. Konsep Sistem Yang Dibangun

Model sistem yang dibangun memiliki tahapan-tahapan sebagai berikut untuk melakukan *monitoring smartphone* Android menggunakan layanan SMS:

a. Tahapan memilih instruksi. Pengguna akan memilih fitur yang disediakan oleh sistem kemudian selanjutnya akan dibentuk *plain text* yang berbentuk URL berparameter.

b. Proses pembacaan kunci publik dari server yang didapatkan oleh pengguna saat *log in* menggunakan *smartphone*-nya.

c. Proses enkripsi *plaintext* menjadi *ciphertext* menggunakan kunci publik yang telah didapatkan sebelumnya. Setelah enkripsi dilakukan penambahan simbol validasi '$$' pada awal teks untuk mengenali apakah SMS tersebut merupakan SMS instruksi atau bukan.

d. Proses mengirimkan SMS instruksi terenkripsi melalui jaringan SMS. Sistem yang dibangun akan membebankan biaya pengiriman SMS kepada pengguna sebanyak satu buah SMS.

e. Proses dekripsi *ciphertext* menggunakan kunci privat yang didapatkan bersamaan dengan kunci publik saat pengguna melakukan *log in* ke dalam sistem. Apabila pesan didekripsi dengan pasangan kunci yang berbeda maka sistem tidak dapat mengeksekusi instruksi *monitoring*.

f. Proses membaca parameter yang ada pada URL yang tersimpan dalam *plaintext* untuk kemudian dieksekusi perintahnya.

g. Proses unggah konten apabila instruksi monitoring adalah ambil foto, lokasi saat ini, atau back up kontak.

2.4 Pemilihan menu instruksi *monitoring*

Didalam sistem yang akan dibangun terdapat menu instruksi yang dapat digunakan untuk proses *monitoring smartphone* Android. Daftar instuksi yang dapat digunakan untuk melakukan *monitoring smartphone* Android dapat dilihat pada Tabel 1 berikut.

Tabel 1. Daftar Instruksi *Monitoring*

Instruksi	**Nilai**
`action_available_camera`	Mengakses kamera yang tersedia untuk mengambil foto
`action_current_location`	Mencari keberaadaan lokasi *smartphone* Android terkini.
`action_ringing`	Menderingkan *smartphone* Android.
`action_erase_external`	Menghapus *file-file* yang tersimpan pada direktori eksternal.
`action_wipe`	Melakukan *wipe* pada *smartphone* Android
`action_lock`	Mengunci *smartphone* Android sesuai dengan pengaturan yang diterapkan oleh pengguna.
`action_lock_new_password`	Mengatur ulang proteksi *smartphone* Android milik pengguna
`action_get_all_contacts`	Melakukan *back up* kontak *smartphone*
`action_get_sim_contacts`	Melakukan *back up* kontak yang tersimpan pada SIM *Card smartphone*
`sim_card_changed`	Digunakan untuk mendeteksi pergantian SIM Card

2.5 Daftar Kebutuhan Hak Akses Sistem

Setiap fitur yang disediakan memiliki kebutuhan sumber daya yang berbeda. Pada tahap ini adalah mendefinisikan akses apa saja yang dibutuhkan agar fungsionalitas sistem dapat berjalan. Kebutuhan hak akses sistem dapat dilihat pada Tabel 2 berikut.

Tabel 2. Daftar Kebutuhan Akses Instruksi

Instruksi	**Kebutuhan Akses**
`action_available_camera`	Akses *hardware* kamera, paket data internet, *storage*
`action_current_location`	Akses GPS, paket data internet
`action_ringing`	Akses audio
`action_erase_external`	Akses *storage*
`action_wipe`	Akses *device admin*
`action_lock`	Akses *device admin*
`action_lock_new_password`	Akses *device admin*
`action_get_all_contacts`	Akses baca kontak, paket data internet
`action_get_sim_contacts`	Akses baca kontak, paket data internet
`sim_card_changed`	Akses baca kontak, akses kirim pesan SMS

2.6 Pembentukan Objek Instruksi

Selanjutnya instruksi akan dibentuk menjadi objek yang dapat diakses sumber dayanya. Bentuk objek yang digunakan dalam sistem yaitu menggunakan format URL sehingga dapat menampung parameter. Parameter tersebut nantinya akan diisi oleh jenis instruksi dan pesan yang ingin disampaikan pengguna terhadap *smartphone* target. Pada tabel 3 berikut adalah instruksi yang telah dibentuk menjadi objek URL.

Tabel 3. Contoh Instruksi Berbentuk URL

Intruksi (Format URL)
`https://catch.id/?action=action_available_camera`
`https://catch.id/?action=action_current_location`
`https://catch.id/?action=action_ringing`
`https://catch.id/?action=action_erase_external`
`https://catch.id/?action=action_wipe`
`https://catch.id/?action=action_lock`
`https://catch.id/action=action_lock_new_password&msg=pass`
`https://catch.id/?action=action_get_all_contacts`
`https://catch.id/?action=action_get_sim_contacts`

Setelah instruksi diubah kedalam bentuk yang dapat diakses sumber dayanya maka tahap selanjutnya adalah bagaimana mengamankan instruksi tersebut. Salah satu cara yang dapat digunakan adalah menggunakan kriptografi. Didalam penelitian ini metode kriptografi yang digunakan adalah RSA (Rivest, Shamir, Adleman). Metode RSA bekerja menggunakan kunci asimetris yang dapat digunakan untuk sistem yang melibatkan beberapa fihak untuk proses enkripsinya. Pada sistem yang dibangun yaitu menggunakan metode RSA 512 bit dimana dapat menampung plaintext sepanjang 64 karakter. Pada tabel 4 adalah contoh keluaran *ciphertext* dari instruksi ambil foto yang dihasilkan melalui metode enkripsi RSA 512 bit yaitu sebagai berikut.

Tabel 4. Contoh Proses Enkripsi Instruksi

Instruksi (Format URL)	***Cipher Text***
`https://catch.id/?action=action_available_camera`	`NjqsFhXSZN5JF/QdsavWYqJKNrAzLV4ymOuMpePcA/lwIqR+BlRmAYekshbKIcaPXO2+IgIum/oZ5RMzMEEWUQ==`

Cipher text pada tabel 4 tidak langsung dikirimkan kepada *smartphone* target. *Cipher text* akan disisipkan simbol "$$" terlebih dahulu supaya validasi SMS dapat dilakukan. Apabila hal ini tidak dilakukan maka sistem akan mencoba melakukan dekripsi pada setiap SMS yang masuk. Pada tabel 5 adalah contoh hasil akhir instruksi mengakses kamera depan yang siap untuk dikirimkan ke *smartphone* target.

Tabel 5. Contoh Instruksi Berbentuk URL

Simbol "$$" + ***Cipher Text***
`$$NjqsFhXSZN5JF/QdsavWYqJKNrAzLV4ymOuMpePcA/lwIqR+BlRmAYekshbKIcaPXO2+IgIum/oZ5RMzMEEWUQ==`

2.7 Analisis *Event Listener* Pada Sistem *Monitoring*

Saat sistem pada Android menerima pesan SMS masuk maka sebetulnya pesan tersebut akan di-*broadcast* ke semua aplikasi yang menggunakan fitur SMS *listener*. Selanjutnya aplikasi akan mencoba untuk memilah apakah pesan SMS tersebut ada kaitannya dengan sistem yang dibangun atau tidak. Apabila SMS mengandung simbol "$$" maka akan dilakukan dekripsi terhadap pesan SMS mulai dari huruf ketiga setelah simbol "$$". Dekripsi dilakukan menggunakan *private key* sehingga menghasilkan *plain text* berbentuk URL yang dapat diakses sumber dayanya. Sistem akan mencoba membaca parameter apa saja yang terlampir pada URL hasil dekripsi dan kemudian memproses sesuai algoritma dari setiap instruksi.

Hal yang sama pula berlaku untuk fitur notifikasi pergantian SIM *card*. Saat terjadi perubahan status *smartphone* seperti dari *airplane mode* menjadi terkoneksi internet maka sebetulnya sistem Android akan mem-*broadcast* pesan ke semua aplikasi yang dapat mendengarkan event tersebut. Selanjutnya aplikasi akan mencoba membaca ID unik pada SIM *card* dan membandingkan dengan data sebelumnya untuk mengetahui apakah terjadi pergantian SIM *card* atau tidak. Adapun pada Tabel 6 ini membahas tentang *listener* pada sistem.
yaitu sebagai berikut.

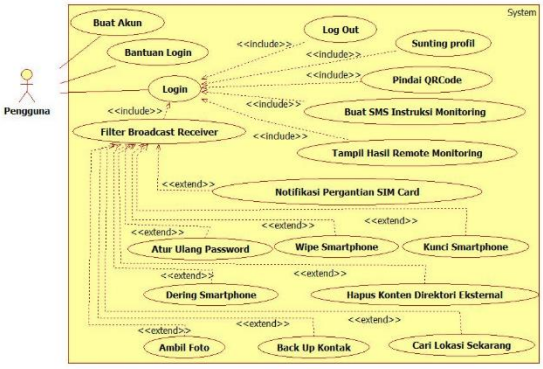
Tabel 6. *Event* dan *Listener* Pada Sistem *Monitoring*

Event	***Listener***
`action_available_camera`	1. Mengindeks jumlah kamera tersedia. 2. Buka akses *hardware* kamera 3. Ambil foto 4. Simpan hasil foto pada penyimpanan eksternal 5. Unggah foto ke server 6. Ulangi langkah 1 sampai seluruh kamera terakses.
`action_current_location`	1. Cek status GPS. 2. Membuat koneksi menggunakan *GoogleApiClient* dan mencari

Event	*Listener*
	koordinat sekarang 3. Unggah informasi ke server berupa *latitude* dan *longitude* apabila tersedia.
`action_ringing`	1. Putar nada dering dari sistem
`action_erase_external`	1. Ambil direktori eksternal 2. Lakukan penghapusan secara rekursif
`action_wipe`	1. Cek apakah terdapat hak akses *device admin* atau tidak 2. Lakukan *wipe* apabila diketahui telah mendapatkan izin
`action_lock_now`	1. Cek apakah terdapat hak akses *device admin* atau tidak 2. Kunci *smartphone* apabila diketahui telah mendapatkan izin
`action_lock_now_new_password`	1. Cek apakah terdapat hak akses *device admin* atau tidak 2. Lakukan *update password* apabila diketahui telah mendapatkan izin
`action_get_all_contacts`	1. Baca kontak *smartphone* 2. Buat sebuah file dan isi dengan data kontak yang dibaca 3. Unggah *file* ke server
`action_get_sim_contacts`	1. Baca data kontak pada SIM *Card* 2. Buat sebuah file dan isi dengan data kontak yang dibaca 3. Unggah *file* ke server
`sim_card_changed`	1. Baca ID pada SIM *Card* 2. Bandingkan dengan ID sebelumnya 3. Apabila terjadi perubahan maka kirim notifikasi ke nomor terpercaya apabila tersedia

2.8 Use Case Diagram Sistem Monitoring

Use case diagram dari sistem *monitoring* terdiri dari 19 *use case* yaitu buat akun, bantuan login, *log in*, *log out*, sunting profil, buat sms instruksi *monitoring*, pindai QRcode, tampil hasil remote *monitoring*, filter *broadcast receiver*, notifikasi pergantian SIM *card*, atur ulang *password*, wipe *smartphone*, kunci *smartphone*, dering *smartphone*, hapus konten direktori ekternal, ambil foto, *back up* kontak, cari lokasi sekarang. Adapun use case diagram dapat dilihat pada gambar 3 berikut.



Gambar 3. Diagram *Use Case* Sistem *Monitoring*

2.9 Implementasi Perangkat Keras

Perangkat keras yang digunakan untuk menguji aplikasi yaitu sebagai berikut:

Tabel 7. Implementasi Perangkat Keras

No.	Perangkat Keras	Spesifikasi
1.	*Processor*	Quad-core dengan kecepatan 1.2 GHz
2.	RAM	3 GB
3.	Layar	5 *inch*
4.	Storage	40 MB
5.	Kamera	16 MP & 3.7 MP

2.10 Implementasi Perangkat Lunak

Perangkat lunak yang digunakan untuk menguji aplikasi yaitu sebagai berikut:

Tabel 8. Implementasi Perangkat Lunak

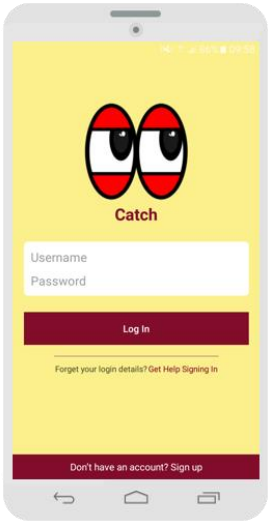
No.	Perangkat Lunak	Spesifikasi
1.	Sistem Operasi Komputer	Windows 8.1 Pro
2.	Android Studio	Versi 2.3.2
3.	Android SDK	Level 22 ke atas
4.	Bahasa pemrograman	Java, PHP, JSON
5.	*Database*	*MySQL*
6.	Sistem Operasi *Smartphone*	Android *Lollipop* 5.1 (API Level 22) keatas

2.11 Implementasi Antarmuka

Antarmuka merupakan sarana bagi pengguna untuk berinteraksi dengan sebuah sistem. Implementasi antamuka dari aplikasi *remote monitoring smartphone* Android menggunakan layanan *Short Message Service*(SMS) yaitu sebagai berikut.

a. Antarmuka Tampilan *Log In*

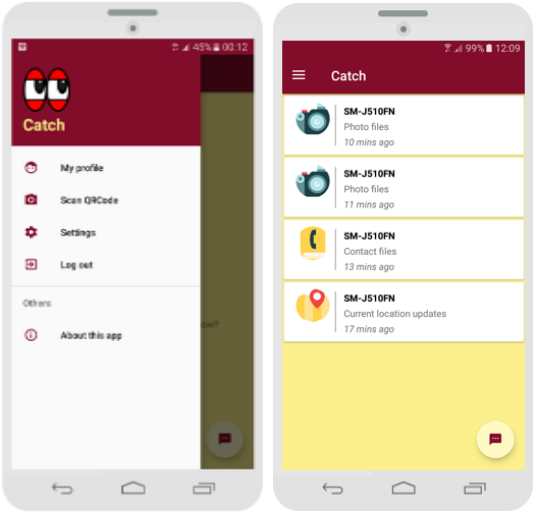
Antarmuka tampilan login digunakan untuk masuk kedalam sistem menggunakan nama pengguna dan kata sandi.



Gambar 4. Antarmuka Tampilan *Log In*

b. Antarmuka Tampilan Beranda

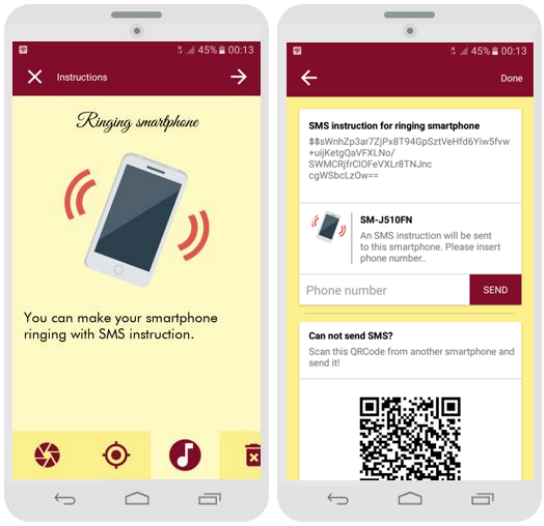
Antarmuka tampilan beranda digunakan untuk melihat hasil *monitoring* yang berupa gambar, lokasi, dan kontak.



Gambar 5. Antarmuka Tampilan Beranda

c. Antarmuka Buat SMS Instruksi

Antarmuka ini digunakan untuk memilih instruksi apa dan memilih *smartphone* mana yang akan di-*monitoring*.



Gambar 6. Antarmuka Buat SMS Instruksi

2.12 Pengujian Sistem

Pengujian dilakukan dengan tujuan untuk menemukan kesalahan dan kekurangan pada perangkat lunak yang diuji. Dengan dilakukannya pengujian maka dapat mengetahui apakah perangkat lunak yang dibuat sudah memenuhi kriteria sesuai dengan tujuan perancangan perangkat lunak. Pengujian dilakukan dengan teknik pengujian *black box* yang lebih berfokus pada menemukan kesalahan program secara fungsional.

Pengujian *black box* dilakukan untuk komponen fungsional autentikasi, filter pesan *broadcast*, membuat SMS instruksi, sunting profil, mengirim SMS, pengaturan, dan pindai gambar QRCode. Dalam pengujian ini dilakukan dengan menggunakan tujuh buah *smartphone* yang berbeda untuk mengukur akurasi keberhasilan aplikasi. Adapun prosentase keberhasilan aplikasi pada perangkat yang berbeda dapat dilihat pada tabel 9 berikut.

Tabel 9. Hasil Pengujian Terhadap Smartphone

Smartphone	**API Level**	**Keberhasilan**
Samsung A3 2017	23	100%
Sony E5	22	98%
Xiaomi Redmi 4	23	100%
Samsung J5 2016	23	100%
Sony C4	23	100%
Xiaomi 5	23	100%
Samsung J7 Prime	23	100%

Berdasarkan hasil pengujian terhadap beberapa *smartphone* android maka dapat disimpulkan bahwa aplikasi dapat berjalan pada *smartphone* yang berbeda jenis, hanya saja masih terdapat kelemahan yaitu *smartphone* Sony E5 yang tidak didukung dengan API yang lengkap pada sistem operasi Android *Lollipop*-nya dimana menyebabkan aplikasi

menjadi *crash* saat membaca informasi kontak. Selain kekurangan tersebut didapatkan bahwa semua fungsional aplikasi dapat berjalan dengan baik pada tujuh *smartphone* android yang digunakan ketika pengujian.

Selain itu, pengujian juga dilakukan pada bulan Agustus 2017 dengan menyebarkan kuesioner terhadap 17 responden yang menggunakan *smartphone* dengan sistem operasi Android *Lollipop* (API Level 22) keatas. Responden memberikan jawaban yang variatif, namun dari nilai rata-rata yang didapat bisa ditarik kesimpulan bahwa responden setuju dengan adanya aplikasi ini memudahkan dalam melakukan *monitoring smartphone* Android miliknya.

Tabel 10. Hasil Pengujian Beta

Pertanyaan pertama : Apakah dengan adanya aplikasi *remote monitoring smartphone* Android menggunakan layanan SMS ini dapat mempermudah Anda dalam melakukan *monitoring smartphone* Android?				
SS	S	RR	KS	TS
8	8	1	0	0
Rata-rata = (75/85) *100% = 88.2%				
Pertanyaan kedua : Apakah dengan adanya aplikasi *remote monitoring smartphone* Android menggunakan layanan SMS ini dapat mempermudah Anda dalam mendapatkan notifikasi pergantian kartu SIM (SIM *Card*)?				
SS	S	RR	KS	TS
3	12	2	0	0
Rata-rata = (69/85) *100% = 81.1%				
Pertanyaan ketiga : Apakah aplikasi *remote monitoring smartphone* Android menggunakan layanan SMS ini mudah saat digunakan?				
SS	S	RR	KS	TS
4	13	0	0	0
Rata-rata = (72/85) *100% = 84.7%				
Pertanyaan keempat : Apakah aplikasi *remote monitoring smartphone* Android menggunakan layanan SMS ini memiliki tampilan yang menarik?				
SS	S	RR	KS	TS
3	11	3	0	0
Rata-rata = (68/85) *100% = 80%				

Berdasarkan hasil perhitungan pengujian *beta* yang telah disajikan pada tabel 10 maka dapat diambil beberapa kesimpulan hasil pengujian yaitu :

a. Pengguna merasa terbantu dalam melakukan *monitoring smartphone* Android miliknya. Dikarenakan dengan hadirnya aplikasi ini maka keterbatasan media internet dalam melakukan interaksi dengan *smartphone* dengan tujuan *monitoring* dapat dilengkapi
b. Pengguna merasa terbantu karena fitur pergantian notifikasi dari smartphone Android miliknya. Dikarenakan dengan hadirnya aplikasi ini maka pergantian SIM *Card* yang tidak diinginkan dapat dihindari
c. Aplikasi ini mudah digunakan karena menggunakan konsep yang sama seperti membuat SMS pada umumnya
d. Aplikasi ini memiliki tampilan yang menarik

3. PENUTUP

3.1 Kesimpulan

Berdasarkan hasil penelitian, analisis, perancangan sistem, dan implementasi serta pengujian sistem, dapat diambil kesimpulan sebagai berikut :

a. Aplikasi yang telah dibangun dapat mempermudah pengguna dalam melakukan *remote monitoring smartphone* Android karena dengan hadirnya aplikasi ini maka keterbatasan media internet dalam melakukan interaksi dengan *smartphone* dengan tujuan *monitoring* dapat dilengkapi.
b. Aplikasi yang telah dibangun dapat membantu dalam memberikan notifikasi saat terjadi pergantian SIM *Card* sehingga pergantian SIM *Card* yang tidak diinginkan dapat dihindari.

3.2 Saran

Didalam penelitian ini perlu dilakukan pengembangan dan penyempurnaan lebih lanjut supaya dapat meningkatkan keamanan sistem dan privasi pengguna yang terlibat didalamnya. Adapun saran-saran terhadap pengembangan aplikasi ini untuk kedepannya adalah sebagai berikut:

a. Pengembangan aplikasi pada *smartphone* yang memiliki arsitektur sistem operasi Android berbeda dari yang diresmikan oleh Google karena aplikasi tidak sepenuhnya cocok untuk beberapa perangkat Android yang beredar di pasaran.
b. Perlu dilakukan penyesuaian apabila sistem operasi yang baru telah dirilis karena aplikasi mungkin akan sulit untuk menjangkau API yang berfungsi mengakses perangkat keras smartphone Android

UCAPAN TERIMA KASIH

Penulis ucapkan terima kasih sebanyak-banyaknya kepada:

a. Sahabat alumni SMAN 27 Bandung diantaranya Azhar Hidayatullah, Seftian Irhaf, Anggianto, dan terutama Regie Aditya Wibawa yang telah mengizinkan perangkat Androidnya digunakan

untuk pengembangan aplikasi selama penelitian ini berlangsung.
b. Rekan-rekan Divisi Hardware diantaranya Abdul Rohim, Risqi Windu, M. Azmi Burhan, Dicky Armansyah yang telah membantu menyukseskan dalam proses publikasi aplikasi hingga dapat digunakan oleh publik.
c. Rekan-rekan dari program studi teknik informatika diantaranya Zamaludin Abdulah, Farid Yahya, Dodi Aprianto, Siddik Tauhid Winarya yang telah memberi dukungan selama penelitian berlangsung.

Dengan bantuan rekan-rekan akhirnya penulis dapat menyelesaikan penelitian ini sebagaimana mestinya.

DAFTAR PUSTAKA

[1] R. Meier and W. , Professional Android Application Development, First Edition ed.

[2] D. Kumar and M. A. Qadeer, "SMS Based Emerging Techniques for Monitoring and Controlling Android Mobiles," *IACSIT International Journal of Engineering and Technology*, vol. 4, p. 6, 2012.

[3] W. Zhang, H. He, Q. Zhang and T.-h. Kim, "Protecting User Privacy on the Android-Based Mobile Platform," *International Journal of Distributed Sensor Networks*, p. 10, 2014.

[4] K. Kuppusamy, R. Senthilraja and G. Aghila, "A MODEL FOR REMOTE ACCESS AND PROTECTION OF SMARTPHONES USING SHORT MESSAGE SERVICE," *International Journal of Computer Science, Engineering and Information Technology (IJCSEIT)*, vol. 2, 2012.

[5] K. Rohitaksha, C. Madhu, B. Nalini and C. Nirupama, "Android Application for Vehicle Theft Prevention and Tracking System," *International Journal of Computer Science and Information Technologies (IJCSIT)*, vol. 5, p. 3, 2014.

[6] K. Punjabi, P. Bolaj, P. Mantur and S. Wali, "Bus Locator via SMS Using Android Application," *International Journal of Computer Science and Information Technologies (IJCSIT)*, vol. 5, p. 2, 2014.

[7] R. Gupta and B. Reddy, "GPS and GPRS Based Cost Effective Human Tracking System Using Mobile Phones," VIEWPOINT, vol. 2, p. 1, 2011.

[8] V. Dave and A. Welekar, "Design of an Android Application to provide Emergency Service," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 3, no. 3, 2015.

[9] A. Chandran, "Smartphone Monitoring System," *International Journal of Computer Science & Engineering Technology (IJCSET)*, vol. 4, p. 4, 2013.

[10] R. Rayarikar, S. Upadhyay and P. Pimpale, "SMS Encryption using AES Algorithm on Android," *International Journal of Computer Applications*, vol. 50, p. 19, 2012.

[11] J. Choi and H. Kim, "A Novel Approach for SMS security," *International Journal of Security and its Applications*, 2012.