



TUGAS AKHIR - KI091391

SISTEM PEMANTAU PEMAKAIAN PERANGKAT BERGERAK ANAK

BASKARA NUR PATRIA
NRP 5111 100 097

Dosen Pembimbing I
Dr.Tech. Ir. R.V. Hari Ginardi, M.Sc.

Dosen Pembimbing II
Ir. F.X. Arunanto, M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2015



TUGAS AKHIR - KI091391

SISTEM PEMANTAU PEMAKAIAN PERANGKAT BERGERAK ANAK

BASKARA NUR PATRIA
NRP 5111 100 097

Dosen Pembimbing I
Dr.Tech. Ir. R.V. Hari Ginardi, M.Sc.

Dosen Pembimbing II
Ir. F.X. Arunanto, M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2015



FINAL PROJECT - KI091391

KIDS MOBILE DEVICE USAGE MONITORING SYSTEM

BASKARA NUR PATRIA
NRP 5111 100 097

Advisor
Dr.Tech. Ir. R.V. Hari Ginardi, M.Sc.
Ir. F.X. Arunanto, M.Sc.

INFORMATICS DEPARTMENT
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2015

LEMBAR PENGESAHAN

Sistem Pemantau Pemakaian Perangkat Bergerak Anak.

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Algoritma Pemrograman
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

BASKARA NUR PATRIA

NRP : 5111 100 097

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Dr.Tech. Ir. R.V. Hari Ginardi, M.Sc.

NIP: 19650518199203100

(pembimbing 1)

Ir. F.X. Arunanto, M.Sc.

NIP: 195701011983031004

(pembimbing 2)

SURABAYA

JUNI 2015

Sistem Pemantau Pemakaian Perangkat Bergerak Anak.

Nama Mahasiswa : Baskara Nur Patria
NRP : 5111 100 097
Jurusan : Teknik Informatika FTIf-ITS
Dosen Pembimbing I : Dr.Tech. Ir. R.V. Hari Ginardi, M.Sc.
Dosen Pembimbing II : Ir. F.X. Arunanto, M.Sc.

ABSTRAK

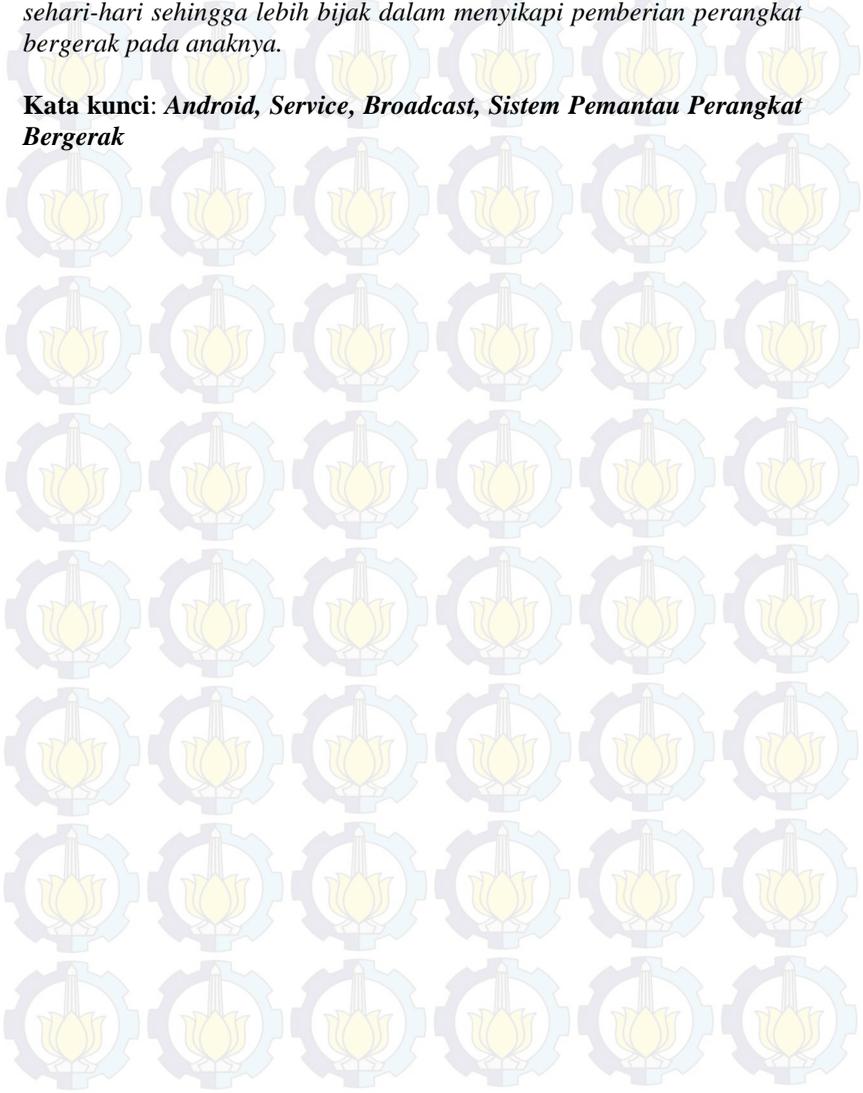
Dewasa ini semakin banyak orang tua yang memberikan perangkat bergerak berupa ponsel cerdas atau tablet kepada anaknya. Ini dilakukan untuk kemudahan komunikasi antara keduanya, maupun sebagai digital babysitter supaya anak lebih tenang. Namun, perlu disadari juga bahwa hal ini dapat membuat anak mengalami ketergantungan terhadap layanan yang diberikan perangkat bergerak tersebut. Layanan-layanan tersebut antara lain video game, pesan singkat, sampai browser. Sebagai orang tua yang memiliki kesibukan dan tidak mungkin mengawasi aktivitas anak dengan perangkat bergeraknya, diperlukan sebuah sistem pemantau perangkat bergerak yang mampu menginformasikan hal ini kepada para orang tua secara rutin.

Sistem pemantau aktivitas perangkat bergerak yang dibuat pada tugas akhir ini terdiri dari sebuah aplikasi perangkat bergerak dan sebuah aplikasi yang berjalan pada server. Aplikasi pada perangkat bergerak ini memanfaatkan layanan service dan broadcast yang disediakan oleh sistem operasi android. Service ini akan berjalan selama perangkat aktif dan berfungsi untuk mencatat seluruh aktivitas perangkat bergerak dan menampilkan hasil record yang telah disimpan. Sementara layanan broadcast digunakan untuk beberapa hal seperti menangkap telepon masuk maupun keluar dan pesan singkat yang masuk ke perangkat. Selain itu, aplikasi ini juga akan mencatat lokasi-lokasi perangkat tersebut berada sehingga orang tua tahu dimana anak beserta perangkatnya berada selama sehari penuh.

Aplikasi perangkat bergerak kemudian mengirimkan data-data yang sudah dicatat ke aplikasi kedua yang berjalan pada server. Nantinya, aplikasi kedua meneruskan data-data tersebut ke alamat

email orang tua. Diharapkan dengan data-data yang dihasilkan oleh sistem ini, orang tua dapat lebih mudah mengevaluasi kegiatan anak sehari-hari sehingga lebih bijak dalam menyikapi pemberian perangkat bergerak pada anaknya.

Kata kunci: *Android, Service, Broadcast, Sistem Pemantau Perangkat Bergerak*



Sistem Pemantau Orang Tua Terhadap Pemakaian Perangkat Bergerak Anak.

Student Name : Baskara Nur Patria
NRP : 5111 100 097
Major : Informatics Engineering FTIf-ITS
Advisor I : Dr.Tech. Ir. R.V. Hari Ginardi, M.Sc.
Advisor II : Ir. F.X. Arunanto, M.Sc.

ABSTRACT

Today, more parents are giving mobile device such as smartphone or tablet to their children. They are doing to facilitate communication between them and as a digital babysitting tool to calm their children. However, we need to realize that this can make children addicted to services or applications that provided by a gadget such as video game, short message service, and browser. As parents who are busy and do not really have time for monitoring children's activities with their gadget, we need a mobile device usage monitoring system that is able to inform their activities to parents regularly.

Mobile device usage monitoring system that developed in this final project consists of a mobile application and a web service application. The mobile application uses system service and broadcast that provided by android. This service will run every time and . Service ini akan berjalan selama perangkat aktif dan serves to record all device's activities and show the records that have been stored. The broadcast service is used for listening some activities such as incoming call, outgoing call, and incoming sms. This application is also able to record the device's location. So, parents will know their children position all day long.

The mobile application will send records to second application that runs in a server. Later, the second application will forward those records to registered parent's email address. Hopefully, parents can evaluate their children's activities easier with their gadget using records that provided by this system.

Keywords: Android, Service, Broadcast, Gadget Monitoring System

KATA PENGANTAR

Puji syukur kepada Allah Yang Maha Esa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan tugas akhir yang berjudul:

Sistem Pemantau Pemakaian Perangkat Bergerak Anak

Melalui lembar ini, penulis ingin menyampaikan ucapan terima kasih dan penghormatan yang sebesar-besarnya kepada:

1. Allah SWT atas segala nikmat dan rahmat yang telah diberikan selama ini.
2. Ibu penulis yang selalu mencurahkan kasih sayang, dukungan, doa dan selalu menjadi motivasi bagi penulis untuk selalu berusaha sebaik-baiknya.
3. Saudara kandung, saudara ipar, dan keponakan penulis yang selalu menjadi motivasi kepada penulis dalam menuntut ilmu dan menyelesaikan buku ini.
4. Bapak Dr.tech. Ir. R. V. Hari Ginardi, M.Sc. dan Bapak Ir. F.X. Arunanto, M.Sc. selaku dosen pembimbing yang telah memberikan arahan, ilmu, motivasi, dan meluangkan waktu untuk membantu pengerjaan tugas akhir ini.
5. Ibu Umi Laili Yuhana selaku dosen wali penulis yang telah meluangkan waktu untuk konsultasi mengenai masalah-masalah akademik.
6. Bapak dan Ibu dosen yang telah membina dan memberikan ilmu kepada penulis selama menempuh studi di Teknik Informatika ITS.
7. Teman-teman yang telah membantu penulis ketika sedang mengalami hambatan teknis dalam pengerjaan

tugas akhir dan hal lainnya: Ali, Indra, Galih, Wilik, Andy, Atnis, Tania, dan Novandi.

8. Teman-teman yang telah memberikan referensi buku tugas akhir kepada penulis: Ghani, Monika, Helmy, dan Fahmi.
9. Teman-teman yang pernah satu kelompok pada tugas kuliah di setiap semester.
10. Keluarga besar angkatan TC 2011 yang menemani keseharian penulis dalam kehidupan kampus.
11. Pihak-pihak lain yang namanya tidak dapat penulis sebutkan satu -persatu.

Bagaimanapun juga penulis telah berusaha semaksimal mungkin dalam menyusun tugas akhir ini, namun penulis mohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, Juni 2015

Baskara Nur Patria

DAFTAR ISI

ABSTRAK	i
ABSTRACT	iii
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Permasalahan.....	2
1.3 Batasan Permasalahan	3
1.4 Tujuan dan Manfaat.....	3
1.5 Metodologi	4
1.6 Sistematika	5
BAB II TINJAUAN PUSTAKA.....	7
2.1 Sistem Pemantau Pemakaian <i>Gadget</i>	7
2.2 Android SDK.....	8
2.3 MPAndroidChart	9
2.4 <i>Reverse Geocoding</i>	9
2.5 Basis Data SQLite	9
2.6 <i>JavaScript Object Notation</i>	10
2.7 <i>Content Provider</i>	10
2.8 <i>Location Based Services</i>	11

2.9	<i>Webservice</i>	11
BAB III ANALISIS DAN PERANCANGAN		13
3.1.	Analisis	13
3.1.1.	Analisis Permasalahan.....	13
3.1.2.	Deskripsi Umum Perangkat Lunak.....	14
3.1.3.	Kebutuhan Fungsional Aplikasi Android	15
3.1.4.	Kebutuhan Fungsional Aplikasi <i>Webservice</i>	23
3.2.	Perancangan	25
3.2.1.	Perancangan Umum Arsitektur Sistem.....	25
3.2.2.	Perancangan Basis Data	26
3.2.3.	Perancangan Algoritma	33
3.2.4.	Perancangan Antarmuka Sistem.....	36
BAB IV IMPLEMENTASI		49
4.1.	Lingkungan Implementasi	49
4.2.	Implementasi Antarmuka.....	49
4.3.	Implementasi Proses Pada Aplikasi Perangkat Bergerak.....	59
4.3.1.	Mencatat Aplikasi yang Dibuka	59
4.3.2.	Mencatat Lokasi Pengguna.....	62
4.3.3.	Mencatat Panggilan Telepon	63
4.3.4.	Mencatat Pesan Singkat Masuk.....	65
4.3.5.	Mencatat Pesan Singkat Keluar	65
4.3.6.	Mencatat Alamat Situs yang Dikunjungi.....	67
4.3.7.	Menampilkan <i>Record</i>	68
4.3.8.	Mengatur Waktu Pengiriman <i>Record</i>	69

4.3.9.	Mengirim <i>Record</i>	69
4.3.9.1.	Memeriksa Jaringan	69
4.3.9.2.	Membuat Struktur JSON	70
4.3.9.3.	Mengirimkan JSON.....	71
4.3.10.	Mengubah Akun.....	72
4.3.11.	Mendaftarkan dan <i>Login</i> Akun.....	72
4.3.12.	Menghitung Statistik	73
4.3.12.1.	Aplikasi	73
4.3.12.2.	Panggilan Masuk dan Keluar.....	74
4.3.12.3.	Pesan Singkat Masuk dan Keluar	74
4.3.12.4.	Durasi	74
4.3.13.	Menampilkan Statistik.....	75
4.3.14.	Menambahkan Favorit.....	76
4.3.15.	Menghapus Favorit.....	77
4.4.	Implementasi Proses Pada Aplikasi <i>Webservice</i>	77
4.5.	Implementasi Tampilan <i>Report</i>	79
BAB V PENGUJIAN DAN EVALUASI		81
5.1.	Lingkungan Pelaksanaan Pengujian	81
5.2.	Pengujian Fungsionalitas.....	81
5.2.1.	Pengujian Melakukan Pendaftaran dan <i>Login</i> Akun	82
5.2.2.	Pengujian Pencatatan Aplikasi	85
5.2.3.	Pengujian Pencatatan Pesan Singkat Masuk	86
5.2.4.	Pengujian Pencatatan Pesan Singkat Keluar	88
5.2.5.	Pengujian Pencatatan Panggilan Telepon Masuk.....	89

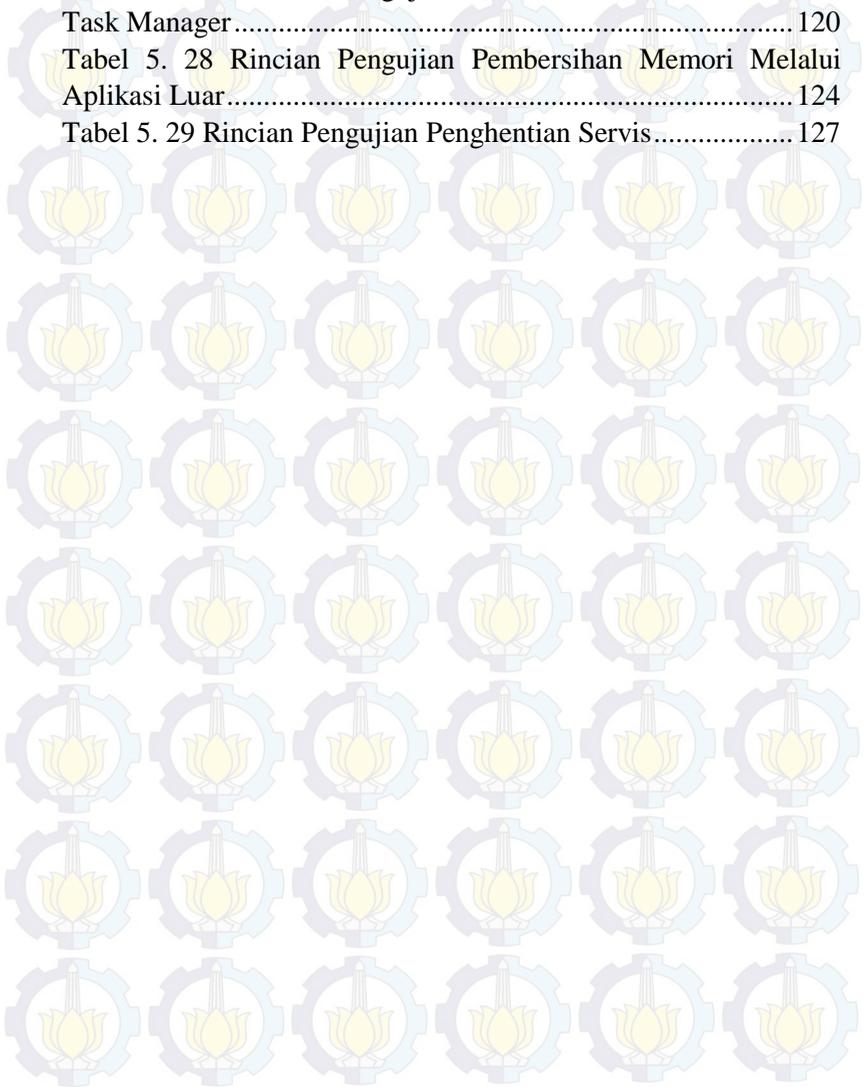
5.2.6. Pengujian Pencatatan Panggilan Telepon Keluar	92
5.2.7. Pengujian Pencatatan Alamat Situs	94
5.2.8. Pengujian Pencatatan Lokasi	95
5.2.9. Pengujian Menampilkan Statistik	97
5.2.10. Pengujian Ubah Akun.....	104
5.2.11. Pengujian Atur Jadwal dan Pengiriman <i>Record</i>	109
5.2.12. Pengujian Penambahan Aplikasi Favorit.....	113
5.2.13. Pengujian Penambahan <i>Domain</i> Favorit	114
5.2.14. Pengujian Penghapusan Aplikasi Favorit	116
5.2.15. Pengujian Penghapusan <i>Domain</i> Favorit.....	117
5.3. Pengujian Non Fungsionalitas	119
5.3.1. Pembersihan Memori Melalui <i>Task Manager</i>	119
5.3.2. Pembersihan Memori Melalui Aplikasi Luar	124
5.3.3. Penghentian Servis Melalui <i>Application Manager</i> ..	127
5.3.4. Konsumsi Memori	129
5.4. Evaluasi Pengujian.....	133
5.4.1. Evaluasi Pengujian Fungsionalitas	133
5.4.2. Evaluasi Pengujian Non Fungsionalitas	136
BAB VI PENUTUP.....	139
5.1. Kesimpulan.....	139
5.2. Saran	140
DAFTAR PUSTAKA.....	141
BIODATA PENULIS.....	143

DAFTAR TABEL

Tabel 3. 1 Daftar Kasus Penggunaan Aplikasi Perangkat Bergerak	19
Tabel 3. 2 Daftar Kasus Penggunaan Aplikasi <i>Webservice</i>	24
Tabel 3. 3 Rincian Tabel Pengguna.....	26
Tabel 3. 4 Rincian Tabel SMS Masuk	27
Tabel 3. 5 Rincian Tabel SMS Keluar	27
Tabel 3. 6 Rincian Tabel Telepon Masuk	28
Tabel 3. 7 Rincian Tabel Telepon Keluar	29
Tabel 3. 8 Rincian Tabel Aplikasi.....	29
Tabel 3. 9 Rincian Tabel Lokasi	30
Tabel 3. 10 Rincian Tabel Situs	30
Tabel 3. 11 Rincian Tabel Aplikasi Favorit	31
Tabel 3. 12 Rincian Tabel Situs Favorit.....	31
Tabel 3. 13 Rincian Tabel Alarm	32
Tabel 3. 14 Rincian Tabel SMS Sementara.....	32
Tabel 3. 15 Keterangan Halaman Antarmuka Daftar	36
Tabel 3. 16 Keterangan Halaman Antarmuka Login.....	38
Tabel 3. 17 Keterangan Halaman Antarmuka Overview	39
Tabel 3. 18 Keterangan Halaman Antarmuka Edit Akun.....	41
Tabel 3. 19 Keterangan Halaman Antarmuka Statistik	43
Tabel 3. 20 Keterangan Halaman Antarmuka Atur Jadwal.....	44
Tabel 3. 21 Keterangan Halaman Antarmuka Menambahkan Favorit	46
Tabel 3. 22 Keterangan Halaman Antarmuka Menghapus Favorit	47
Tabel 5. 1 Spesifikasi Perangkat Bergerak.....	81
Tabel 5. 2 Rincian Pengujian Melakukan Pendaftaran	82
Tabel 5. 3 Rincian Pengujian Melakukan Login	84
Tabel 5. 4 Rincian Pengujian Melakukan Pencatatan Aplikasi... ..	85

Tabel 5. 5 Rincian Pengujian Melakukan Pencatatan SMS Masuk	87
Tabel 5. 6 Rincian Pengujian Melakukan Pencatatan SMS Keluar	88
Tabel 5. 7 Rincian Pengujian Pencatatan Panggilan Masuk dan Diangkat	90
Tabel 5. 8 Rincian Pengujian Pencatatan Missed Call	91
Tabel 5. 9 Rincian Pengujian Pencatatan Panggilan Keluar.....	92
Tabel 5. 10 Rincian Pengujian Pencatatan Alamat Situs.....	94
Tabel 5. 11 Rincian Pengujian Melakukan Pencatatan Lokasi....	96
Tabel 5. 12 Rincian Pengujian Menampilkan Statistik Aplikasi .97	
Tabel 5. 13 Rincian Pengujian Menampilkan Statistik Panggilan Masuk	98
Tabel 5. 14 Rincian Pengujian Menampilkan Statistik Panggilan Keluar.....	100
Tabel 5. 15 Rincian Pengujian Menampilkan Statistik SMS Masuk	101
Tabel 5. 16 Rincian Pengujian Menampilkan Statistik SMS Keluar.....	103
Tabel 5. 17 Rincian Pengujian Mengubah Alamat Email	104
Tabel 5. 18 Rincian Pengujian Mengubah Password	106
Tabel 5. 19 Rincian Pengujian Mengubah Alamat Email dan Password.....	107
Tabel 5. 20 Rincian Pengujian Pengiriman Record Saat Terhubung Internet	109
Tabel 5. 21 Rincian Pengujian Pengiriman Record Saat Tidak Terhubung Internet	111
Tabel 5. 22 Rincian Pengujian Pengiriman Record Saat Mode Pesawat Terbang.....	112
Tabel 5. 23 Rincian Pengujian Menambahkan Aplikasi Favorit	113
Tabel 5. 24 Rincian Pengujian Menambahkan Domain Favorit	115
Tabel 5. 25 Rincian Pengujian Menghapus Aplikasi Favorit....	116

Tabel 5. 26 Rincian Pengujian Menghapus Domain Favorit	118
Tabel 5. 27 Rincian Pengujian Pembersihan Memori Melalui Task Manager.....	120
Tabel 5. 28 Rincian Pengujian Pembersihan Memori Melalui Aplikasi Luar.....	124
Tabel 5. 29 Rincian Pengujian Penghentian Servis.....	127



DAFTAR GAMBAR

Gambar 3. 1 Diagram Modul Aplikasi Perangkat Bergerak	17
Gambar 3. 2 Diagram Kasus Penggunaan Aplikasi Perangkat Bergerak	18
Gambar 3. 3 Diagram Kasus Penggunaan Aplikasi Webservice.....	24
Gambar 3. 4 Arsitektur Umum Sistem	26
Gambar 3. 5 Alur Kerja Aplikasi Perangkat Bergerak.....	33
Gambar 3. 6 Alur Kerja Servis Utama	34
Gambar 3. 7 Alur Kerja Proses Pengiriman Data.....	35
Gambar 3. 8 Rancangan Antarmuka Halaman Daftar.....	36
Gambar 3. 9 Rancangan Antarmuka Halaman Login	38
Gambar 3. 10 Rancangan Antarmuka Halaman Overview	39
Gambar 3. 11 Rancangan Antarmuka Halaman Edit Akun	41
Gambar 3. 12 Rancangan Antarmuka Halaman Statistik.....	42
Gambar 3. 13 Rancangan Antarmuka Halaman Atur Jadwal.....	44
Gambar 3. 14 Rancangan Antarmuka Halaman Menambahkan Favorit	45
Gambar 3. 15 Rancangan Antarmuka Halaman Menghapus Favorit	47
Gambar 4. 1 Antarmuka Halaman Daftar.....	50
Gambar 4. 2 Antarmuka Halaman Login	51
Gambar 4. 3 Antarmuka Halaman Overview	52
Gambar 4. 4 Antarmuka Halaman Overview – Aplikasi	52
Gambar 4. 5 Antarmuka Halaman Overview - Lokasi.....	53
Gambar 4. 6 Antarmuka Halaman Overview - Telepon Masuk.....	53
Gambar 4. 7 Antarmuka Halaman Overview - Telepon Keluar.....	54
Gambar 4. 8 Antarmuka Halaman Overview - Pesan Masuk.....	54
Gambar 4. 9 Antarmuka Halaman Overview - Pesan Keluar.....	55
Gambar 4. 10 Antarmuka Halaman Overview - Browser	55
Gambar 4. 11 Antarmuka Halaman Edit Akun	56

Gambar 4. 12 Antarmuka Halaman Statistik.....	56
Gambar 4. 13 Antarmuka Halaman Atur Jadwal.....	57
Gambar 4. 14 Antarmuka Halaman Menambah Favorit	58
Gambar 4. 15 Antarmuka Halaman Menghapus Favorit.....	58
Gambar 4. 16 Implementasi <i>Report</i>	80
Gambar 5. 1 Hasil Pengujian PA-01	83
Gambar 5. 2 Hasil Pengujian PA-02	84
Gambar 5. 3 Hasil Pengujian PA-03	86
Gambar 5. 4 Hasil Pengujian PA-04	88
Gambar 5. 5 Hasil Pengujian PA-05	89
Gambar 5. 6 Hasil Pengujian PA-06	91
Gambar 5. 7 Hasil Pengujian PA-07	92
Gambar 5. 8 Hasil Pengujian PA-08	94
Gambar 5. 9 Hasil Pengujian PA-09	95
Gambar 5. 10 Hasil Pengujian PA-10	97
Gambar 5. 11 Hasil Pengujian PA-11	98
Gambar 5. 12 Hasil Pengujian PA-12	100
Gambar 5. 13 Hasil Pengujian PA-13	101
Gambar 5. 14 Hasil Pengujian PA-14	102
Gambar 5. 15 Hasil Pengujian PA-15	104
Gambar 5. 16 Hasil Pengujian PA-16	106
Gambar 5. 17 Hasil Pengujian PA-17	107
Gambar 5. 18 Hasil Pengujian PA-18	109
Gambar 5. 19 Hasil Pengujian PA-19	111
Gambar 5. 20 Hasil Pengujian PA-22	114
Gambar 5. 21 Hasil Pengujian PA-23	116
Gambar 5. 22 Hasil Pengujian PA-24	117
Gambar 5. 23 Hasil Pengujian PA-25	119
Gambar 5. 24 Proses Pembersihan Memori Dari Task Manager	121
Gambar 5. 25 Notifikasi Foreground Service.....	122

Gambar 5. 26 Tampilan Notifikasi Foreground Service Berjalan	123
Gambar 5. 27 Tampilan Menu Application Manager	124
Gambar 5. 28 Tampilan Clear Master	125
Gambar 5. 29 Notifikasi Foreground Service.....	126
Gambar 5. 30 Tampilan Application Manager.....	127
Gambar 5. 31 Tampilan Saat Akan Dihentikan.....	128
Gambar 5. 32 Tampilan Setelah Dihentikan	129
Gambar 5. 33 Konsumsi Memori < 1 Jam	130
Gambar 5. 34 Konsumsi Memori 2 Jam.....	131
Gambar 5. 35 Konsumsi Memori 7 Jam.....	132
Gambar 5. 36 Konsumsi Memori 9 Jam.....	133



BIODATA PENULIS

Penulis lahir di Kediri, 17 Agustus 1993. Sebelum menempuh jenjang pendidikan sarjana di Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya, penulis menempuh pendidikan di SD Pawyatan Daha 2, SMP Negeri 1 Kediri, dan SMA Negeri 1 Kediri. Dalam menyelesaikan pendidikan S1, penulis mengambil bidang minat algoritma pemrograman. Selama kuliah penulis aktif dalam beberapa kegiatan organisasi.

Pada bidang nonakademik, penulis magang sebagai staff di departemen minat dan bakat Himpunan Mahasiswa Teknik Computer Informatika ITS (HMTC) pada periode kepengurusan 2011-2012. Selama 2 tahun berikutnya penulis aktif sebagai staf dan staf ahli departemen pengembangan sumber daya mahasiswa di HMTC. Penulis juga sempat menjabat wakil koordinator REEVA Schematics 2013. Penulis memiliki ketertarikan pada pembuatan aplikasi perangkat bergerak khususnya berbasis android dan ios, dan sistem *enterprise*. Pada bidang non-teknis penulis memiliki ketertarikan pada bisnis *start-up*, sepak bola, dan travelling. Penulis dapat dihubungi melalui alamat *email* [baskarapatricia17@gmail\[dot\]com](mailto:baskarapatricia17@gmail.com).

BAB I PENDAHULUAN

1.1 Latar Belakang

Saat ini semakin banyak orang tua yang memberikan perangkat bergerak berupa *smartphone* maupun *tablet* kepada anaknya pada usia dini. Hal ini pada awalnya dilakukan untuk kemudahan komunikasi ketika orang tua sedang tidak bersama anaknya. Seiring berjalannya waktu, situs *theAsianparent Indonesia* menuliskan bahwa penggunaan perangkat bergerak pada anak yang berlebihan memiliki dampak negatif sehingga perlu dibatasi. Diantaranya gangguan tidur, terlambat bicara bagi balita, sikap pasif, gangguan bagi perkembangan mata, sulit konsentrasi, dan kecanduan¹. Oleh karena itu, orang tua harus tetap memberikan pengawasan terhadap penggunaan perangkat bergerak milik anaknya sehingga mereka tahu frekuensi kegiatan yang digunakan anak terhadap perangkat bergerak.

Perkembangan seorang anak bergantung sikap orang tua sejak dini, salah satunya adalah dalam memberikan, membimbing, dan mengawasi anak dalam penggunaan media informasi dan teknologi secara terarah dan terpadu. Untuk menentukan sikap orang tua terhadap anaknya secara tepat, diperlukan sebuah *report* pemakaian perangkat bergerak oleh anak secara berkala dan teratur. Dengan *report* ini, orang tua dapat lebih memahami kegiatan anak ketika mengoperasikan perangkat bergerak miliknya sehingga orang tua diharapkan tidak salah dalam mengambil keputusan terhadap perilaku anaknya. Oleh sebab itu, diperlukan aplikasi yang berjalan sebagai *background service* pada sebuah perangkat bergerak yang dapat mencatat seluruh aktivitas pengguna dengan perangkat tersebut. Aplikasi ini harus selalu berjalan menggunakan sumber daya yang

¹ <http://id.theasianparent.com/baikkah-memberikan-tablet-untuk-anak-balita/>

dimiliki perangkat dengan efektif sehingga tidak mengganggu kinerja perangkat ketika digunakan untuk aktivitas lain.

Dalam tugas akhir ini, dibuat aplikasi yang memberikan data serta statistik pemakaian perangkat bergerak untuk mengevaluasi pemakaian perangkat oleh anak. Data-data yang akan diproduksi adalah catatan waktu pemakaian telepon (baik panggilan masuk maupun keluar) beserta nomor yang dituju, pesan singkat masuk dan keluar beserta nomor yang dituju, aplikasi yang dibuka beserta waktu dan lokasinya, dan situs-situs yang dikunjungi oleh *browser* perangkat tersebut. Sementara statistik yang akan dihasilkan dari data yang ada adalah nomor telepon yang paling sering dihubungi dan menghubungi anak melalui panggilan maupun pesan singkat serta aplikasi-aplikasi yang paling sering dibuka oleh anak sehingga orang tua dapat mengawasi apakah anaknya kecanduan terhadap suatu aplikasi. Seluruh *record* beserta statistik pemakaian perangkat yang dihasilkan akan dikirimkan secara berkala dan terjadwal ke alamat *email* yang dimasukkan oleh orang tua. Diharapkan dengan adanya data yang dihasilkan oleh aplikasi ini, orang tua akan lebih mudah menentukan sikap terhadap perilaku anak dalam menggunakan perangkat bergerak.

1.2 Rumusan Permasalahan

Rumusan masalah yang diangkat dalam tugas akhir ini dipaparkan sebagai berikut.

1. Bagaimana cara kerja *service*, *broadcast*, dan *asynchronous task* pada perangkat bergerak.
2. Bagaimana cara untuk membuat *service* selalu berjalan pada latar belakang sistem.
3. Bagaimana cara merancang dan membangun sistem yang mampu memantau aktivitas perangkat bergerak.
4. Bagaimana cara merancang dan membangun sistem yang mampu memproduksi statistik pemakaian perangkat bergerak.

1.3 Batasan Permasalahan

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, di antaranya sebagai berikut.

1. Aplikasi dibangun khusus untuk perangkat bergerak dengan sistem operasi Android dengan versi 4.x.x.
2. Aplikasi membutuhkan koneksi internet untuk mengirimkan *record*.
3. Aplikasi memanfaatkan koneksi internet untuk memperoleh posisi lintang dan bujur.
4. API Google *Geocoding* yang dipakai merupakan versi tidak berbayar.
5. Data alamat situs yang dicatat merupakan alamat situs yang dikunjungi menggunakan *browser native* perangkat.

1.4 Tujuan dan Manfaat

Tujuan dari pembuatan tugas akhir ini adalah sebagai berikut.

1. Mengetahui cara kerja *service*, *broadcast*, dan *asynchronous task* pada android.
2. Merancang cara untuk membuat *service* selalu berjalan di latar belakang sistem.
3. Merancang dan membangun sistem yang mampu memantau aktivitas perangkat bergerak.
4. Merancang dan membangun sistem yang mampu memproduksi statistik pemakaian perangkat bergerak.

Sementara manfaat dari pembuatan tugas akhir ini antara lain sebagai berikut.

1. Memberikan data aktivitas perangkat bergerak secara teratur.
2. Membantu orang tua dalam mengevaluasi kegiatan anak selama menggunakan perangkat bergerak berdasarkan *record* serta hasil statisik yang dikirim ke alamat *email* orang tua.

1.5 Metodologi

Langkah yang ditempuh dalam pengerjaan tugas akhir ini adalah sebagai berikut.

1. Studi Literatur

Tahap ini merupakan tahap pengumpulan informasi dan pembelajaran yang akan digunakan pada tugas akhir ini. Studi literatur meliputi diskusi dan pemahaman mengenai topik tugas akhir ini, diantaranya mengenai:

- a. *Service, Broadcast, dan Asynchronous Task* pada android.
- b. *Content Provider* pada android.
- c. Proses komunikasi data antara aplikasi perangkat perangkat bergerak yang berbasis android dengan *webservice* yang dibangun menggunakan PHP.
- d. Eksplorasi *library* untuk pembuatan diagram pada android.

2. Perancangan Sistem

Tahap ini merupakan tahap perancangan sistem dengan menggunakan studi literatur dan mempelajari konsep aplikasi yang akan dibuat. Dengan berbekal teori, metode, dan informasi yang sudah terkumpul pada tahap sebelumnya diharapkan akan membantu dalam proses perancangan sistem.

3. Implementasi

Tahap ini dilakukan pembuatan perangkat lunak berdasarkan rancangan yang telah dibuat pada proses sebelumnya. Rincian pada tahap ini sebagai berikut.

- a. Implementasi rancangan basis data.
- b. Implementasi *webservice* yang berfungsi untuk menangani komunikasi aplikasi perangkat bergerak dengan *server*.
- c. Implementasi rancangan aplikasi perangkat bergerak.

4. Pengujian dan Evaluasi

Pengujian aplikasi ini dilakukan dengan cara sebagai berikut.

- a. Uji coba pemakaian pada perangkat ketika terhubung dan tidak terhubung dengan internet.
- b. Uji coba penggunaan aplikasi pada lokasi yang berbeda.
- c. Uji coba penggunaan aplikasi pada kondisi *airplane mode*.
- d. Melakukan pengecekan kebenaran data serta statistik yang dihasilkan oleh aplikasi dengan kejadian yang sebenarnya terjadi.
- e. Pengecekan konsumsi aplikasi terhadap memori karena jika berlebihan akan sangat mempengaruhi performa perangkat.

5. Penyusunan Buku Tugas Akhir

Pada tahap ini ditulis buku yang bertujuan untuk mendokumentasikan seluruh konsep, rancangan, dasar teori, literatur, proses yang dilakukan dan hasil yang diperoleh selama pengerjaan tugas akhir. Buku yang ditulis bertujuan untuk memberikan gambaran dari pengerjaan tugas akhir dan berguna untuk pembaca yang tertarik untuk melakukan pengembangan sistem lebih lanjut.

1.6 Sistematika

Buku tugas akhir ini disusun dengan sistematika yang terdiri dari beberapa bagian, yaitu:

Bab I Pendahuluan

Bab ini membahas latar belakang masalah, tujuan pembuatan tugas akhir, rumusan permasalahan, batasan permasalahan, metodologi yang digunakan, dan sistematika penyusunan tugas akhir.

Bab II Tinjauan Pustaka

Bab ini membahas teori pendukung dan literatur yang berkaitan dengan bahasan dan mendasari pembuatan Tugas Akhir ini.

Bab III Analisis dan Perancangan

Bab ini membahas tentang desain dan rancangan dari perangkat lunak.

Bab IV Implementasi

Bab ini membahas tentang implementasi hasil analisis dan perancangan dalam bentuk *pseudocode*. Bab ini membahas proses pembangunan perangkat lunak.

Bab V Pengujian dan Evaluasi

Bab ini membahas tentang pengujian aplikasi berdasarkan skenario yang telah ditentukan. Mengevaluasi fitur aplikasi apakah telah memenuhi kebutuhan fungsional.

Bab V Kesimpulan dan Saran

Bab ini berisi kesimpulan baik dari proses pengembangan perangkat lunak dan hasil uji coba.

Daftar Pustaka

Merupakan daftar referensi yang digunakan dalam pembuatan Tugas Akhir.

BAB II

TINJAUAN PUSTAKA

Pada bab ini dibahas mengenai dasar teori dan literatur yang menjadi dasar pembuatan tugas akhir. Dasar teori yang dibahas di dalam bab ini adalah kajian mengenai sistem operasi android dan *library* MPAndroidChart. Selain itu juga dibahas mengenai basis data SQLite dan struktur data JSON yang digunakan untuk pertukaran data serta beberapa hal lainnya.

2.1 Sistem Pemantau Pemakaian Gadget

Aplikasi pemantau penggunaan perangkat bergerak merupakan aplikasi yang digunakan untuk membantu pengguna memantau pemakaian perangkat bergerak. Bagi orang tua, dengan aplikasi pemantau berarti mengetahui aktivitas anak tanpa memblokir hak aksesnya.

Saat ini ada beberapa aplikasi yang dapat digunakan untuk memantau maupun mengatur pemakaian perangkat bergerak. Situs *tomsguide*, telah merilis beberapa aplikasi pengatur dan pemantau perangkat bergerak anak oleh orang tua yang dianggap terbaik saat ini, antara lain: PhoneSheriff, Qustodio, Net Nanny, My Mobile Watchdog dan Mobile Spy (Hall, 2015). Aplikasi-aplikasi ini memiliki fitur-fitur yang berbeda satu sama lain. Sebagai aplikasi yang dianggap terbaik, PhoneSheriff mampu memantau aplikasi apa saja yang ada di perangkat, pesan singkat yang masuk dan keluar, situs internet yang dikunjungi, serta lokasi anak. Karena konsepnya merupakan aplikasi pengatur oleh orang tua, maka aplikasi ini memungkinkan orang tua untuk memblokir aplikasi lain di perangkat milik anak dan memblokir nomor telepon. Sementara aplikasi lain dari yang disebutkan di atas rata-rata memiliki beberapa fitur yang dimiliki oleh PhoneSheriff, namun tidak selengkap PhoneSheriff.

Aplikasi-aplikasi tersebut tidak dirancang secara khusus untuk memantau aktivitas anak dalam menggunakan perangkat bergerak. Hal ini tercermin pada tidak mampunya aplikasi-

aplikasi tersebut mencatat aplikasi-aplikasi apa saja yang dibuka oleh anak ketika menggunakan perangkat bergerak miliknya.

2.2 Android SDK

Android SDK merupakan perangkat lunak yang digunakan oleh para pengembang untuk membangun aplikasi pada *platform* android. Android SDK menyediakan API yang dibutuhkan oleh pengembang perangkat lunak untuk membuat aplikasi android.

Pada aplikasi yang menjadi tugas akhir ini, eksplorasi dilakukan pada penggunaan *service* dimana kelas yang mengekstensi kelas abstrak *Service* akan menjalankan proses secara terus menerus di balik layar². Normalnya ketika memori pada perangkat tersisa sedikit, sistem operasi akan secara otomatis mematikan servis-servis tersebut. Oleh karena aplikasi pada tugas akhir ini membutuhkan suatu servis yang harus berjalan terus menerus, diperlukan suatu sistem atau cara yang mencegah servis tersebut dihentikan secara paksa oleh sistem operasi ketika memori perangkat tersisa sedikit.

Android memiliki sistem *broadcast* yang dapat digunakan oleh pengembang ingin suatu proses atau komputasi ketika perangkat dalam *state* tertentu³. Untuk mendapatkan suatu *broadcast* dari sistem, pengembang harus mengimplementasikan fungsi khusus pada suatu kelas sehingga kelas tersebut akan menerima suatu *trigger* dari *broadcast* yang dikeluarkan oleh sistem. Perbedaan paling mendasar kelas ini dengan kelas-kelas yang mengekstensi kelas lainnya adalah cara pemanggilannya.

Android juga memiliki sistem yang membuat suatu kelas abstrak bernama *AsyncTask* dimana kelas yang mengekstensi kelas ini akan bersifat seperti servis, namun dirancang agar digunakan untuk mengeksekusi atau melakukan komputasi sebanyak satu kali.

² <http://developer.android.com/reference/android/app/Service.html>

³ <http://developer.android.com/reference/android/content/BroadcastReceiver.html>

2.3 MPAndroidChart

MPAndroidChart⁴ merupakan *library* yang dikembangkan oleh Philipp Jahoda untuk membuat tampilan grafis berupa diagram pada perangkat android. Jenis diagram yang dapat dibuat menggunakan *library* ini antara lain diagram garis, batang, pencar, *candlestick*, pai atau lingkaran, dan jaring laba-laba atau radar. *Library* ini juga mendukung animasi pada diagram, kustomisasi warna dan ukuran sesuai keinginan pengembang aplikasi. Semua aplikasi yang berjalan pada android versi 2.2 ke atas dapat menggunakan fitur-fitur *library* ini.

2.4 Reverse Geocoding

Geocoding merupakan proses melakukan konversi sebuah alamat (misal 1600 Amphitheatre Parkway, Mountain View, CA) menjadi data koordinat geografis (misal lintang 37.423021 dan bujur -122.083739). Sementara *reverse geocoding* merupakan kebalikannya, dimana koordinat geografis diterima sebagai input dan dihitung alamat fisik terdekat dari koordinat tersebut (Stenneth, Wolfson, Yu, & Xu, 2011). Fitur *reverse geocoding* pada tugas akhir ini menggunakan API milik Google dan diimplementasikan pada *webservice* sistem.

2.5 Basis Data SQLite

Basis data SQLite⁵ merupakan basis data yang berukuran kecil, berdiri sendiri, bukan berupa basis data *client-server*, tanpa konfigurasi khusus, namun memiliki fitur penuh perintah SQL. SQLite merupakan produk *public domain*, yang artinya setiap orang dapat menggunakannya secara bebas. Setiap android memakai basis data SQLite, dimana fitur ini terdapat pada *package* android.database.sqlite. Perangkat berbasis android akan memakai SQLite versi 3.4.0.

⁴ <https://github.com/PhilJay/MPAndroidChart>

⁵ <http://www.sqlite.org/about.html>

2.6 *JavaScript Object Notation*

JSON (*JavaScript Object Notation*)⁶ adalah format pertukaran data yang ringan. JSON mudah untuk dibaca dan ditulis oleh manusia dan memudahkan mesin untuk mengurai dan menciptakan. Format ini berdasar dari himpunan bagian bahasa pemrograman JavaScript standar ECMA-262 edisi ke-3, Desember 1999. JSON merupakan format teks yang benar – benar berbeda atau independen namun tetap menggunakan konvensi yang mudah dikenali bagi *programmer* yang sudah terbiasa dengan bahasa C, C++, C#, Java, JavaScript, Perl, Python dan bahasa lainnya. Sifat JSON yang seperti ini menjadikannya format pertukaran data yang ideal.

2.7 *Content Provider*

*Content Provider*⁷ pada android merupakan suatu layanan yang mengatur akses suatu aplikasi ke suatu set data data yang terstruktur. Mereka mengenkapsulasi data tersebut dan membuat suatu mekanisme untuk mengakses isinya agar keamanan data terjaga. *Content Provider* merupakan antarmuka standar yang dapat digunakan oleh pengembang suatu aplikasi untuk melakukan proses mengakses data yang terdapat pada proses lain di luar aplikasi tempat proses itu berjalan.

Ketika pengembang ingin mengakses data pada *content provider*, pengembang dapat menggunakan objek *content resolver* pada aplikasinya untuk melakukan pengiriman data dengan *content provider*. Nantinya, aplikasi akan dianggap sebagai *client* dan *content provider* akan bertindak seperti *server*. *Content provider* ini nanti akan menerima permintaan dari *client*, lalu akan melakukan proses sesuai yang diminta, dan akan mengembalikan hasilnya kepada *client*. Tentunya untuk membaca isi dari *content provider*, pengembang perlu mencantumkan izin apa saja yang dibutuhkan oleh aplikasinya ke dalam *file manifest* milik android.

⁶ <http://json.org/json-id.html>

⁷ <http://developer.android.com/guide/topics/providers/content-providers.html>

2.8 Location Based Services

Location Based Services atau layanan berbasis lokasi adalah sebuah layanan yang dapat diakses oleh perangkat bergerak melalui untuk mendapatkan posisi perangkat secara geografis (Singhal & Shukla, 2012). Layanan berbasis lokasi memanfaatkan lokasi geografis perangkat untuk menyediakan informasi-informasi yang dibutuhkan. Ada beberapa teknologi yang digunakan layanan ini untuk mendapatkan lokasi geografis sebuah perangkat bergerak, salah satunya *Global Positioning System* atau GPS.

GPS merupakan salah satu teknologi yang terkenal akurat untuk digunakan dalam proses pencarian informasi lokasi suatu perangkat. GPS memanfaatkan sinyal dari beberapa satelit untuk melakukan pendeteksian lokasi. Hal ini menyebabkan adanya kelemahan pada teknologi GPS dimana ketika suatu perangkat berada di dalam ruangan, teknologi ini tidak akan bekerja secara maksimal. Hal ini ditandai dengan lamanya suatu perangkat mendapatkan koordinat lintang dan bujurnya ketika perangkat tersebut menggunakan GPS di dalam ruangan.

Pada perangkat android yang memiliki fitur GPS, pengembang umumnya memiliki dua pilihan untuk mendapatkan lokasi perangkat. Yang pertama menggunakan GPS, kedua menggunakan jaringan internet. Seperti yang telah dijelaskan di awal, GPS menghasilkan koordinat lintang dan bujur yang lebih akurat daripada ketika menggunakan jaringan internet⁸. Namun melihat kelemahan GPS ketika digunakan dalam pendeteksian lokasi di dalam ruangan, pencarian lokasi menggunakan jaringan internet lebih sesuai dengan kebutuhan aplikasi tugas akhir ini.

2.9 Webservice

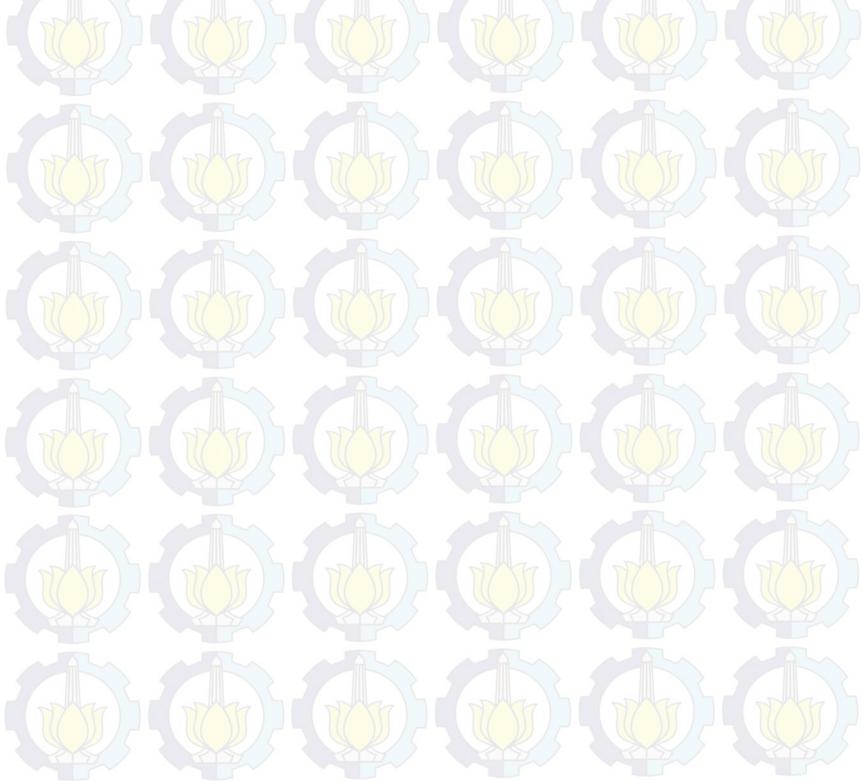
*Webservice*⁹ adalah mekanisme paling standar yang menghubungkan sebuah perangkat lunak dengan perangkat lunak

⁸ <http://stackoverflow.com/questions/1513485/how-do-i-get-the-current-gps-location-programmatically-in-android>

⁹ http://en.wikipedia.org/wiki/Web_service

lain melalui jaringan internet. *Webservice* menggunakan XML untuk melakukan proses *encoding* data melalui protokol HTTP sehingga hampir seluruh *platform* dapat menerima data yang disediakan oleh *webservice*. Selain HTTP dan XML, *webservice* biasanya dibangun dengan dua *platform* tambahan yaitu WSDL dan SOAP.

Dalam tugas akhir ini, *webservice* digunakan untuk menerima data dari *client* yang dikirimkan oleh aplikasi pada perangkat android. Data tersebut diterima dalam bentuk JSON, lalu diolah menjadi tampilan berbentuk tabel. Hasil data olahan ini nantinya dikirimkan ke alamat *email* orang tua.



BAB III

ANALISIS DAN PERANCANGAN

Pada bab ini dibahas analisis kebutuhan, proses, dan desain dari perangkat lunak yang dibangun dalam tugas akhir ini. Bagian awal bab akan dibahas tentang analisis permasalahan dan kebutuhan calon pengguna. Berikutnya dibahas fungsional sistem yang berdasarkan hasil analisis kebutuhan. Bagian berikutnya akan dibahas rancangan perangkat lunak yang ditujukan untuk memberikan gambaran tentang perangkat lunak yang dibuat.

3.1. Analisis

Pada sub bab berikut akan dijelaskan analisis pembuatan aplikasi pemantau pemakaian perangkat bergerak oleh anak. Analisis yang dilakukan meliputi analisis permasalahan, deskripsi umum sistem, dan kebutuhan fungsional sistem.

3.1.1. Analisis Permasalahan

Dewasa ini, semakin banyak orang tua yang memberikan perangkat bergerak kepada anaknya baik berupa ponsel pintar maupun tablet, tidak peduli berapa usia anak. Untuk anak usia balita yang masih belum mengenyam jenjang pendidikan, hal ini dilakukan untuk penenang anak atau semacam *digital babysitter* (Anna, 2011). Sedangkan untuk anak yang telah menginjak jenjang pendidikan formal, hal tersebut dilakukan untuk kemudahan berkomunikasi dengan orang tua maupun teman anak sampai penunjang kegiatan akademik anak.

Namun, dengan semakin banyaknya hiburan serta akses internet nyaris tanpa batas yang ditawarkan oleh perangkat-perangkat bergerak, tidak dapat dipungkiri bahwa perangkat bergerak memiliki ancaman-ancaman kepada anak seperti ketergantungan terhadap *video game*, ketergantungan terhadap media sosial, bahkan memungkinkan anak terlibat dengan tindakan kriminal seperti peredaran narkoba, konsumsi minuman keras, seks bebas, maupun *bullying* yang semestinya bisa dicegah

jika orang tua bisa memantau aktivitas sosial anak dan aktivitas anak ketika menggunakan perangkat bergerak.

3.1.2. Deskripsi Umum Perangkat Lunak

Aplikasi yang dibangun pada tugas akhir ini akan menghasilkan luaran berupa data. Data ini nantinya dapat diolah menjadi hal-hal yang perlu dievaluasi terkait kegiatan anak selama menggunakan perangkat bergerak. Aplikasi ini memiliki kemampuan untuk mencatat aktivitas anak dengan perangkat bergerak miliknya lalu mengirimkan *record* tersebut ke alamat *email* orang tua.

Dalam implementasinya, perangkat lunak ini terdiri dari dua aplikasi. Aplikasi pertama adalah aplikasi yang berjalan di perangkat bergerak android yang nantinya akan dijalankan di perangkat bergerak milik anak. Sedangkan aplikasi kedua merupakan aplikasi *webservice* yang akan menerima data yang dikirim oleh aplikasi pertama, lalu mengolahnya untuk diteruskan ke alamat *email* orang tua. Cara kerja aplikasi pertama ini adalah orang tua cukup melakukan instalasi pada perangkat milik anak, lalu melakukan registrasi menggunakan alamat *email* yang nantinya dijadikan alamat tujuan pengiriman *record* dan menetapkan *password* untuk mengakses aplikasi ini pada perangkat milik anak. Kemudian, orang tua mengatur jam berapa aplikasi akan mengirimkan *email* yang berisi *record* serta statistik pemakaian perangkat oleh anak.

Setiap 24 jam sekali sesuai waktu yang diatur oleh orang tua, aplikasi pertama mengirimkan data-data berupa JSON ke aplikasi kedua. Ketika aplikasi kedua menerima data tersebut, aplikasi kedua akan mengirimkan data-data di JSON tadi ke alamat *email* orang tua. Kebutuhan umum perangkat lunak aplikasi perangkat bergerak apabila diringkas adalah sebagai berikut.

1. Dapat mencatat lokasi perangkat.
2. Dapat mencatat waktu panggilan masuk beserta nomor telepon yang menghubungi.
3. Dapat mencatat waktu panggilan keluar beserta nomor telepon yang dihubungi.

4. Dapat mencatat waktu pesan singkat masuk beserta nomor telepon pengirim dan isi pesan.
5. Dapat mencatat waktu pesan singkat keluar beserta nomor telepon penerima dan isi pesan.
6. Dapat mencatat nama aplikasi yang dibuka beserta waktu saat aplikasi itu dibuka.
7. Dapat mencatat situs *web* yang dikunjungi oleh *browser* perangkat.
8. Dapat menampilkan statistik.
9. Dapat mengubah alamat *email* tujuan penerima *record*.
10. Dapat mengubah *password*.
11. Dapat mengubah waktu *record* dikirimkan ke *webservice*.
12. Dapat mengirimkan *record* ke *webservice*.

Sedangkan aplikasi berbasis *web* sifatnya hanya untuk meneruskan data yang dikirimkan melalui JSON ke alamat *email* tujuan dalam bentuk HTML sehingga dapat dilihat oleh orang tua. Kebutuhan umum perangkat lunak aplikasi *webservice* apabila diringkas adalah sebagai berikut.

1. Dapat menerima data yang dikirimkan oleh perangkat bergerak.
2. Dapat mengirimkan *email* yang berisi *record* perangkat bergerak.

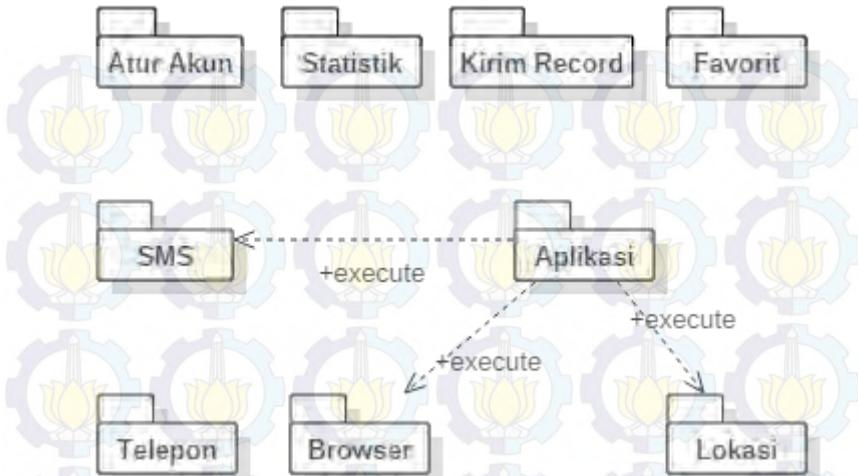
3.1.3. Kebutuhan Fungsional Aplikasi Android

Aplikasi pada perangkat bergerak digunakan untuk menangani proses penyimpanan dan pengolahan data pemakaian perangkat bergerak. Berikut daftar kebutuhan fungsional dari aplikasi pada perangkat bergerak.

1. Menampilkan lokasi perangkat.
2. Menampilkan waktu panggilan masuk.
3. Menampilkan nomor telepon yang menghubungi.
4. Menampilkan waktu panggilan keluar.
5. Menampilkan nomor telepon yang dihubungi.

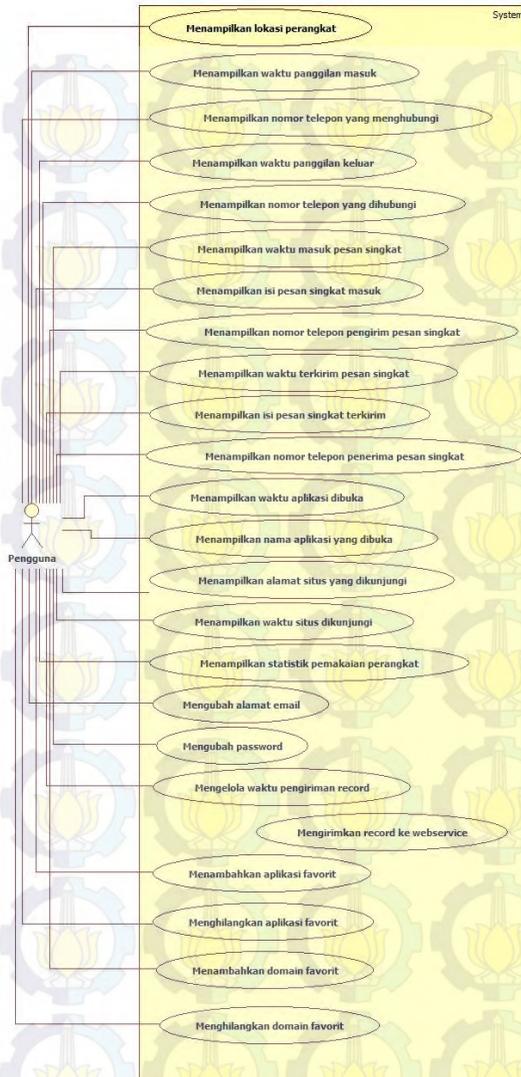
6. Menampilkan waktu masuk pesan singkat.
7. Menampilkan isi pesan singkat masuk.
8. Menampilkan nomor telepon pengirim pesan singkat.
9. Menampilkan waktu terkirim pesan singkat.
10. Menampilkan isi pesan singkat terkirim.
11. Menampilkan nomor telepon penerima pesan singkat.
12. Menampilkan waktu aplikasi dibuka.
13. Menampilkan nama aplikasi yang dibuka.
14. Menampilkan alamat situs yang dikunjungi.
15. Menampilkan waktu situs dikunjungi.
16. Menampilkan statistik pemakaian perangkat.
17. Mengubah alamat *email*.
18. Mengubah *password*.
19. Mengelola waktu pengiriman *record* ke alamat *email* orang tua.
20. Mengirimkan *record* ke *webservice*.
21. Menambahkan aplikasi favorit.
22. Menghilangkan aplikasi favorit.
23. Menambahkan *domain* situs favorit.
24. Menghilangkan *domain* situs favorit.

Dari kebutuhan-kebutuhan fungsional di atas, modul-modul utama yang akan menjadi penyusun aplikasi pada perangkat bergerak adalah sebagai berikut.



Gambar 3. 1 Diagram Modul Aplikasi Perangkat Bergerak

Sementara pada aplikasi *webservice*, hanya terdapat satu modul yaitu modul pengiriman data JSON ke alamat *email*. Sementara penjelasan modul-modul di atas digambarkan lebih spesifik dalam bentuk diagram kasus penggunaan. Terdapat dua diagram kasus penggunaan yang ditampilkan pada sub bab kali ini. Pertama adalah diagram kasus penggunaan aplikasi pada perangkat bergerak. Kedua adalah diagram kasus penggunaan aplikasi pada *webservice*. Diagram kasus penggunaan aplikasi perangkat bergerak ditampilkan sebagai berikut.



Gambar 3. 2 Diagram Kasus Penggunaan Aplikasi Perangkat Bergerak

Hak akses terhadap aplikasi android hanya diberikan kepada orang tua. Penjelasan lengkap mengenai kasus penggunaan pada aplikasi pemantauan perangkat bergerak anak berbasis android ada pada tabel berikut.

Tabel 3. 1 Daftar Kasus Penggunaan Aplikasi Perangkat Bergerak

No	Kode	Nama	Keterangan
1	UC-A01	Menampilkan lokasi perangkat	Orang tua dapat mengetahui lokasi-lokasi perangkat dalam bentuk koordinat lintang bujur.
2	UC-A02	Menampilkan waktu panggilan masuk	Pengguna dapat mengetahui waktu panggilan yang pernah masuk ke perangkat.
3	UC-A03	Menampilkan nomor telepon yang menghubungi	Pengguna dapat mengetahui nomor telepon berapa saja yang menghubungi perangkat milik anak.
4	UC-A04	Menampilkan waktu panggilan keluar	Pengguna dapat mengetahui waktu setiap panggilan keluar yang dilakukan oleh perangkat bergerak.
5	UC-A05	Menampilkan nomor telepon yang dihubungi	Pengguna dapat mengetahui nomor telepon berapa saja yang dihubungi oleh perangkat milik anak.
6	UC-A06	Menampilkan waktu masuk	Pengguna dapat mengetahui waktu setiap

		pesan singkat	pesan singkat yang masuk ke perangkat.
7	UC-A07	Menampilkan isi pesan singkat masuk	Pengguna dapat mengetahui isi setiap pesan singkat yang masuk ke perangkat.
8	UC-A08	Menampilkan nomor telepon pengirim pesan singkat	Pengguna dapat mengetahui nomor telepon pengirim setiap pesan singkat yang masuk ke perangkat.
9	UC-A09	Menampilkan waktu terkirim pesan singkat	Pengguna dapat mengetahui waktu setiap pesan singkat yang dikirim oleh perangkat.
10	UC-A10	Menampilkan isi pesan singkat terkirim	Pengguna dapat mengetahui isi setiap pesan singkat yang dikirim oleh perangkat.
11	UC-A11	Menampilkan nomor telepon penerima pesan singkat	Pengguna dapat mengetahui nomor telepon tujuan setiap pesan singkat yang dikirim oleh perangkat
12	UC-A12	Menampilkan waktu aplikasi dibuka	Pengguna dapat mengetahui waktu setiap aplikasi yang dibuka di perangkat.
13	UC-A13	Menampilkan nama aplikasi	Pengguna dapat mengetahui nama setiap aplikasi yang dibuka di

		yang dibuka	perangkat.
14	UC-A14	Menampilkan alamat situs yang dikunjungi	Pengguna dapat mengetahui alamat situs yang dikunjungi oleh perangkat.
15	UC-A15	Menampilkan waktu situs dikunjungi	Pengguna dapat mengetahui waktu dari alamat situs yang dikunjungi oleh perangkat.
16	UC-A16	Menampilkan statistik pemakaian perangkat	Pengguna dapat mengetahui statistik pemakaian perangkat. Statistik yang dimaksud antara lain: <ul style="list-style-type: none"> • 5 aplikasi yang paling sering dibuka, • 5 nomor telepon yang paling sering dihubungi, • 5 nomor telepon yang paling sering menghubungi perangkat • 5 nomor telepon yang paling sering menerima pesan singkat dari perangkat, dan • 5 nomor telepon yang paling sering mengirim pesan singkat ke perangkat.
17	UC-A17	Mengubah alamat <i>email</i>	Pengguna dapat mengubah alamat <i>email</i> yang dijadikan sebagai sarana

			untuk <i>login</i> dan tujuan pengiriman <i>record</i> pemakaian perangkat.
18	UC-A18	Mengubah <i>password</i>	Pengguna dapat mengubah <i>password</i> yang dijadikan sebagai sarana untuk <i>login</i> .
19	UC-A19	Mengelola waktu pengiriman <i>record</i> ke alamat <i>email</i> orang tua	Pengguna dapat mengatur waktu kapan aplikasi mengirimkan <i>record</i> ke alamat <i>email</i> yang diinginkan.
20	UC-A20	Mengirimkan <i>record</i> ke <i>webservice</i>	Perangkat mampu mengirimkan <i>record</i> ke <i>webservice</i> dalam bentuk JSON.
21	UC-A21	Menambahkan aplikasi favorit	Pengguna dapat menambahkan aplikasi ke dalam daftar aplikasi favorit untuk ditampilkan dengan cara berbeda dari aplikasi lainnya. Hal ini dilakukan bertujuan untuk kemudahan dalam melihat daftar aplikasi yang dibuka pada perangkat.
22	UC-A22	Menghilangkan aplikasi favorit	Pengguna dapat menghapus salah satu aplikasi dari dalam daftar aplikasi favorit.
23	UC-A23	Menambahkan <i>domain</i> situs	Pengguna dapat menambahkan <i>domain</i>

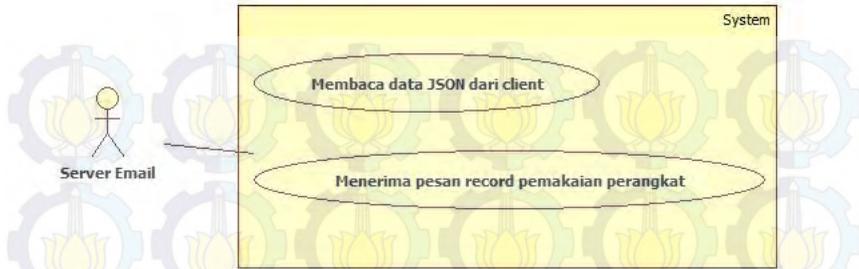
		favorit	situs favorit sehingga setiap situs dengan <i>domain</i> favorit tersebut akan ditampilkan dengan cara berbeda dari situs lainnya. Hal ini juga dilakukan bertujuan untuk kemudahan dalam melihat daftar situs yang dikunjungi perangkat.
24	UC-A24	Menghilangkan <i>domain</i> situs favorit	Pengguna dapat menghapus salah satu <i>domain</i> situs dari dalam daftar <i>domain</i> situs favorit.

3.1.4. Kebutuhan Fungsional Aplikasi *Webservice*

Aplikasi *webservice* ini merupakan aplikasi yang digunakan untuk menerima JSON yang dikirimkan oleh aplikasi pada perangkat bergerak, lalu mengolahnya agar dapat dikirim melalui *email* dan mengirimkannya. Berikut daftar kebutuhan fungsional dari aplikasi *webservice* ini.

1. Membaca data JSON yang dikirim oleh *client*.
2. Mengirim *record* ke alamat *email* yang ditentukan.

Diagram kasus penggunaan aplikasi ini digambarkan sebagai berikut.



Gambar 3. 3 Diagram Kasus Penggunaan Aplikasi Webservice

Oleh karena aplikasi ini sifatnya adalah sebagai *webservice* semata, maka aplikasi ini tidak akan memiliki halaman antarmuka. Berikut penjelasan dari kebutuhan fungsional aplikasi ini.

Tabel 3. 2 Daftar Kasus Penggunaan Aplikasi Webservice

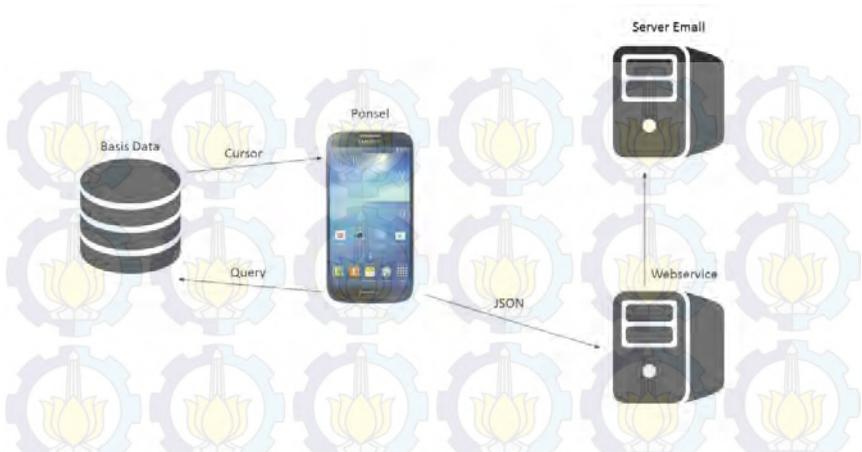
No	Kode	Nama	Keterangan
1	UC-W01	Membaca data JSON yang dikirim oleh <i>client</i>	Aplikasi dapat membaca JSON yang dikirim oleh <i>client</i> lalu melakukan <i>decoding</i> agar nilai-nilai dari JSON tersebut dapat diterjemahkan ke dalam bentuk variabel PHP.
2	UC-W02	Mengirim <i>record</i> ke alamat <i>email</i> yang ditentukan	Aplikasi dapat mengirim sebuah halaman HTML yang berisi <i>record</i> pemakaian perangkat bergerak ke alamat <i>email</i> yang telah ditentukan.

3.2. Perancangan

Sub bab berikut membahas tentang perancangan dari aplikasi pemantau penggunaan perangkat bergerak anak. Sub bab berikut membahas rancangan antarmuka, rancangan proses dan rancangan basis data. Pembahasan lebih detil akan dibahas berikut ini.

3.2.1. Perancangan Umum Arsitektur Sistem

Sistem ini dirancang dengan menggunakan *server* dan *client* sebagai pengumpul dan pengolah data. *Server* berupa *webservice* sedangkan *client* berupa perangkat bergerak android. *Client* bertugas mengumpulkan data berupa catatan penggunaan perangkat bergerak tersebut kemudian ketika waktu pengiriman tiba, *client* akan mengirimkan data tersebut ke *server*. *Server* bertugas menerima data tersebut lalu mengirimkan seluruh data tersebut ke alamat *email* yang dituju. Seluruh data pada *client* disimpan pada basis data lokal yang ada pada masing-masing perangkat. Berikut gambar perancangan arsitektur sistem secara umum.



Gambar 3. 4 Arsitektur Umum Sistem

3.2.2. Perancangan Basis Data

Pada sub bab ini akan dijelaskan bagaimana rancangan basis data yang digunakan pada aplikasi pemantau penggunaan perangkat bergerak. Basis data pada sistem yang dibangun pada tugas akhir ini menggunakan sistem manajemen basis data relasional SQLite. SQLite digunakan untuk penyimpanan data pengguna, pesan singkat, telepon, lokasi, aplikasi, dan situs. Oleh karena tabel-tabel yang dirancang tidak berkaitan dan berelasi satu sama lain, maka rancangan basis data akan langsung dijabarkan sebagai berikut.

3.2.1.1. Rancangan Tabel Pengguna (*user*)

Tabel pengguna digunakan untuk menyimpan data pengguna yang digunakan sebagai akses untuk masuk dalam aplikasi pada perangkat android.

Tabel 3. 3 Rincian Tabel Pengguna

Nama Kolom	Tipe Data	Keterangan
_id	Integer	<i>Primary Key</i> dari

		tabel
email	Text	Alamat pengguna <i>email</i>
password	Text	Kata kunci pengguna

3.2.1.2. Rancangan Tabel SMS Masuk (*incomingSMS*)

Tabel ini digunakan untuk menyimpan data pesan singkat yang masuk ke dalam perangkat.

Tabel 3. 4 Rincian Tabel SMS Masuk

Nama Kolom	Tipe Data	Keterangan
_id	Integer	<i>Primary Key</i> dari tabel
number	Text	Nomor telepon pengirim pesan singkat
time	Text	Waktu pesan singkat masuk
content	Text	Isi pesan singkat
flag	Text	Jenis tabel

3.2.1.3. Rancangan Tabel SMS Keluar (*outgoingSMS*)

Tabel ini digunakan untuk menyimpan data pesan singkat yang dikirim oleh perangkat.

Tabel 3. 5 Rincian Tabel SMS Keluar

Nama Kolom	Tipe Data	Keterangan
------------	-----------	------------

_id	Integer	<i>Primary Key</i> dari tabel
number	Text	Nomor telepon penerima pesan singkat
time	Text	Waktu pesan singkat dikirim
content	Text	Isi pesan singkat
flag	Text	Jenis tabel

3.2.1.4. Rancangan Tabel Telepon Masuk (*incomingCall*)

Tabel ini digunakan untuk menyimpan data panggilan telepon yang masuk ke perangkat.

Tabel 3. 6 Rincian Tabel Telepon Masuk

Nama Kolom	Tipe Data	Keterangan
_id	Integer	<i>Primary Key</i> dari tabel
number	Text	Nomor telepon yang menghubungi perangkat
details	Text	Detail panggilan telepon berupa waktu panggilan masuk sampai waktu panggilan berakhir

3.2.1.5. Rancangan Tabel Telepon Keluar (*outgoingCall*)

Tabel ini digunakan untuk menyimpan data panggilan telepon dari perangkat.

Tabel 3. 7 Rincian Tabel Telepon Keluar

Nama Kolom	Tipe Data	Keterangan
_id	Integer	<i>Primary Key</i> dari tabel
number	Text	Nomor telepon yang dihubungi perangkat
details	Text	Detail panggilan berupa waktu panggilan keluar dimulai sampai waktu panggilan berakhir

3.2.1.6. Rancangan Tabel Aplikasi (*apps*)

Tabel aplikasi digunakan untuk menyimpan data aplikasi yang dibuka oleh perangkat.

Tabel 3. 8 Rincian Tabel Aplikasi

Nama Kolom	Tipe Data	Keterangan
_id	Integer	<i>Primary Key</i> dari tabel
package	Text	Nama aplikasi yang dibuka
time	Text	Waktu aplikasi dibuka

banned	Text	Keterangan apakah aplikasi masuk daftar favorit
duration	Integer	Durasi aplikasi dibuka dalam satuan detik

3.2.1.7. Rancangan Tabel Lokasi (*location*)

Tabel lokasi digunakan untuk menyimpan data lokasi perangkat selama digunakan.

Tabel 3. 9 Rincian Tabel Lokasi

Nama Kolom	Tipe Data	Keterangan
_id	Integer	<i>Primary Key</i> dari tabel
coordinate	Text	Koordinat lokasi berupa lintang dan bujur
time	Text	Waktu pencatatan lokasi
flag	Text	Jenis tabel

3.2.1.8. Rancangan Tabel Situs (*browser*)

Tabel situs digunakan untuk menyimpan situs yang dikunjungi oleh perangkat.

Tabel 3. 10 Rincian Tabel Situs

Nama Kolom	Tipe Data	Keterangan
------------	-----------	------------

_id	Integer	<i>Primary Key</i> dari tabel
time	Text	Waktu dikunjungi situs
url	Text	Alamat situs
title	Text	Judul situs
flag	Text	Jenis tabel

3.2.1.9. Rancangan Tabel Aplikasi Favorit (*bannedApps*)

Tabel ini digunakan untuk mencatat aplikasi-aplikasi apa saja yang dimasukkan ke daftar aplikasi favorit oleh orang tua.

Tabel 3. 11 Rincian Tabel Aplikasi Favorit

Nama Kolom	Tipe Data	Keterangan
_id	Integer	<i>Primary Key</i> dari tabel
package	Text	Nama aplikasi

3.2.1.10. Rancangan Tabel Situs Favorit (*bannedLink*)

Tabel ini digunakan untuk mencatat *domain* situs apa saja yang dimasukkan ke daftar *domain* favorit oleh orang tua.

Tabel 3. 12 Rincian Tabel Situs Favorit

Nama Kolom	Tipe Data	Keterangan
_id	Integer	<i>Primary Key</i> dari tabel

url	Text	Alamat <i>domain</i> situs
-----	------	----------------------------

3.2.1.11. Rancangan Tabel Alarm (*alarm*)

Tabel ini digunakan untuk menyimpan waktu pengiriman *record* ke *webservice*.

Tabel 3. 13 Rincian Tabel Alarm

Nama Kolom	Tipe Data	Keterangan
_id	Integer	<i>Primary Key</i> dari tabel
hour	Text	Jam pengiriman <i>record</i> ke <i>webservice</i>
minute	Text	Menit pengiriman <i>record</i> ke <i>webservice</i>

3.2.1.12. Rancangan Tabel SMS Sementara (*temporarySMS*)

Tabel ini digunakan untuk menyimpan pesan terkirim terakhir dari perangkat. Pesan singkat terkirim terakhir perangkat tidak disimpan pada variabel khusus untuk menghindari terhapusnya isi variabel tersebut oleh sistem.

Tabel 3. 14 Rincian Tabel SMS Sementara

Nama Kolom	Tipe Data	Keterangan
_id	Integer	<i>Primary Key</i> dari tabel
sms	Text	Isi pesan beserta nomor tujuan penerima pesan

3.2.3. Perancangan Algoritma

Perancangan algoritma merupakan tahap untuk membentuk alur proses yang berjalan pada sistem. Berikut perancangan alur kerja aplikasi pada perangkat bergerak secara garis besar.



Gambar 3. 5 Alur Kerja Aplikasi Perangkat Bergerak

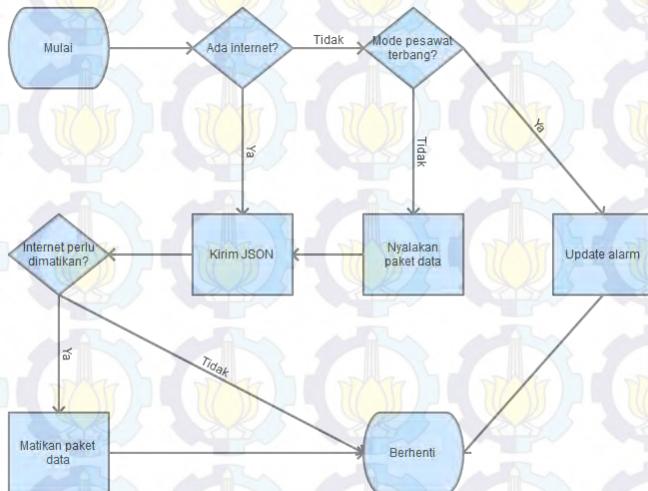
3.2.3.1 Servis Utama

Servis utama ini merupakan inti dari proses pencatatan pemakaian perangkat bergerak. Dari penjelasan data-data yang akan dicatat oleh aplikasi perangkat bergerak ini, ada beberapa jenis data yang bisa didapatkan melalui *broadcast* yang dikirimkan oleh sistem operasi namun ada juga data yang harus didapatkan secara manual dengan membaca isi *content provider*. Data-data yang harus dibaca secara manual melalui isi *content provider* inilah yang harus dikerjakan oleh *service* khusus yang harus dibuat. Data-data tersebut antara lain data aplikasi yang sedang dibuka, data pesan yang dikirimkan, data situs yang dikunjungi, dan data lokasi perangkat berada. Secara garis besar berikut diagram alur servis utama yang berjalan di latar belakang perangkat android.

utama akan menghentikan servis *browser* jika servis tersebut sedang berjalan. Hal ini ditujukan untuk mengantisipasi terjadinya *crash* karena aplikasi menjadi rakus memori.

3.2.3.2 Pengiriman Data

Pada proses pengiriman data, hal yang perlu dipertimbangkan ada dua hal. Pertama, untuk menyiasati ketika anak mematikan akses internet perangkat bergerak saat aplikasi akan mengirimkan JSON ke *webservice*. Kedua, jika perangkat bergerak sedang berada dalam penerbangan pesawat dan menyalakan fitur mode pesawat, aplikasi seharusnya menunda pengiriman JSON ke *webservice*. Oleh sebab itu, aplikasi dirancang agar mengetahui situasi perangkat bergerak tempatnya bernaung. Berikut diagram alur pengiriman data ke *webservice* dengan mempertimbangkan kendala di atas.



Gambar 3. 7 Alur Kerja Proses Pengiriman Data

3.2.4. Perancangan Antarmuka Sistem

Pada subbab ini akan dibahas secara mendetail rancangan antarmuka hanya pada aplikasi perangkat bergerak.

3.2.4.1. Antarmuka Halaman Daftar

Halaman ini berfungsi untuk menampung pengguna yang belum pernah terdaftar pada aplikasi ini. Berikut tampilan antarmuka halaman daftar.



Gambar 3. 8 Rancangan Antarmuka Halaman Daftar

Komponen-komponen penyusun halaman di atas terdapat pada tabel berikut.

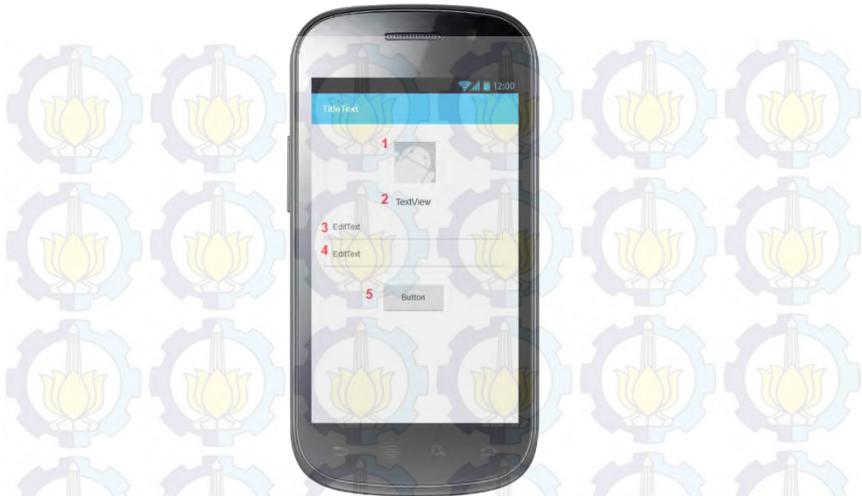
Tabel 3. 15 Keterangan Halaman Antarmuka Daftar

Nomor	Nama Komponen	Keterangan
1	ImageView	Digunakan untuk menampilkan ikon aplikasi.
2	TextView1	Digunakan untuk menampilkan

		nama aplikasi.
3	EditText1	Digunakan sebagai tempat pengisian alamat <i>email</i> pengguna untuk dapat masuk ke aplikasi.
4	EditText2	Digunakan sebagai tempat pengisian <i>password</i> pengguna.
5	EditText3	Digunakan sebagai tempat pengisian ulang <i>password</i> pengguna.
6	Button	Digunakan untuk melakukan proses registrasi ke basis data aplikasi.
7	TextView2	Digunakan untuk menampilkan tulisan pemberitahuan kepada pengguna bahwa masukan alamat <i>email</i> harus benar.

3.2.4.2. Antarmuka Halaman Masuk

Halaman masuk ini berfungsi menangani pengguna yang ingin mengakses aplikasi ketika aplikasi ini memiliki data akun seorang pengguna di dalam basis. Berikut tampilan antarmuka halaman masuk.



Gambar 3. 9 Rancangan Antarmuka Halaman Login

Komponen-komponen penyusun halaman di atas adalah sebagai berikut.

Tabel 3. 16 Keterangan Halaman Antarmuka Login

Nomor	Nama Komponen	Keterangan
1	ImageView	Digunakan untuk menampilkan ikon aplikasi.
2	TextView	Digunakan untuk menampilkan nama aplikasi.
3	EditText1	Digunakan sebagai tempat pengisian alamat <i>email</i> pengguna untuk dapat masuk ke aplikasi.
4	EditText2	Digunakan sebagai tempat pengisian <i>password</i> pengguna.
5	Button	Digunakan untuk melakukan proses pengecekan <i>username</i> dan

		<i>password</i> yang dimasukkan pengguna ke basis data aplikasi.
--	--	--

3.2.4.3. Antarmuka Halaman *Overview*

Halaman *overview* merupakan halaman yang menampilkan seluruh hasil pencatatan yang dilakukan terhadap aktivitas yang dilakukan dengan perangkat. Berikut tampilan antarmuka halaman *overview*.



Gambar 3. 10 Rancangan Antarmuka Halaman Overview

Halaman ini terdiri dari sebuah *spinner* dan sebuah *listview*. Berikut komponen-komponen penyusun halaman *overview*.

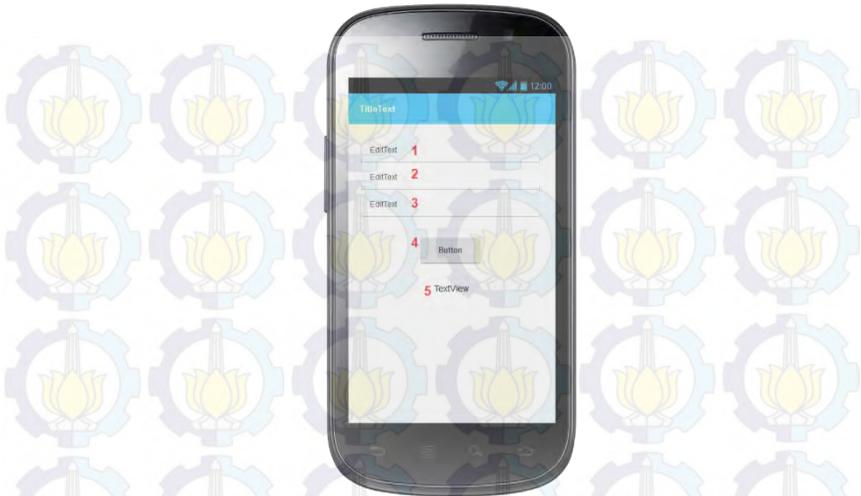
Tabel 3. 17 Keterangan Halaman Antarmuka Overview

Nomor	Nama Komponen	Keterangan
1	Spinner	Digunakan untuk menampilkan

		pilihan data apa saja yang ingin ditampilkan pada <i>listview</i> .
2	Listview	Digunakan untuk menampilkan hasil <i>record</i> yang telah tersimpan pada basis data aplikasi. Pada <i>listview</i> ini berisi sebuah <i>textview</i> dimana <i>textview</i> ini nantinya dijadikan sebagai media untuk menampilkan hasil <i>record</i> .

3.2.4.4. Antarmuka Halaman *Edit Akun*

Halaman *edit* akun merupakan halaman yang digunakan pengguna untuk mengelola akun yang digunakan untuk masuk ke dalam aplikasi. Pada halaman ini terdiri dari kolom isian alamat *email* dan *password* baru yang dikehendaki oleh pengguna jika ingin melakukan perubahan. Berikut tampilan halaman *edit* akun.



Gambar 3. 11 Rancangan Antarmuka Halaman Edit Akun

Berikut komponen-komponen penyusun halaman *edit* akun.

Tabel 3. 18 Keterangan Halaman Antarmuka Edit Akun

Nomor	Nama Komponen	Keterangan
1	EditText1	Digunakan sebagai tempat pengisian alamat <i>email</i> baru pengguna untuk dapat masuk ke aplikasi.
2	EditText2	Digunakan sebagai tempat pengisian <i>password</i> baru pengguna.
3	EditText3	Digunakan sebagai tempat konfirmasi pengisian <i>password</i> baru.
4	Button	Digunakan untuk melakukan

		proses pengubahan alamat <i>email</i> dan <i>password</i> baru pada basis data.
5	TextView	Digunakan untuk menampilkan keterangan pada halaman.

3.2.4.5. Antarmuka Halaman Statistik

Halaman statistik merupakan halaman yang digunakan untuk menampilkan statistik pemakaian perangkat bergerak dalam bentuk diagram. Diagram yang ditampilkan merupakan diagram pai. Berikut rancangan antarmuka halaman statistik.



Gambar 3. 12 Rancangan Antarmuka Halaman Statistik

Komponen-komponen penyusun halaman statistik ada dalam tabel berikut.

Tabel 3. 19 Keterangan Halaman Antarmuka Statistik

Nomor	Nama Komponen	Keterangan
1	Spinner	Digunakan untuk menampilkan pilihan data statistik apa saja yang ingin ditunjukkan pada <i>piechart</i> .
2	PieChart	Digunakan untuk menampilkan statistik pemakaian perangkat bergerak. Bagian ini terdiri dari sebuah gambar diagram dan sebuah legenda yang menjelaskan maksud dari warna-warna yang ada pada diagram pai tersebut.

3.2.4.6. Antarmuka Halaman Atur Jadwal

Halaman atur jadwal merupakan halaman yang digunakan untuk menangani kebutuhan pengguna dalam mengatur jadwal kapan *email* yang berisi *record* pemakaian perangkat bergerak dikirimkan ke alamat *email* orang tua. Berikut rancangan antarmuka halaman atur jadwal.



Gambar 3. 13 Rancangan Antarmuka Halaman Atur Jadwal

Komponen-komponen penyusun halaman ini antara lain sebagai berikut.

Tabel 3. 20 Keterangan Halaman Antarmuka Atur Jadwal

Nomor	Nama Komponen	Keterangan
1	TextView	Digunakan untuk menampilkan tulisan yang menerangkan tentang fungsi halaman.
2	DatePicker	Digunakan sebagai media pengguna untuk menentukan jam berapa <i>record</i> dikirimkan oleh aplikasi ke alamat <i>email</i> pengguna.
3	Button	Digunakan sebagai proses untuk menyimpan waktu pengiriman <i>record</i> .

3.2.4.7. Antarmuka Halaman Menambahkan Favorit

Halaman ini merupakan halaman yang digunakan untuk menangani kebutuhan pengguna dalam menandai aplikasi-aplikasi serta *domain-domain* situs mana saja yang perlu ditandai menjadi aplikasi dan *domain* favorit. Halaman ini nantinya menampilkan seluruh aplikasi dan situs yang telah diakses oleh anak sehingga orang tua tinggal memilih aplikasi dan situs dengan *domain* apa yang dianggap perlu diperhatikan. Berikut rancangan antarmuka halaman ini.



Gambar 3. 14 Rancangan Antarmuka Halaman Menambahkan Favorit

Komponen-komponen penyusun halaman ini antara lain sebagai berikut.

Tabel 3. 21 Keterangan Halaman Antarmuka Menambahkan Favorit

Nomor	Nama Komponen	Keterangan
1	Spinner	Digunakan untuk menampilkan pilihan jenis favorit yang ingin ditambahkan. Nantinya <i>spinner</i> ini jika ditekan akan memunculkan sebuah fragmen <i>popup</i> yang berisi pilihan aplikasi dan <i>domain</i> .
2	ListView	Digunakan sebagai media untuk menampilkan daftar calon aplikasi atau situs yang bisa ditambahkan ke dalam objek favorit oleh pengguna. <i>Listview</i> ini nantinya berisi gabungan <i>checkbox</i> dan <i>textview</i> .
3	Button	Digunakan sebagai proses untuk menyimpan daftar konten-konten yang ditandai sebagai objek favorit.

3.2.4.8. Antarmuka Halaman Menghapus Favorit

Halaman ini merupakan halaman untuk menangani kebutuhan pengguna dalam menghilangkan aplikasi atau *domain* dari daftar favorit yang dimiliki. Berikut tampilan rancangan antarmuka dari halaman menghapus favorit.



Gambar 3. 15 Rancangan Antarmuka Halaman Menghapus Favorit

Komponen-komponen penyusun antarmuka halaman ini antara lain sebagai berikut.

Tabel 3. 22 Keterangan Halaman Antarmuka Menghapus Favorit

Nomor	Nama Komponen	Keterangan
1	Spinner	Digunakan untuk menampilkan pilihan jenis favorit yang ingin dihapus dari daftar. Nantinya <i>spinner</i> ini jika ditekan akan memunculkan sebuah fragmen <i>popup</i> yang berisi pilihan aplikasi dan <i>domain</i> .
2	ListView	Digunakan sebagai media untuk menampilkan daftar calon aplikasi atau <i>domain</i> situs yang bisa dihapus dari dalam daftar objek

		favorit oleh pengguna. <i>Listview</i> ini nantinya berisi gabungan <i>checkbox</i> dan <i>textview</i> .
3	Button	Digunakan sebagai proses untuk menyimpan perubahan isi dari daftar konten yang ditandai sebagai objek favorit.

BAB IV IMPLEMENTASI

Pada bab ini dijelaskan mengenai implementasi dari perancangan perangkat lunak. Implementasi yang dijelaskan meliputi lingkungan pembangunan perangkat lunak, implementasi antarmuka pengguna, dan implementasi proses-proses yang terjadi pada perangkat lunak. Implementasi sistem mengacu pada perancangan yang ditulis pada Bab III.

4.1. Lingkungan Implementasi

Dalam merancang perangkat lunak ini digunakan beberapa perangkat pendukung. Perangkat keras yang digunakan dalam pengembangan sistem ini berupa komputer dengan spesifikasi AMD Phenom II X6 1090T 3,20 Ghz dengan memori 8GB. Sementara spesifikasi perangkat lunak yang digunakan untuk pengembangan sistem adalah sebagai berikut :

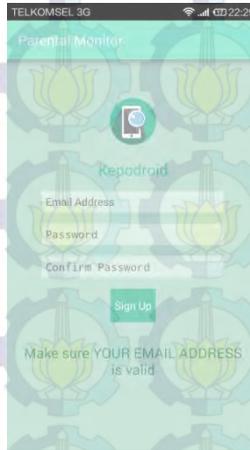
- Microsoft Windows 7 64bit sebagai sistem operasi.
- Android Studio 1.1.0 sebagai IDE untuk implementasi aplikasi.
- Android SDK versi 24.1.2 sebagai kerangka aplikasi perangkat bergerak.
- Apache versi 2.4.12 dan PHP versi 5.4.41 sebagai lingkungan web server aplikasi *webservice*.

4.2. Implementasi Antarmuka

Pada subbab ini akan dibahas implementasi antarmuka berdasarkan rancangan antarmuka yang telah dibahas pada bab sebelumnya. Antarmuka yang akan dibahas merupakan antarmuka pada aplikasi perangkat bergerak. Berikut rincian rancangan antarmuka pada aplikasi perangkat bergerak.

- a. **Halaman daftar** merupakan halaman yang akan terbuka pertama kali ketika pengguna baru saja melakukan instalasi

aplikasi ke perangkat bergerak. Halaman ini digunakan oleh pengguna untuk mendaftarkan alamat *email* beserta *password* yang diinginkan untuk digunakan sebagai alat masuk ke aplikasi. Berikut implementasi dari halaman daftar pada aplikasi perangkat bergerak.



Gambar 4. 1 Antarmuka Halaman Daftar

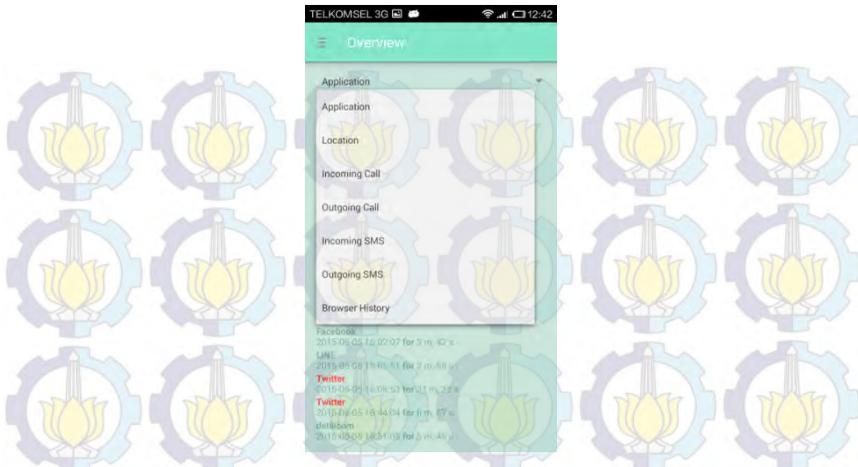
- b. **Halaman masuk** merupakan halaman yang akan terbuka pertama kali ketika pengguna membuka aplikasi, lalu aplikasi mendeteksi bahwa telah ada pengguna yang terdaftar sebelumnya. Berikut implementasi dari halaman masuk pada aplikasi perangkat bergerak.



Gambar 4.2 Antarmuka Halaman Login

Pada gambar di atas, pengguna diminta untuk memasukkan alamat *email* dan *password* untuk masuk ke aplikasi. Jika terjadi kesalahan, maka aplikasi akan menampilkan *toast* yang menginformasikan bahwa alamat *email* atau *password* yang dimasukkan salah.

- c. **Halaman *overview*** merupakan halaman yang akan terbuka setelah pengguna melewati halaman daftar maupun halaman masuk. Halaman ini berfungsi untuk menampilkan hasil pemantauan terhadap aktivitas perangkat bergerak. Pada gambar di bawah ditampilkan bahwa tampilan *default* dari antarmuka ini adalah tampilan daftar aplikasi yang dibuka pada perangkat bergerak. Daftar *record* ini ditampilkan pada sebuah *listview* yang berisi sekumpulan *textview*. Di atas *listview* ini, ada sebuah *spinner* yang menampilkan pilihan jenis *record* yang bisa dipilih pengguna untuk ditampilkan pada halaman. Berikut tampilan antarmuka halaman *overview*.



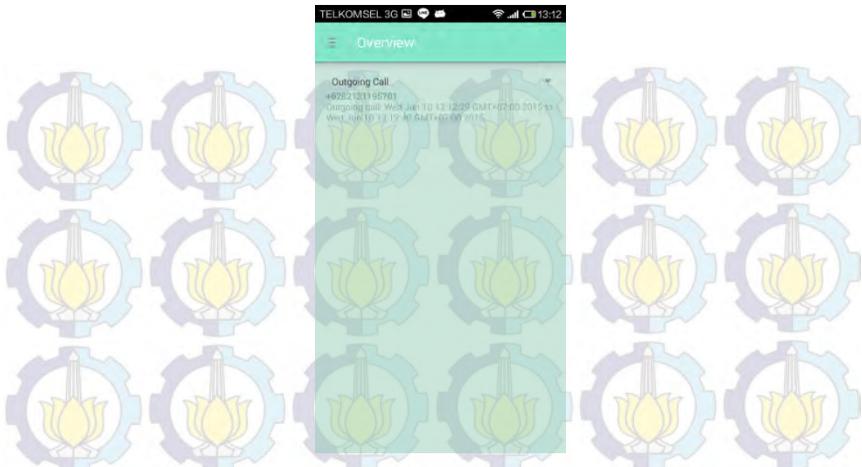
Gambar 4. 3 Antarmuka Halaman Overview

- Halaman antarmuka berikut adalah tampilan ketika pengguna memilih *application* pada *spinner*.



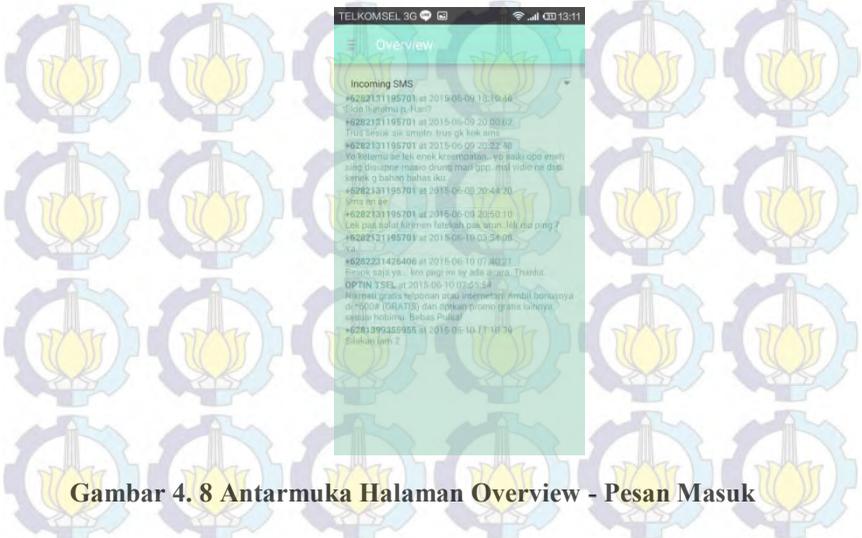
Gambar 4. 4 Antarmuka Halaman Overview – Aplikasi

- Halaman antarmuka berikut adalah tampilan ketika pengguna memilih *location* pada *spinner*.



Gambar 4.7 Antarmuka Halaman Overview - Telepon Keluar

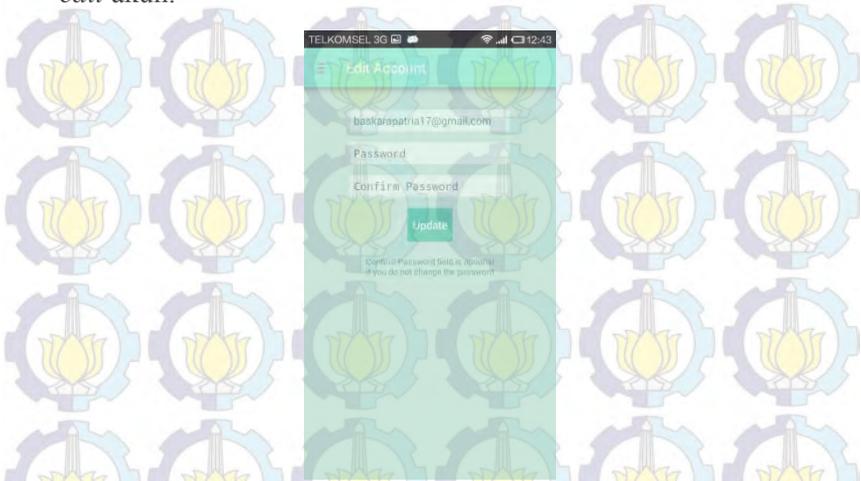
- Halaman antarmuka berikut adalah tampilan ketika pengguna memilih *incoming SMS* pada *spinner*.



Gambar 4.8 Antarmuka Halaman Overview - Pesan Masuk

- Halaman antarmuka berikut adalah tampilan ketika pengguna memilih *outgoing SMS* pada *spinner*.

email maupun *password*. Berikut tampilan antarmuka halaman *edit* akun.



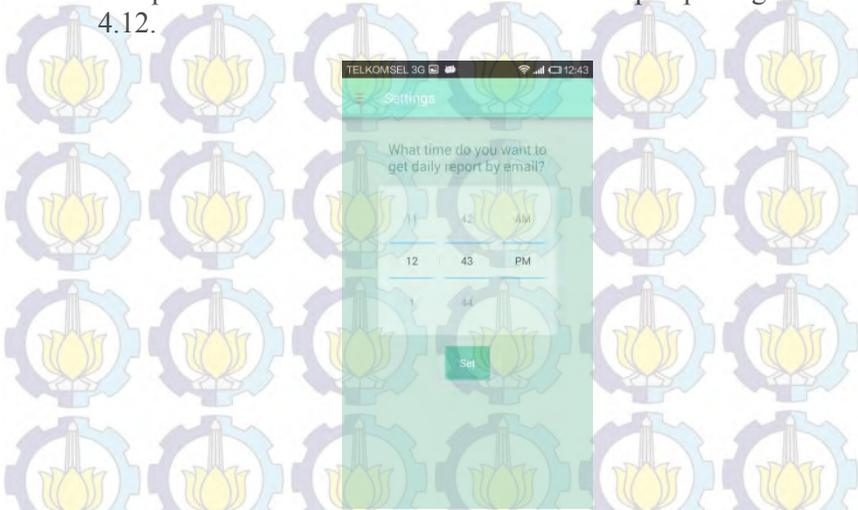
Gambar 4. 11 Antarmuka Halaman Edit Akun



Gambar 4. 12 Antarmuka Halaman Statistik

- e. **Halaman statistik** merupakan halaman yang menampilkan hasil statistik aktivitas perangkat bergerak. Diagram pai yang

ada di antarmuka ini memakai *library* dari MPAndroidChart. Tampilan antarmuka halaman statistik terdapat pada gambar 4.12.



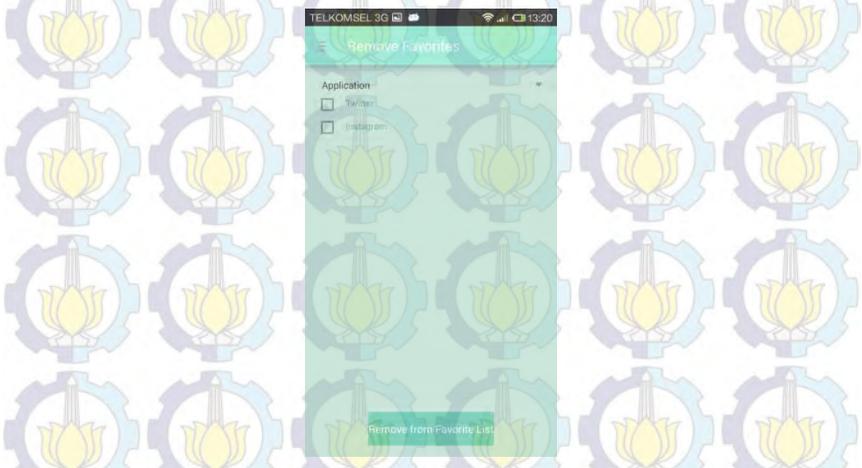
Gambar 4. 13 Antarmuka Halaman Atur Jadwal

f. **Halaman atur jadwal** merupakan halaman yang digunakan pengguna untuk mengatur waktu pengiriman *record* ke alamat *email* yang terdaftar. tampilan antarmuka halaman atur jadwal terdapat pada gambar 4.13.

g. **Halaman menambahkan favorit** merupakan halaman yang digunakan pengguna untuk memilih aplikasi atau *domain* apa saja yang ingin ditambahkan ke daftar favorit. Aplikasi maupun *domain* yang ditambahkan ke daftar favorit akan ditampilkan dengan warna merah pada halaman *overview*. Berikut tampilan antarmuka halaman menambahkan favorit.



Gambar 4. 14 Antarmuka Halaman Menambah Favorit



Gambar 4. 15 Antarmuka Halaman Menghapus Favorit

- h. **Halaman menghapus favorit** merupakan halaman yang digunakan untuk menghilangkan aplikasi atau *domain* yang sebelumnya ada pada daftar favorit. Ketika aplikasi atau *domain* ini dihilangkan dari daftar favorit, maka aplikasi dan

situs dengan *domain* tadi akan ditampilkan dengan warna normal. Tampilan antarmuka halaman menghapus favorit ditunjukkan pada gambar 4.15.

4.3. Implementasi Proses Pada Aplikasi Perangkat Bergerak

Pada subbab ini akan dijabarkan implementasi dari proses-proses yang ada pada aplikasi yang berjalan pada perangkat bergerak. Proses yang ada meliputi pencatatan aktivitas, penampilan *record*, pembuatan dan penampilan statistik, sampai pengiriman *record* ke *webservice*.

4.3.1. Mencatat Aplikasi yang Dibuka

Berikut ini merupakan implementasi proses pencatatan aplikasi yang dibuka oleh perangkat bergerak. Proses ini memanfaatkan kelas bertipe *service* yang berjalan terus menerus di latar belakang perangkat. Ketika pertama kali *service* ini dijalankan, secara otomatis fungsi *onCreate()* dipanggil dan dilanjutkan ke fungsi *onStartCommand()*. Fungsi-fungsi ini merupakan *override* dari kelas *Service*. Berikut *pseudocode* implementasi dari fungsi *onCreate()*.

```
PUBLIC VOID onCreate() {
    inialisasi ActivityManager;
    inialisasi databaseConnector;
}
```

ActivityManager di atas diinisialisasi untuk digunakan dalam berinteraksi dengan status global dari perangkat. Sedangkan *database connector* merupakan kelas yang berfungsi untuk menghubungkan aplikasi dengan basis data SQLite. Secara otomatis, ketika fungsi di atas selesai, maka proses akan berlanjut ke fungsi *onStartCommand()*. Berikut *pseudocode* implementasi dari fungsi tersebut.

```
PUBLIC INT onStartCommand(Intent intent, int flag, int
startId) {
    START servis lokasi
    IF servis sms tidak nyala THEN
        START servis sms
    END IF
```

```

REGISTER screen listener
SET launcherName
STOP servis lokasi
START foreground service
RETURN START_STICKY
}

```

Kelas servis yang dibuat mengimplementasikan kelas *interface Runnable* sehingga ketika servis akan mengeksekusi fungsi *run()* secara otomatis. Pada fungsi ini, proses dilakukan dengan cara mengecek apakah layar perangkat menyala atau tidak. Jika tidak, maka proses pengecekan aplikasi yang sedang berjalan tidak dilakukan. Jika iya, maka sebaliknya. Setelah itu, proses mengecek apakah yang sedang terbuka merupakan *launcher* perangkat bergerak. Jika iya, maka *looping* akan dilanjutkan ke iterasi berikutnya. Kemudian, proses akan mengecek apakah nama *package* aplikasi yang sedang dibuka tidak sama dengan nama *package* aplikasi yang terakhir dibuka ketika proses mencatat pada iterasi sebelumnya. Jika iya, maka data aplikasi yang terakhir dimasukkan dahulu ke basis data. Lalu, data nama dan waktu dibukanya aplikasi yang saat ini terbuka disimpan pada sebuah variabel yang nantinya dipanggil ketika akan dimasukkan ke basis data. Selanjutnya, proses akan memanggil servis lokasi untuk mendapatkan lokasi perangkat pada saat ini. Berikut *pseudocode* dari implementasi fungsi *run()*.

```

PUBLIC VOID run() {
    WHILE isRunning = true
        SLEEP 1 detik
        IF screenOn = true THEN
            nowApp = nama package top activity
            IF nowApp = launcherName THEN
                IF lastApp != launcherName THEN
                    lastApp = launcherName
                    insertApps ()
                END IF
            CONTINUE
        END IF
        IF nowApp != nama package app ini THEN
            IF nowApp != lastApp THEN
                insertApps ()
            END IF
        END IF
    END WHILE
}

```

```

        lastApp = nowApp
        IF nowApp = browser THEN
            START servis browser
        ELSE
            url = ""
            IF servis browser berjalan THEN
                STOP servis browser
            END IF
        END IF
        appName = getLabel(nowApp)
        appTime = getTimeStamp()
        IF bisa akses lokasi THEN
            START servis lokasi
            SLEEP 1 detik
            simpan lokasi ke basis data
            STOP servis lokasi
        END IF
    END IF
END IF
END IF
END WHILE
}

```

Berikut *pseudocode* dari fungsi *insertApps()* yang digunakan untuk memasukkan data aplikasi ke basis data.

```

PUBLIC VOID insertApps() THROWS ParseException {
    IF appName != null && appTime != null THEN
        now = getCurrentTimeStamp()
        format = new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss")
        date1 = format.parse(appTime)
        date2 = format.parse(now)
        difference = date2.getTime() - date1.getTime()
        duration = difference / 1000
        IF lastAppTime != appTime THEN
            lastAppTime = appTime
            databaseConnector.addApps(appName, appTime,
duration)
        END IF
    END IF
}

```

Setiap melakukan pengisian ke basis data, dilakukan pengecekan apakah data aplikasi yang akan dimasukkan termasuk aplikasi yang ada dalam daftar favorit. Berikut *query* beserta fungsi yang dijalankan untuk memeriksa apakah aplikasi termasuk ke daftar aplikasi favorit.

```

PUBLIC BOOLEAN cekBannedApps(String name) {
    sql = "SELECT COUNT(*) FROM " + TABLE_BANNEDAPPS + "
WHERE package = '" + name + "'"
    cursor = database.rawQuery(sql, null)
    cursor.moveToFirst()
    count = cursor.getInt(0)
    cursor.close()
    IF count > 0 THEN
        RETURN TRUE
    ELSE
        RETURN FALSE
    END IF
}

```

Oleh karena proses ini seharusnya berjalan terus menerus, maka ketika pengguna dengan sengaja mematikan servis ini, servis akan mengirimkan *broadcast* agar servis ini dinyalakan kembali. Berikut implementasi dari proses pengiriman *broadcast* yang dilakukan oleh fungsi *onDestroy()*.

```

PUBLIC VOID onDestroy() {
    UNREGISTER screen listener
    isRunning = false
    intent = new Intent("restartApps")
    SEND BROADCAST intent
}

```

Broadcast “restartApps” akan diterima oleh kelas lain yang mengekstensi *BroadcastReceiver* lalu kelas tersebut akan memulai kelas servis utama ini lagi.

4.3.2. Mencatat Lokasi Pengguna

Proses pencatatan lokasi pengguna dilakukan dengan menggunakan sebuah kelas yang berjalan sebagai servis dan mengimplementasikan kelas *interface LocationListener*. Berikut *pseudocode* dari fungsi *onStartCommand()* yang dijalankan pertama setelah servis lokasi ini dimulai.

```

PUBLIC INT onStartCommand(Intent intent, int flag, int
startId) {
    inisialisasi locationManager
    SET tipe request NETWORK_PROVIDER
    RETURN START_NOT_STICKY
}

```

Kemudian dengan fungsi *onLocationChanged()* yang diimplementasi dari kelas *interface*, setiap perubahan lokasi yang dideteksi oleh perangkat akan disimpan ke variabel global lintang dan bujur. Hal ini dilakukan karena proses pengisian data lokasi dilakukan pada servis utama. Berikut *pseudocode* fungsi *onLocationChanged()*.

```
PUBLIC VOID onLocationChanged(Location location) {
    latitude = location.getLatitude()
    longitude = location.getLongitude()
    REMOVE update lokasi
}
```

4.3.3. Mencatat Panggilan Telepon

Proses pencatatan panggilan telepon ini diimplementasikan dengan cara membuat sebuah kelas abstrak yang mengekstensi *BroadcastReceiver*. Kelas ini memiliki sebuah fungsi *onReceive()* dimana pada fungsi ini setiap proses *state* telepon yang masuk atau keluar akan dicatat. Berikut *pseudocode* dari fungsi *onReceive()*.

```
PUBLIC VOID onReceive(Context context, Intent intent) {
    IF broadcast diterima = "NEW_OUTGOING_CALL" THEN
        simpan nomor telepon
    ELSE
        stateStr = GET extra state
        number = GET extra number
        int state = 0;
        IF stateStr = "EXTRA_STATE_IDLE" THEN
            state = "CALL_STATE_IDLE"
        ELSE IF stateStr = "EXTRA_STATE_OFFHOOK" THEN
            state = "CALL_STATE_OFFHOOK"
        ELSE IF stateStr = "EXTRA_STATE_RINGING" THEN
            state = "CALL_STATE_RINGING"
        END IF
        onCallStateChanged(context, state, number)
    END IF
}
```

Kemudian pada fungsi *onCallStateChanged()* dilakukan pengecekan apakah jenis perubahan *state* yang terjadi termasuk telepon masuk atau keluar. Jika telepon masuk, apakah jenisnya

missed call atau telepon terangkat. Berikut *pseudocode* dari fungsi *onCallStateChanged()*.

```

PUBLIC VOID onCallStateChanged(Context context, int state,
String number) {
    IF lastState = state THEN
        RETURN
    END IF
    CASE state OF
        CALL_STATE_RINGING:
            isIncoming = true
            time = new Date()
            savedNumber = number
            onIncomingCallStarted(context, number, time)
        CALL_STATE_OFFHOOK:
            IF lastState != "CALL_STATE_RINGING" THEN
                isIncoming = false
                time = new Date()
                onOutgoingCallStarted(context,
savedNumber, time)
            END IF
        CALL_STATE_IDLE:
            IF lastState = "CALL STATE RINGING" THEN
                onMissedCall(context, savedNumber, time)
            ELSE IF isIncoming = true THEN
                onIncomingCallEnded(context,
savedNumber, time, new Date())
            ELSE
                onOutgoingCallEnded(context,
savedNumber, time, new Date())
            END IF
    END CASE
    lastState = state
}

```

Ketika proses mendeteksi perubahan *state* dari RINGING menjadi IDLE, berarti terjadi *missed call* yang masuk ke perangkat. Jika ada perubahan *state* dari RINGING ke OFFHOOK lalu menjadi IDLE, berarti telepon yang masuk ke perangkat diangkat oleh pengguna. Selain itu, *state* yang masuk dianggap sebagai aksi panggilan keluar. Setiap *state* IDLE maka fungsi di atas memanggil fungsi untuk melakukan proses pengisian ke basis data berdasarkan jenis panggilan yang terjadi.

4.3.4. Mencatat Pesan Singkat Masuk

Proses mencatat pesan singkat masuk memanfaatkan *broadcast* yang dikeluarkan sistem operasi ketika ada pesan singkat masuk. Kelas ini mengekstensi kelas abstrak *BroadcastReceiver* sehingga untuk mendapatkan *broadcast* pesan masuk, cukup mengimplementasikan fungsi *onReceive()*. Berikut *pseudocode* fungsi *onReceive()*.

```
PUBLIC VOID onReceive(Context context, Intent intent) {
    IF intent.getAction() = "SMS_RECEIVED" THEN
        handleMessage(intent.getExtras(), context)
    END IF
}
```

Pada fungsi *handleMessage()* dilakukan proses pemilahan isi dari *intent* yang dikirimkan. *Intent* ini diterima dalam bentuk *Bundle*, lalu dalam *bundle* tersebut diambil objek PDU di dalamnya. PDU merupakan singkatan dari *Protocol Data Unit*, berupa suatu format objek yang digunakan untuk mengirimkan data pesan singkat melalui jaringan seluler. Dari objek PDU tersebut, nantinya didapatkan informasi nomer asal pesan singkat dan isi pesan singkat yang baru saja masuk. Berikut *pseudocode* implementasi fungsi *handleMessage()*.

```
PUBLIC VOID handleMessage(Bundle bundle, Context context) {
    inisialisasi databaseConnector
    IF bundle != null THEN
        pdus = (Object[]) bundle.get("pdus")
        FOR Object pdu IN pdus
            msg = SmsMessage.createFromPdu((byte[]) pdu)
            number = msg.getOriginatingAddress()
            body = msg.getMessageBody()
            simpan ke basis data
        END FOR
    END IF
}
```

4.3.5. Mencatat Pesan Singkat Keluar

Pada proses pencatatan data pesan singkat keluar, dibuat sebuah servis yang berjalan terus menerus. Hal ini dikarenakan untuk menangkap pesan singkat yang dikirimkan oleh perangkat,

sistem operasi tidak menyediakan *broadcast* yang bisa ditangkap oleh aplikasi eksternal. Servis yang dibuat ini dimulai dengan fungsi *onStartCommand()* yang berisi *register Content Observer* untuk membaca isi *Content Provider*. Berikut implementasi dari fungsi ini.

```
PUBLIC INT onStartCommand(Intent intent, int flag, int
startId) {
    inialisasi databaseConnector
    inialisasi smsObserver
    contentResolver = getContentResolver()
    registerContentObserver(Uri.parse
("content://sms/sent"), true, smsObserver)
    RETURN START_STICKY
}
```

SMSObserver merupakan kelas yang mengekstensi *ContentObserver* sehingga memiliki sebuah fungsi *onChange()*. Fungsi ini akan dipanggil setiap ada perubahan isi dari *Content Provider* yang telah di-*register* melalui *Content Resolver*. Berikut *pseudocode* fungsi *onChange()* pada kelas *SMSObserver*.

```
PUBLIC VOID onChange(booleam selfChange) {
    uri = Uri.parse("content://sms/sent")
    cur = getContentResolver().query(uri, null,
null, null, null)
    cur.moveToNext()
    content = cur.getString(kolom "body")
    smsNumber = cur.getString(kolom "address")
    smsDate = cur.getString(kolom "date")
    IF smsNumber = null || smsNumber.length < 1 THEN
        smsNumber = "Unknown"
    END IF
    cur.close()
    IF sms telah disimpan ke basis data THEN
        simpan ke basis data
    END IF
}
```

Sebelum dilakukan penyimpanan ke basis data, perlu dicek apakah data pesan singkat yang akan disimpan telah disimpan sebelumnya. Hal ini dilakukan karena pada kenyataannya, fungsi *onChange()* ini kadang terpanggil lebih dari satu kali ketika terdeteksi adanya perubahan isi dari *Content Provider*. Servis ini dijalankan secara terus menerus, sehingga ketika proses

mendeteksi bahwa servis dimatikan secara paksa oleh pengguna, maka proses akan mengirim *broadcast* agar servis ini dinyalakan kembali oleh sistem. Proses pengiriman *broadcast* hampir sama dengan yang dilakukan oleh servis utama.

4.3.6. Mencatat Alamat Situs yang Dikunjungi

Proses pencatatan alamat situs yang dikunjungi relatif sama dengan proses pencatatan pesan singkat keluar, yaitu memanfaatkan sebuah servis yang mengolah data dari *Content Observer*. Bedanya, servis ini dijalankan hanya jika servis utama mendeteksi bahwa pengguna sedang membuka aplikasi *browser native* milik perangkat. Berikut implementasi *pseudocode* dari fungsi *onStartCommand()*.

```
PUBLIC INT onStartCommand(Intent intent, int flag, int
startId) {
    inisialisasi databaseConnector
    inisialisasi browserObserver
    registerContentObserver(Browser.BOOKMARKS_URI, true,
browserObserver)
    RETURN START_STICKY
}
```

Kelas *BrowserObserver* memiliki sebuah fungsi implementasi yaitu fungsi *onChange()* yang dipanggil setiap ada perubahan pada isi *Content Provider* yang telah di-*register*. Berikut *pseudocode* fungsi *onChange()*.

```
PUBLIC VOID onChange(boolean selfChange) {
    proj = new String[] {
    Browser.BookmarkColumns.TITLE, Browser.BookmarkColumns.URL,
    Browser.BookmarkColumns.DATE }
    selection = Browser.BookmarkColumns.BOOKMARK + "
= 0"
    cur =
getContentResolver().query(Browser.BOOKMARKS_URI, proj,
selection, null, null)
    cur.moveToLast()
    url = cur.getString(1)
    date = cur.getString(2)
    IF url belum dimasukkan basis data THEN
        SLEEP 1 detik
        title = cur.getString(0)
        simpan ke basis data
```

```

        END IF
        cur.close()
    }

```

Sama seperti setiap fungsi *onChange()* sebelumnya, fungsi ini kadang dipanggil lebih dari satu kali ketika sistem mendeteksi adanya perubahan isi dari *Content Provider*. Sehingga diperlukan pengecekan apakah situs yang baru saja terdeteksi sudah dimasukkan ke dalam basis data atau belum sebelum melakukan penyimpanan. Pada saat melakukan proses penyimpanan ke basis data, ketika halaman belum ter-load sepenuhnya, judul situs yang tersimpan adalah alamat situs tersebut.

4.3.7. Menampilkan *Record*

Pada proses menampilkan *record*, proses akan membaca isi dari basis data, lalu menyimpannya pada sebuah *String*, dan menampilkannya menjadi *TextView*. Khusus data aplikasi, untuk menampilkan durasi pemakaian suatu aplikasi diperlukan sebuah fungsi untuk mengkonversi data dengan satuan detik menjadi satuan waktu dengan format jam, menit, dan detik. Berikut *pseudocode* untuk melakukan konversi durasi.

```

STRING convertDuration(int seconds) {
    String duration
    IF seconds >= 60 THEN
        sec = seconds % 60
        IF seconds >= 3600 THEN
            hour = seconds / 3600;
            min = (seconds - (hour * 3600)) / 60
            duration = hour + " h, " + min + " m, " +
sec + " s"
        ELSE
            min = seconds / 60
            duration = min + " m, " + sec + " s"
        END IF
    ELSE
        duration = seconds + " s"
    END IF
    RETURN duration
}

```

4.3.8. Mengatur Waktu Pengiriman *Record*

Untuk proses pengaturan waktu pengiriman *record*, dilakukan dua tahap. Pertama adalah membaca isi *timepicker* yang ada pada antarmuka ketika tombol ditekan. Berikut *pseudocode* proses tersebut.

```
hour = timePicker.getCurrentHour()
minute = timePicker.getCurrentMinute()
simpan waktu alarm ke basis data
alarm.setAlarm()
```

4.3.9. Mengirim *Record*

Proses pengiriman *record* dilakukan ketika kelas yang mengekstensi *BroadcastReceiver* menerima *broadcast* berupa semacam *alarm*. Proses ini dibagi menjadi beberapa tahap, pertama aplikasi akan memeriksa status ketersediaan koneksi internet. Setelah koneksi internet dipastikan tersedia, proses dilanjutkan dengan penyusunan struktur JSON untuk dikirimkan ke *webservice*. Setelah objek JSON selesai disusun, baru data tersebut dikirim ke *webservice*. Setelah objek dikirim, aplikasi akan menghapus isi *record* di basis data dan mengatur ulang waktu pengiriman *record* menjadi 24 jam berikutnya.

4.3.9.1. Memeriksa Jaringan

Proses pemeriksaan jaringan dilakukan dengan cara pertama memeriksa apakah perangkat bisa terhubung dengan internet. Jika iya, maka proses berikutnya dilakukan. Jika tidak, aplikasi akan memeriksa apakah status perangkat saat itu sedang dalam mode pesawat terbang. Jika iya, maka pengiriman *report* ditunda menjadi 30 menit berikutnya. Jika tidak, maka aplikasi akan menyalakan koneksi internet perangkat lalu mengirimkan *record* untuk kemudian dimatikan lagi setelah beberapa saat. Berikut *pseudocode* proses pemeriksaan jaringan.

```
IF jam sekarang = jam alarm & menit sekarang = menit alarm
THEN
    flagMobileData = 0
    IF globalVariable.getFlagEmailSent() = 0 THEN
```

```

globalVariable.setFlagEmailSent(1)
IF internetAvailable() THEN
    START alarm
ELSE
    IF cek mode airplane = true THEN
        delay alarm 30 menit
    ELSE
        nyalakan mobile data
        SLEEP 10 detik
        IF internetAvailable() THEN
            flagMobileData = 1
            START alarm
        ELSE
            delay alarm 24 jam
        END IF
    END IF
END IF
ELSE
    globalVariable.setFlagEmailSent(0)
END IF

```

4.3.9.2. Membuat Struktur JSON

Setelah dipastikan bahwa *record* dapat dikirim saat itu juga, maka aplikasi akan menyusun objek JSON dari data yang ada di basis data. Berikut *pseudocode* proses pembuatan objek JSON sesuai struktur yang dibutuhkan.

```

PRIVATE STRING createJsonObject (Cursor cursor) {
    inisialisasi jsonArray
    cursor.moveToFirst()
    WHILE posisi cursor < ukuran cursor
        inisialisasi jsonObject
        IF cursor != null THEN
            FOR i=0 TO jumlah kolom cursor
                IF cursor.getColumnname(i) != null &
                cursor.getString(i) != null THEN
                    col = cursor.getColumnname(i)
                    IF col = "count()" THEN
                        col = "total"
                    END IF
                    jsonObject.put(col,
                    cursor.getString(i))
                END IF
            END FOR
        END IF
    END WHILE
}

```

```

        jsonArray.put(jsonObject);
        cursor.moveToNext()
    END WHILE
    CLOSE cursor
    RETURN jsonArray.toString()
}

```

4.3.9.3. Mengirimkan JSON

Proses pengiriman JSON dilakukan dengan sederhana. Setelah semua data *record* yang ingin dikirim ke *webservice* terkumpul menjadi sebuah kumpulan JSON, data yang berupa *String* ini dikirim sebagai *Object* ke sebuah kelas yang mengekstensi *AsyncTask*. Pada kelas ini, diimplementasikan sebuah fungsi *doInBackground()* yang berjalan di samping proses lainnya. Berikut *pseudocode* fungsi *doInBackground()*.

```

PROTECTED OBJECT doInBackground(Object... args) {
    inisialisasi httpClient
    inisialisasi httpPost ke alamat webservice
    hapus brackets dari String email
    POST args.toString()
    RETURN null
}

```

Selanjutnya, untuk menunggu proses pengiriman selesai, aplikasi akan melakukan *sleep* selama 20 detik. Hal ini dilakukan untuk berjaga-jaga jika koneksi internet yang dimiliki perangkat tidak cepat. Kemudian, proses akan memeriksa apakah sebelumnya aplikasi menyalakan koneksi internet dengan sendirinya. Jika iya, maka aplikasi akan mematikan koneksi internet yang ada, lalu mengosongkan isi basis data. Berikut *pseudocode* proses ini secara keseluruhan.

```

GET alamat email dari basis data
GET objek JSON yang akan dikirimkan
POST JSON
SET alarm menjadi 24 jam berikutnya
SLEEP 20 detik
IF flagMobileData = 1 THEN
    matikan koneksi internet
END IF
EMPTY basis data

```

4.3.10. Mengubah Akun

Proses perubahan akun diimplementasikan dengan cara membaca *field editText* pada antarmuka, kemudian dilakukan pengecekan isian data. Terdapat fungsi *updateEmail()* dan *updatePassword()* yang melakukan *query* ke basis data dan akan mengembalikan nilai 1 jika bernilai benar, nilai 0 jika tidak benar. Berikut *pseudocode* untuk menjalankan proses perubahan akun.

```

successUpdateEmail = updateEmail(newEmail)
IF oldPassword = newPassword OR editText is empty THEN
  IF successUpdateEmail = 1 THEN
    TOAST update email sukses
  ELSE
    TOAST update email gagal
  END IF
ELSE
  IF confirmNewPassword != newPassword THEN
    TOAST password tidak sama
  ELSE
    successUpdatePassword = updatePassword(newPassword)
    IF successUpdateEmail = 1 AND successUpdatePassword
= 1 THEN
      TOAST update email & password berhasil
    ELSE
      TOAST update email & password gagal
    END IF
  END IF
END IF

```

4.3.11. Mendaftarkan dan *Login* Akun

Proses untuk masuk ke aplikasi ditangani oleh satu *fragment*. *Fragment* ini pula yang menangani apakah antarmuka yang ditampilkan berupa antarmuka untuk mendaftar atau *login*. Berikut *pseudocode* untuk memeriksa antarmuka mana yang harus ditampilkan.

```

IF ada user di basis data THEN
  isLogin = true
  confirmPassText visibility = GONE
  SET teks pada tombol menjadi "Sign In"
  SET judul menjadi "Sign In"

```

```
ELSE
    tampilkan activity tutorial
END IF
```

Setelah mengetahui antarmuka seperti apa yang akan ditampilkan, jika pengguna menekan tombol yang ada di antarmuka, maka proses berikut yang akan dijalankan.

```
IF akan login THEN
    IF field tidak kosong THEN
        bolehLogin = cek email dan password
        IF bolehLogin = true THEN
            tampilkan halaman overview
        ELSE
            TOAST email atau password salah
        END IF
    ELSE
        TOAST field kosong
    END IF
ELSE
    IF field tidak kosong THEN
        IF password dan confirmPassword sama THEN
            simpan user dan password ke basis data
            tampilkan halaman overview
        ELSE
            TOAST password tidak sama
        END IF
    ELSE
        TOAST field kosong
    END IF
END IF
```

4.3.12. Menghitung Statistik

Proses penghitungan statistik dilakukan melalui eksekusi *query* ke basis data. Berikut implementasi *query* untuk melakukan penghitungan masing-masing statistik.

4.3.12.1. Aplikasi

Berikut proses penghitungan frekuensi aplikasi diurutkan mulai dari yang paling banyak.

```
database.query(TABLE_APPS, new String[]{"package",
    "count()"}, null, null, "package", null, "count() DESC")
```

4.3.12.2. Panggilan Masuk dan Keluar

Berikut proses penghitungan frekuensi nomor yang melakukan panggilan masuk ke perangkat diurutkan mulai dari yang paling banyak.

```
database.query(TABLE_INCOMINGCALL, new String[]{"number",
"count()"}, null, null, "number", null, "count() DESC")
```

Berikut proses penghitungan frekuensi nomor yang paling sering dihubungi oleh perangkat dan diurutkan mulai dari yang paling banyak.

```
database.query(TABLE_OUTGOINGCALL, new String[]{"number",
"count()"}, null, null, "number", null, "count() DESC")
```

4.3.12.3. Pesan Singkat Masuk dan Keluar

Berikut proses penghitungan frekuensi nomor yang paling sering mengirim pesan singkat ke perangkat diurutkan berdasarkan yang paling banyak.

```
database.query(TABLE_INCOMINGSMS, new String[]{"number",
"count()"}, null, null, "number", null, "count() DESC")
```

Berikut proses penghitungan frekuensi nomor yang paling sering menerima pesan singkat dari perangkat diurutkan dari yang paling banyak.

```
database.query(TABLE_OUTGOINGSMS, new String[]{"number",
"count()"}, null, null, "number", null, "count() DESC")
```

4.3.12.4. Durasi

Berikut proses penghitungan durasi pemakaian masing-masing aplikasi diurutkan mulai dari yang terlama.

```
database.rawQuery("SELECT package, sum(duration) as total
FROM " + TABLE_APPS + " GROUP BY package ORDER BY
sum(duration) DESC", null)
```

4.3.13. Menampilkan Statistik

Proses menampilkan statistik diawali dengan eksekusi *query* statistik yang diinginkan. Selanjutnya, hasil eksekusi yang berupa *Cursor* dikirim ke fungsi *drawchart()* beserta deskripsi diagram yang diinginkan. Mula-mula proses akan menghitung isi data pada *cursor* tersebut. Batas jumlah data yang akan ditampilkan pada diagram maksimal 5 data. Selanjutnya, data jumlah frekuensi yang ada pada *Cursor* beserta namanya dimasukkan ke dalam *array* yang nantinya dijadikan sebagai *dataset* untuk dibaca dan ditampilkan pada diagram. Setelah itu, hal yang perlu dilakukan cukup mengatur beberapa atribut tampilan diagram. Berikut *pseudocode* dari fungsi *drawChart()*.

```

VOID drawChart(Cursor cursor, String description) {
    count = cursor.getCount()
    totalCount = 0
    IF count > 5 THEN
        count = 5
    END IF
    cursor.moveToFirst()
    FOR i=0 TO count
        totalCount += cursor.getInt(1)
        cursor.moveToNext()
    END FOR
    inisialisasi array ENTRY yVal
    inisialisasi array STRING xVal
    SET array warna diagram
    cursor.moveToFirst()
    FOR i=0 TO count
        yVal.add(new Entry(cursor.getInt(1), i))
        xVal.add(cursor.getString(0))
        cursor.moveToNext()
    END FOR
    inisialisasi pieDataSet dengan yVal
    SET slice space diagram
    SET warna diagram
    inisialisasi pieData dengan xVal dan pieDataSet
    SET ukuran teks
    SET penulisan format teks
    SET pieData ke pieChart
    SET deskripsi diagram
    SET waktu animasi
    cursor.close()
    SET posisi legend
    INVALIDATE diagram
}

```

```
}

```

4.3.14. Menambahkan Favorit

Proses penambahan aplikasi maupun *domain* ke daftar favorit dilakukan dengan memanfaatkan sebuah *listview* yang menampilkan sepasang *checkbox* dan *textview*. Ketika sebuah *checkbox* dipilih oleh pengguna, maka nilai *textview* dimasukkan ke dalam sebuah *array* sebagai tempat penampung sementara. Ketika pengguna menekan tombol untuk menambahkan data ke daftar favorit, proses berikut ini dilakukan oleh aplikasi.

```
count = checkedItems.size()
FOR i=0 TO count
    favorite.add(checkedItems.get(i))
END FOR
addToDatabase()
tampilkan fragment overview

```

Pada fungsi *addToDatabase()* dilakukan dua proses. Pertama adalah menambahkan daftar aplikasi atau domain ke dalam tabel BANNEDAPPS atau BANNEDLINK. Kedua melakukan proses *update* nilai *flag* pada setiap aplikasi atau URL yang termasuk ke daftar ini menjadi bernilai 1. Berikut *pseudocode* fungsi *addToDatabase()*.

```
IF spinner = application THEN
    FOR i=0 TO favorite.size()
        ADD daftar aplikasi favorit
        UPDATE flag aplikasi
    END FOR
ELSE
    FOR i=0 TO favorite.size()
        ADD daftar domain favorit
        UPDATE flag URL
    END FOR
END IF
CLEAR array favorite

```

Setelah proses di atas dilakukan, maka tampilan akan berpindah ke halaman *overview*. Di halaman ini, semua aplikasi atau situs dengan *domain* yang baru saja ditambahkan akan ditampilkan dengan warna merah.

4.3.15. Menghapus Favorit

Proses penghapusan aplikasi maupun *domain* dari daftar favorit dilakukan dengan memanfaatkan sebuah *listview* yang menampilkan sepasang *checkbox* dan *textview*. Ketika sebuah *checkbox* dipilih oleh pengguna, maka nilai *textview* dimasukkan ke dalam sebuah *array* sebagai tempat penampung sementara. Ketika pengguna menekan tombol untuk menghapus data dari daftar favorit, proses berikut ini dilakukan oleh aplikasi.

```
count = checkedItems.size()
FOR i=0 TO count
    removeFromDatabase(checkedItems.get(i))
END FOR
tampilkan fragment overview
```

Pada fungsi *removeFromDatabase()* dilakukan dua proses. Pertama adalah menghapus daftar aplikasi atau domain dari dalam tabel BANNEDAPPS atau BANNEDLINK. Kedua melakukan proses *update* nilai *flag* pada setiap aplikasi atau URL yang dari *array* ini menjadi bernilai 0. Berikut *pseudocode* fungsi *removeFromDatabase()*.

```
IF spinner = application THEN
    DELETE data dari tabel BANNEDAPPS
    UPDATE flag dari tabel APPS
ELSE
    DELETE data dari tabel BANNEDLINK
    UPDATE flag dari tabel URL
END IF
```

Setelah proses di atas dilakukan, maka tampilan akan berpindah ke halaman *overview*. Di halaman ini, semua aplikasi atau situs dengan *domain* yang baru saja dihapus dari daftar favorit akan ditampilkan dengan warna *default*.

4.4. Implementasi Proses Pada Aplikasi *Webservice*

Pada aplikasi *webservice*, pertama kali yang dilakukan adalah menerima *post* data dari *client* dan mengubahnya menjadi data berbentuk *array*. Oleh karena data yang dikirim berbentuk JSON, maka diperlukan proses *decoding* JSON supaya terbaca isi objeknya. Berikut proses yang dilakukan pada tahap ini.

```

$app = $this->input->post('apps');
$brow = $this->input->post('browser');
$incs = $this->input->post('incsms');
$out = $this->input->post('outsms');
$incc = $this->input->post('inccall');
$outc = $this->input->post('outcall');
$loca = $this->input->post('loc');
$mail = $this->input->post('email');
$countapps = $this->input->post('capps');
$countincc = $this->input->post('cicall');
$countoutc = $this->input->post('cocall');
$countincs = $this->input->post('cisms');
$countouts = $this->input->post('cosms');
$countdura = $this->input->post('cdura');

$array_content["apps"] = json_decode($app);
$array_content["brow"] = json_decode($brow);
$array_content["incs"] = json_decode($incs);
$array_content["outs"] = json_decode($outs);
$array_content["incc"] = json_decode($incc);
$array_content["outc"] = json_decode($outc);
$array_content["loca"] = json_decode($loca);
$array_content["countapps"] = json_decode($countapps);
$array_content["countincc"] = json_decode($countincc);
$array_content["countoutc"] = json_decode($countoutc);
$array_content["countincs"] = json_decode($countincs);
$array_content["countouts"] = json_decode($countouts);
$array_content["countdura"] = json_decode($countdura);

```

Setelah *array_content* terbentuk, proses berikutnya yang dilakukan adalah mengubah isi data pada kolom durasi menjadi berformat jam, menit, dan detik. Hal ini dilakukan karena data yang ada pada saat ini dalam format detik. Berikut *pseudocode* fungsi untuk merubah data durasi.

```

FUNCTION seconds_to_time($seconds)
{
    inisialisasi datetime
    IF seconds > 3600 THEN
        RETURN format jam, menit, detik
    ELSE IF seconds > 60 THEN
        RETURN format menit, detik
    ELSE
        RETURN detik
    END IF
}

```

Selanjutnya, aplikasi melakukan proses *request* alamat dari data koordinat lintang dan bujur yang dimiliki ke server Google. Berikut *pseudocode* proses ini.

```
inisialisasi array addresses
FOR i=0 TO count(array_content["loca"])
    latlng = array_content["loca"][i]->coordinate
    IF (i % 10) = 0 THEN
        SLEEP 1 detik
    END IF
    endpoint = request google API dengan latlng
    address = ambil isi endpoint
    PUSH json_decode(address) ke array addresses
END FOR
array_content["addresses"] = addresses
```

Setelah proses-proses di atas dijalankan, variabel *array_content* beserta alamat *email* dikirim ke halaman *view* lalu halaman tersebut dikirim ke alamat *email* yang ingin dituju.

4.5. Implementasi Tampilan *Report*

Pada subbab ini dijabarkan isi dari tampilan *report* yang dikirimkan ke alamat *email* tujuan. Konsep dari tampilan ini hanya dengan membuat tabel-tabel lalu menampilkan isi *array_content* satu per satu ke tabel masing-masing sesuai jenis datanya. Hal ini dilakukan beberapa kali sampai semua data ditampilkan. Berikut contoh implementasi dari pembuatan satu tabel.

Application Duration



Package Name	Duration
9GAG	2 hours, 7 minutes and 31 seconds
Facebook	44 minutes and 7 seconds
LINE	26 minutes and 50 seconds
Instagram	11 minutes and 5 seconds
detikcom	9 minutes and 27 seconds
Twitter	4 minutes and 14 seconds
Goal.com	2 minutes and 35 seconds
Browser	2 minutes and 5 seconds
Settings	1 minutes and 55 seconds
Gallery	1 minutes and 24 seconds
Contacts and dialer	1 minutes and 12 seconds
Messaging	45 seconds
Clean Master	41 seconds
Phone	34 seconds
Android System	9 seconds

Gambar 4. 16 Implementasi *Report*



BAB V PENGUJIAN DAN EVALUASI

Bab ini membahas tentang pengujian dan evaluasi dari perangkat lunak pada tugas akhir ini. Pengujian dilakukan pada fungsionalitas perangkat lunak. Pembahasan tentang pengujian dan evaluasi disampaikan pada subbab-subbab berikut.

5.1. Lingkungan Pelaksanaan Pengujian

Lingkungan pengujian menjelaskan tentang perangkat yang dilibatkan dalam proses pengujian. Lingkungan pengujian ini menggunakan perangkat keras berupa perangkat bergerak berbasis Android yang dapat terhubung dengan. Spesifikasi lingkungan pengujian dijelaskan pada tabel berikut.

Tabel 5. 1 Spesifikasi Perangkat Bergerak

Perangkat Bergerak Android	Processor	Qualcomm 2.5 GHz Quad-core
	RAM	3 GB
	Sistem Operasi	Android versi 4.4.2
	ROM	MIUI
	Status	Tidak <i>root</i>

Sementara spesifikasi server menggunakan *web hosting* milik kuncihost dengan kapasitas *disk* sebesar 500 MB dan *bandwidth* 70 GB.

5.2. Pengujian Fungsionalitas

Subbab ini menjelaskan tentang skenario pengujian fungsionalitas perangkat lunak pada tugas akhir ini. Pengujian didokumentasikan secara sistematis sebagai tolok ukur keberhasilan sistem. Uji coba fungsionalitas terbagi menjadi beberapa skenario berikut.

- Uji coba melakukan pendaftaran akun
- Uji coba melakukan *login* ke aplikasi
- Uji coba pencatatan aplikasi yang dibuka
- Uji coba pencatatan pesan singkat masuk
- Uji coba pencatatan pesan singkat keluar
- Uji coba pencatatan panggilan telepon masuk
- Uji coba pencatatan panggilan telepon keluar
- Uji coba pencatatan alamat situs
- Uji coba pencatatan lokasi
- Uji coba menampilkan statistik
- Uji coba ubah akun
- Uji coba atur jadwal pengiriman *record*
- Uji coba pengiriman *record*
- Uji coba penambahan aplikasi favorit
- Uji coba penambahan *domain* favorit
- Uji coba penghapusan aplikasi favorit
- Uji coba penghapusan *domain* favorit

Khusus pada bagian uji coba pengiriman *record*, hal ini nantinya sekaligus membuktikan apakah *webservice* mampu menangani permintaan pengiriman *record* dengan baik.

5.2.1. Pengujian Melakukan Pendaftaran dan *Login* Akun

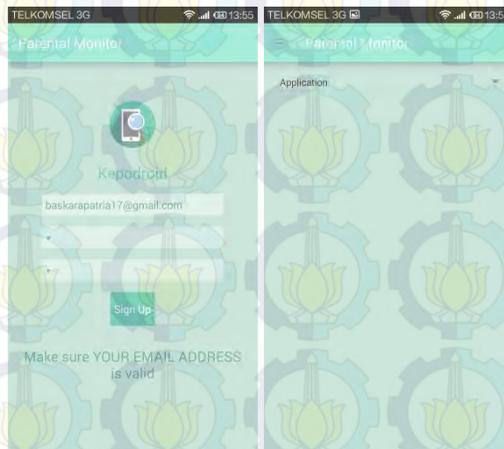
Pengujian melakukan pendaftaran dan *login* akun dilakukan dengan dua tahap. Skenario pertama dijalankan saat pengguna baru saja melakukan instalasi pada perangkat bergerak, sehingga belum ada data pengguna pada basis data. Skenario pengujian pertama dijelaskan pada tabel berikut.

Tabel 5. 2 Rincian Pengujian Melakukan Pendaftaran

Nomor	PA-01
Nama	Melakukan pendaftaran akun
Tujuan	Memeriksa fungsi untuk menangani akun

	baru apakah berfungsi atau tidak
Kondisi awal	Data akun belum ada
Skenario	<ol style="list-style-type: none"> 1. Pengguna melakukan instalasi aplikasi ke perangkat 2. Pengguna memasukkan alamat <i>email</i> dan <i>password</i> beserta konfirmasi <i>password</i> 3. Pengguna menekan tombol <i>sign up</i>
Masukan	Alamat <i>email</i> dan <i>password</i> beserta konfirmasi <i>password</i>
Keluaran yang diharapkan	Halaman <i>overview</i> ditampilkan
Hasil pengujian	Berhasil

Berikut tampilan atarmuka saat akan mendaftar dan halaman *overview* ketika pengguna telah menekan tombol *sign up*.



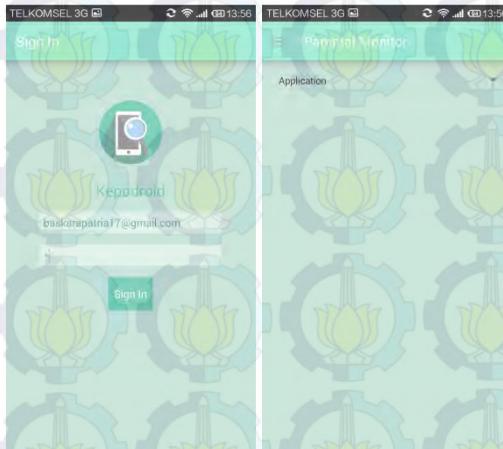
Gambar 5. 1 Hasil Pengujian PA-01

Berikutnya, dengan data pengguna yang didaftarkan pada skenario pertama, dilakukan pengujian *login* akun dengan skenario kedua. Berikut penjelasan skenario pengujian kedua.

Tabel 5. 3 Rincian Pengujian Melakukan Login

Nomor	PA-02
Nama	Melakukan <i>login</i> akun
Tujuan	Memeriksa fungsi untukm menangani <i>login</i> apakah berfungsi atau tidak
Kondisi awal	Data akun sudah ada
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi 2. Pengguna memasukkan alamat <i>email</i> dan <i>password</i> yang sudah didaftarkan 3. Pengguna menekan tombol <i>sign in</i>
Masukan	Alamat <i>email</i> dan <i>password</i>
Keluaran yang diharapkan	Halaman <i>overview</i> ditampilkan
Hasil pengujian	Berhasil

Berikut tampilan layar saat akan melakukan *login* beserta tampilan layar saat pengguna telah melakukan proses *login* dengan berhasil.

**Gambar 5. 2 Hasil Pengujian PA-02**

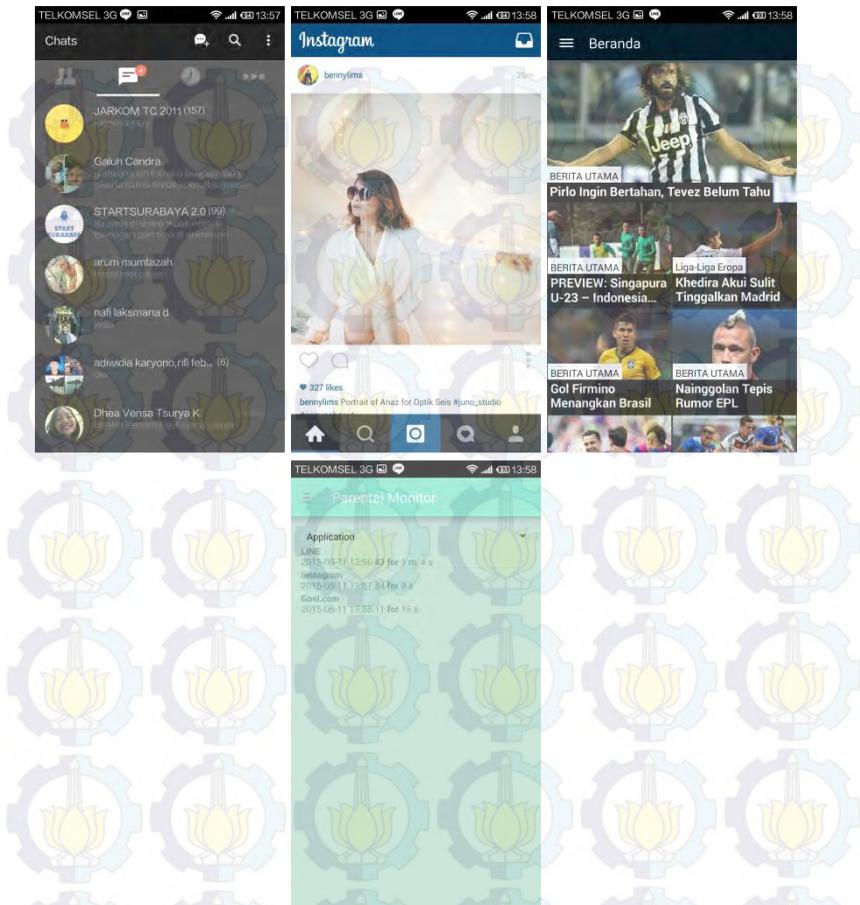
5.2.2. Pengujian Pencatatan Aplikasi

Pengujian ini dilakukan untuk memeriksa apakah fungsi pencatat aktivitas perangkat bergerak yang berjalan pada servis utama bekerja dengan baik. Skenario pengujian dilakukan dengan cara membuka aplikasi apa saja selain aplikasi tugas akhir ini kemudian menutup aplikasi tersebut setelah beberapa saat. Berikut penjelasan skenario pengujian tersebut.

Tabel 5. 4 Rincian Pengujian Melakukan Pencatatan Aplikasi

Nomor	PA-03
Nama	Melakukan pencatatan aplikasi
Tujuan	Memeriksa fungsi untuk mencatat aplikasi yang dibuka oleh perangkat yang berjalan pada servis utama berfungsi atau tidak
Kondisi awal	Servis utama aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi selain aplikasi pemantau 2. Pengguna menutup aplikasi yang baru saja dibuka 3. Pengguna membuka aplikasi pemantau 4. Pengguna melakukan proses <i>login</i>
Masukan	Alamat <i>email</i> dan <i>password</i> untuk <i>login</i>
Keluaran yang diharapkan	Halaman <i>overview</i> ditampilkan, terdapat data aplikasi yang baru saja dibuka sebelumnya
Hasil pengujian	Berhasil

Pada pengujian kali ini, percobaan dilakukan dengan membuka aplikasi LINE, lalu instagram, dan GOAL(dot)com secara berurutan. Berikut tampilan layar saat pengguna membuka beberapa aplikasi lalu tampilan aplikasi pemantau saat menampilkan *record* aplikasi yang baru dibuka.



Gambar 5. 3 Hasil Pengujian PA-03

5.2.3. Pengujian Pencatatan Pesan Singkat Masuk

Pengujian ini dilakukan untuk memeriksa apakah penerima *broadcast* pesan singkat masuk yang ada pada aplikasi ini berjalan. Skenario dilakukan dengan cara mengirimkan pesan singkat dari perangkat lain ke perangkat yang memiliki aplikasi

pemantau pemakaian perangkat bergerak. Berikut penjelasan mengenai skenario ini.

Tabel 5. 5 Rincian Pengujian Melakukan Pencatatan SMS Masuk

Nomor	PA-04
Nama	Melakukan pencatatan pesan singkat masuk
Tujuan	Memeriksa fungsi untuk mencatat data pesan singkat yang masuk ke perangkat apakah berfungsi atau tidak
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Pengguna menerima pesan singkat yang dikirim oleh perangkat lain 2. Pengguna membuka aplikasi pemantau 3. Pengguna melakukan proses <i>login</i> 4. Pengguna memilih <i>incoming sms</i> pada <i>spinner</i> di halaman <i>overview</i>
Masukan	Alamat <i>email</i> dan <i>password</i> untuk <i>login</i>
Keluaran yang diharapkan	Halaman <i>overview</i> ditampilkan, terdapat data pesan singkat yang baru saja masuk
Hasil pengujian	Berhasil

Berikut tampilan hasil pengujian pencatatan pesan singkat yang masuk ke perangkat.



Gambar 5. 4 Hasil Pengujian PA-04

5.2.4. Pengujian Pencatatan Pesan Singkat Keluar

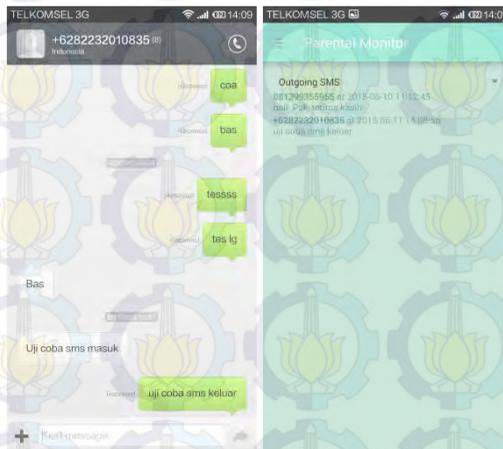
Pengujian ini dilakukan untuk memeriksa apakah servis yang membaca setiap pesan singkat yang dikirimkan oleh perangkat bergerak dapat berfungsi dengan baik. Skenario dilakukan dengan cara mengirim pesan singkat dari perangkat ini ke sebuah nomor telepon. Saat melakukan pengiriman pesan singkat, servis yang menangani proses ini harus dipastikan berjalan terlebih dahulu. Berikut penjelasan mengenai skenario pengujian ini.

Tabel 5. 6 Rincian Pengujian Melakukan Pencatatan SMS Keluar

Nomor	PA-05
Nama	Melakukan pencatatan pesan singkat keluar
Tujuan	Memeriksa fungsi untuk mencatat data pesan singkat yang dikirim oleh perangkat apakah berfungsi atau tidak
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	1. Pengguna mengirim pesan singkat ke suatu nomor telepon

	<ol style="list-style-type: none"> 2. Pengguna memastikan bahwa pesan singkat telah terkirim ke nomor telepon tujuan 3. Pengguna membuka aplikasi pemantau 4. Pengguna melakukan proses <i>login</i> 5. Pengguna memilih <i>outgoing sms</i> pada <i>spinner</i> di halaman <i>overview</i>
Masukan	Alamat <i>email</i> dan <i>password</i> untuk <i>login</i>
Keluaran yang diharapkan	Halaman <i>overview</i> ditampilkan, terdapat data pesan singkat yang baru saja terkirim
Hasil pengujian	Berhasil

Berikut tampilan antarmuka saat proses pengujian pencatatan pesan singkat yang dikirimkan perangkat tengah berlangsung



Gambar 5.5 Hasil Pengujian PA-05

5.2.5. Pengujian Pencatatan Panggilan Telepon Masuk

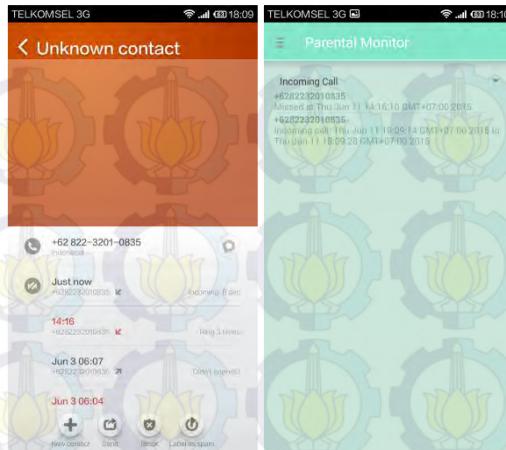
Pengujian ini dilakukan untuk memeriksa apakah aplikasi dapat mencatat data panggilan telepon yang masuk ke perangkat. Selain itu, aplikasi juga harus mampu membedakan jenis panggilan yang masuk, apakah diangkat oleh pemilik perangkat

atau *missed call*. Oleh karena itu, terdapat dua skenario untuk pengujian kali ini. Berikut penjelasan skenario pertama.

Tabel 5. 7 Rincian Pengujian Pencatatan Panggilan Masuk dan Diangkat

Nomor	PA-06
Nama	Melakukan pencatatan panggilan telepon masuk dan diangkat
Tujuan	Memeriksa fungsi untuk mencatat panggilan telepon yang masuk ke perangkat dan diangkat oleh perangkat
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Perangkat yang memiliki aplikasi pemantau dihubungi melalui perangkat lain 2. Pengguna mengangkat telepon yang masuk lalu menutup telepon setelah beberapa saat 3. Pengguna membuka aplikasi pemantau 4. Pengguna melakukan proses <i>login</i> 5. Pengguna memilih <i>incoming call</i> pada <i>spinner</i> di halaman <i>overview</i>
Masukan	Alamat <i>email</i> dan <i>password</i> untuk <i>login</i>
Keluaran yang diharapkan	Halaman <i>overview</i> ditampilkan, terdapat data telepon yang masuk diangkat oleh pengguna
Hasil pengujian	Berhasil

Berikut adalah tampilan antarmuka ketika proses pengujian pencatatan panggilan telepon yang masuk ke perangkat tengah berlangsung.



Gambar 5. 6 Hasil Pengujian PA-06

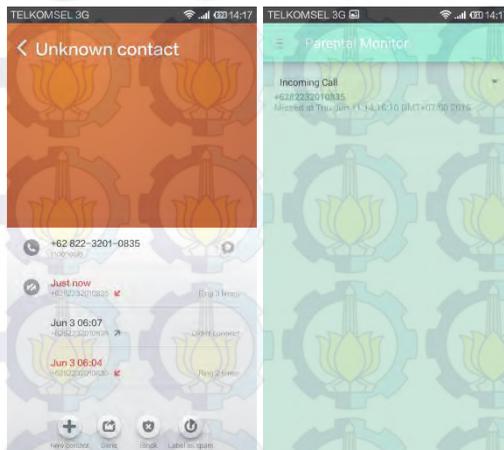
Skenario berikutnya adalah menguji apakah aplikasi pemantau mampu mengetahui bahwa panggilan telepon yang masuk ke perangkat berupa *missed call*. Berikut penjelasan skenario kedua.

Tabel 5. 8 Rincian Pengujian Pencatatan Missed Call

Nomor	PA-07
Nama	Melakukan pencatatan <i>missed call</i>
Tujuan	Memeriksa fungsi untuk mencatat panggilan telepon yang masuk ke perangkat tetapi tidak diangkat
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Perangkat yang memiliki aplikasi pemantau dihubungi melalui perangkat lain 2. Pengguna tidak mengangkat telepon yang masuk 3. Pengguna membuka aplikasi pemantau

	4. Pengguna melakukan proses <i>login</i> 5. Pengguna memilih <i>incoming call</i> pada <i>spinner</i> di halaman <i>overview</i>
Masukan	Alamat <i>email</i> dan <i>password</i> untuk <i>login</i>
Keluaran yang diharapkan	Halaman <i>overview</i> ditampilkan, terdapat data <i>missed call</i> yang baru saja terjadi
Hasil pengujian	Berhasil

Berikut tampilan antarmuka saat proses pengujian pencatatan *missed call* yang masuk ke perangkat sedang berlangsung



Gambar 5. 7 Hasil Pengujian PA-07

5.2.6. Pengujian Pencatatan Panggilan Telepon Keluar

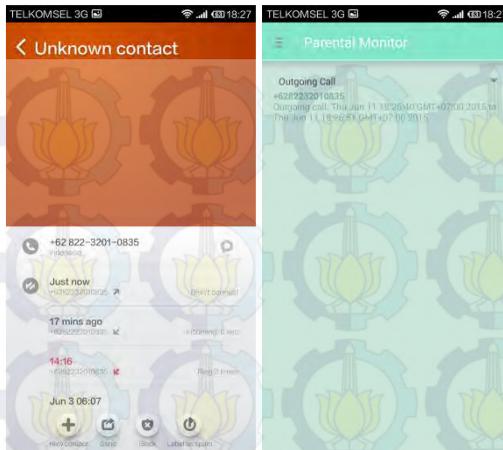
Pengujian pencatatan panggilan telepon keluar dilakukan untuk memeriksa apakah aplikasi dapat menangkap *broadcast* terjadinya panggilan telepon keluar dengan baik. Skenario yang dilakukan untuk pengujian kali ini dijelaskan sebagai berikut.

Tabel 5. 9 Rincian Pengujian Pencatatan Panggilan Keluar

Nomor	PA-08
Nama	Melakukan pencatatan panggilan keluar

Tujuan	Memeriksa fungsi untuk mencatat panggilan telepon yang dilakukan oleh perangkat
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Pengguna menghubungi salah satu nomor telepon 2. Pengguna menutup telepon 3. Pengguna membuka aplikasi pemantau 4. Pengguna melakukan proses <i>login</i> 5. Pengguna memilih <i>outgoing call</i> pada <i>spinner</i> di halaman <i>overview</i>
Masukan	Alamat <i>email</i> dan <i>password</i> untuk <i>login</i>
Keluaran yang diharapkan	Halaman <i>overview</i> ditampilkan, terdapat data panggilan telepon yang baru saja dilakukan
Hasil pengujian	Berhasil

Berikut tampilan antarmuka saat proses pengujian pencatatan panggilan telepon keluar.



Gambar 5. 8 Hasil Pengujian PA-08

5.2.7. Pengujian Pencatatan Alamat Situs

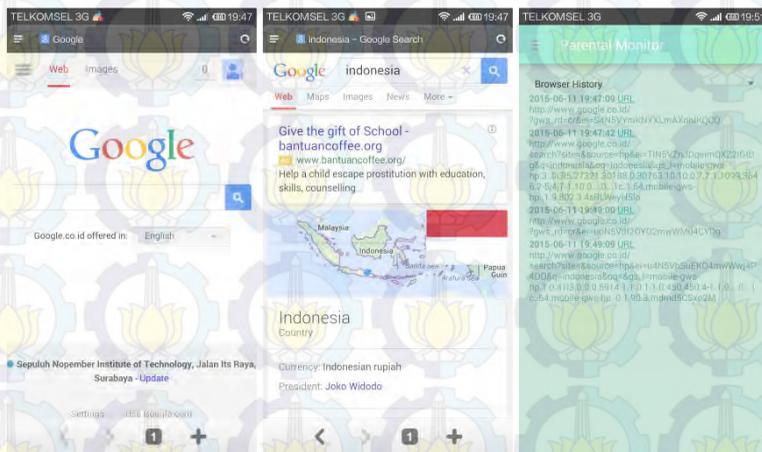
Pengujian ini dilakukan untuk memeriksa apakah servis yang dinyalakan setiap pengguna membuka aplikasi *browser native* perangkat dapat mencatat alamat situs dengan benar. Skenario pengujian dijelaskan pada tabel berikut ini.

Tabel 5. 10 Rincian Pengujian Pencatatan Alamat Situs

Nomor	PA-09
Nama	Melakukan pencatatan alamat situs yang dikunjungi
Tujuan	Memeriksa fungsi untuk mencatat alamat situs yang dikunjungi oleh perangkat
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi <i>browser native</i> pada perangkat 2. Pengguna mengunjungi beberapa situs di internet 3. Pengguna menutup aplikasi <i>browser</i>

	<ol style="list-style-type: none"> 4. Pengguna membuka aplikasi pemantau 5. Pengguna melakukan proses <i>login</i> 6. Pengguna memilih <i>browser history</i> pada <i>spinner</i> di halaman <i>overview</i>
Masukan	Alamat <i>email</i> dan <i>password</i> untuk <i>login</i>
Keluaran yang diharapkan	Halaman <i>overview</i> ditampilkan, terdapat data alamat situs beserta judul situs yang dikunjungi perangkat
Hasil pengujian	Berhasil

Pada pengujian ini, halaman situs masih dalam tahap *loading* setelah satu detik, sehingga judul situs yang tersimpan adalah alamat situsnya. Namun, alamat situs yang tercatat tetap sesuai dengan harapan. Berikut tampilan antarmuka proses pengujian ini.



Gambar 5. 9 Hasil Pengujian PA-09

5.2.8. Pengujian Pencatatan Lokasi

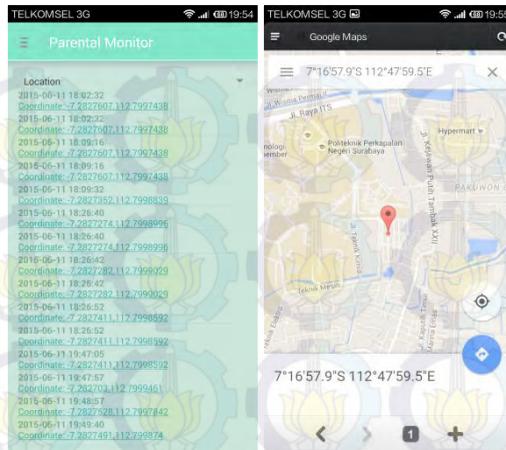
Pengujian pencatatan lokasi dilakukan untuk memeriksa apakah servis yang digunakan untuk mencari posisi koordinat perangkat dapat bekerja dengan baik. Oleh karena proses pencatatan lokasi dilakukan setiap pengguna membuka aplikasi

pada perangkat, skenario pengujian pencatatan lokasi hampir sama dengan skenario pencatatan data aplikasi yang dibuka oleh perangkat. Berikut penjelasan skenario pengujian ini.

Tabel 5. 11 Rincian Pengujian Melakukan Pencatatan Lokasi

Nomor	PA-10
Nama	Melakukan pencatatan lokasi
Tujuan	Memeriksa fungsi untuk mencatat koordinat lokasi perangkat bekerja dengan baik
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi apa saja selain aplikasi pemantau 2. Pengguna menutup aplikasi yang baru saja dibuka 3. Pengguna membuka aplikasi pemantau 4. Pengguna melakukan proses <i>login</i> 5. Pengguna memilih <i>location</i> pada <i>spinner</i> di halaman <i>overview</i> 6. Pengguna menekan koordinat lokasi
Masukan	Alamat <i>email</i> dan <i>password</i> untuk <i>login</i>
Keluaran yang diharapkan	Halaman peta google ditampilkan dengan <i>pinpoint</i> posisi koordinat
Hasil pengujian	Berhasil

Pada pengujian kali ini dilakukan di Jalan Teknik Komputer U-42, Perumahan Dosen ITS Surabaya. Hasil pengujian menunjukkan bahwa koordinat yang dicatat oleh aplikasi pemantau merepresentasikan posisi perangkat dengan cukup baik. Berikut tampilan antarmuka hasil pengujian ini.



Gambar 5. 10 Hasil Pengujian PA-10

5.2.9. Pengujian Menampilkan Statistik

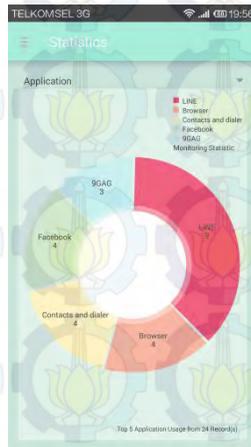
Pengujian ini dilakukan untuk mengetahui apakah *library* MPAndroidChart telah berhasil menampilkan statistik pemakaian perangkat bergerak dengan benar. Pada pengujian kali ini, terdapat lima skenario pengujian. Skenario pertama adalah pengujian untuk menampilkan statistik lima aplikasi yang paling sering dibuka oleh pengguna. Berikut penjelasan dari skenario ini.

Tabel 5. 12 Rincian Pengujian Menampilkan Statistik Aplikasi

Nomor	PA-11
Nama	Menampilkan statistik aplikasi
Tujuan	Memeriksa fungsi untuk menampilkan statistik aplikasi pada perangkat berjalan dengan baik
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna memilih <i>statistics</i> pada

	navigasi aplikasi
Masukan	Alamat <i>email</i> dan <i>password</i> untuk <i>login</i>
Keluaran yang diharapkan	Halaman <i>statistics</i> dengan diagram pai yang menampilkan data lima aplikasi yang paling sering dibuka oleh perangkat
Hasil pengujian	Berhasil

Berikut tampilan antarmuka hasil pengujian statistik aplikasi pada aplikasi pemantau.



Gambar 5. 11 Hasil Pengujian PA-11

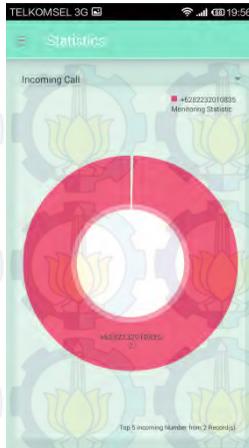
Skenario kedua adalah pengujian untuk menampilkan statistik lima nomor telepon yang paling sering menghubungi perangkat. Berikut penjelasan dari skenario ini.

Tabel 5. 13 Rincian Pengujian Menampilkan Statistik Panggilan Masuk

Nomor	PA-12
Nama	Menampilkan statistik panggilan telepon masuk
Tujuan	Memeriksa fungsi untuk menampilkan

	statistik telepon yang masuk ke perangkat berjalan dengan baik
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna memilih <i>statistics</i> pada navigasi aplikasi 4. Pengguna memilih <i>incoming call</i> pada <i>spinner</i> di halaman <i>statistics</i>
Masukan	Alamat <i>email</i> dan <i>password</i> untuk <i>login</i>
Keluaran yang diharapkan	Halaman <i>statistics</i> dengan diagram pai yang menampilkan data lima nomor telepon yang paling sering menghubungi perangkat
Hasil pengujian	Berhasil

Berikut *screenshot* antarmuka halaman *statistics* yang menampilkan data nomor telepon yang paling sering menghubungi perangkat.



Gambar 5. 12 Hasil Pengujian PA-12

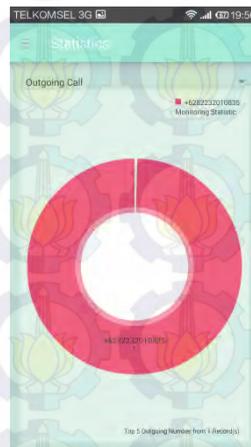
Skenario ketiga adalah pengujian untuk menampilkan statistik lima nomor telepon yang paling sering dihubungi oleh perangkat. Berikut penjelasan dari skenario ini.

Tabel 5. 14 Rincian Pengujian Menampilkan Statistik Panggilan Keluar

Nomor	PA-13
Nama	Menampilkan statistik panggilan telepon keluar
Tujuan	Memeriksa fungsi untuk menampilkan statistik panggilan telepon yang dilakukan oleh perangkat dapat berjalan dengan baik
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna memilih <i>statistics</i> pada navigasi aplikasi 4. Pengguna memilih <i>outgoing call</i> pada

	<i>spinner</i> di halaman <i>statistics</i>
Masukan	Alamat <i>email</i> dan <i>password</i> untuk <i>login</i>
Keluaran yang diharapkan	Halaman <i>statistics</i> dengan diagram pai yang menampilkan data lima nomor telepon yang paling sering dihubungi oleh perangkat
Hasil pengujian	Berhasil

Berikut *screenshot* antarmuka halaman *statistics* yang menampilkan data nomor telepon yang paling sering dihubungi oleh perangkat.



Gambar 5. 13 Hasil Pengujian PA-13

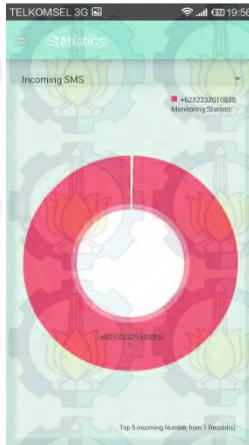
Skenario keempat adalah pengujian untuk menampilkan statistik lima nomor telepon yang paling sering mengirim pesan singkat ke perangkat. Berikut penjelasan dari skenario ini.

Tabel 5. 15 Rincian Pengujian Menampilkan Statistik SMS Masuk

Nomor	PA-14
Nama	Menampilkan statistik pesan singkat masuk
Tujuan	Memeriksa fungsi untuk menampilkan statistik pesan singkat yang masuk ke

	perangkat berjalan dengan baik
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna memilih <i>statistics</i> pada navigasi aplikasi 4. Pengguna memilih <i>incoming sms</i> pada <i>spinner</i> di halaman <i>statistics</i>
Masukan	Alamat <i>email</i> dan <i>password</i> untuk <i>login</i>
Keluaran yang diharapkan	Halaman <i>statistics</i> dengan diagram pai yang menampilkan data lima nomor telepon yang paling sering mengirim pesan singkat ke perangkat
Hasil pengujian	Berhasil

Berikut *screenshot* antarmuka halaman *statistics* yang menampilkan hasil pengujian dengan skenario di atas.



Gambar 5.14 Hasil Pengujian PA-14

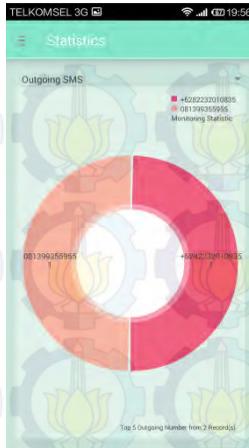
Skenario terakhir adalah pengujian untuk menampilkan statistik lima nomor telepon yang paling sering menjadi tujuan

pengiriman pesan singkat oleh perangkat. Berikut penjelasan dari skenario ini.

Tabel 5. 16 Rincian Pengujian Menampilkan Statistik SMS Keluar

Nomor	PA-15
Nama	Menampilkan statistik pesan singkat keluar
Tujuan	Memeriksa fungsi untuk menampilkan statistik pesan singkat yang dikirim oleh perangkat dapat berjalan dengan baik
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna memilih <i>statistics</i> pada navigasi aplikasi 4. Pengguna memilih <i>outgoing sms</i> pada <i>spinner</i> di halaman <i>statistics</i>
Masukan	Alamat <i>email</i> dan <i>password</i> untuk <i>login</i>
Keluaran yang diharapkan	Halaman <i>statistics</i> dengan diagram pai yang menampilkan data lima nomor telepon yang paling sering menerima pesan singkat dari perangkat
Hasil pengujian	Berhasil

Berikut tampilan antarmuka hasil pengujian dengan skenario di atas.



Gambar 5. 15 Hasil Pengujian PA-15

5.2.10. Pengujian Ubah Akun

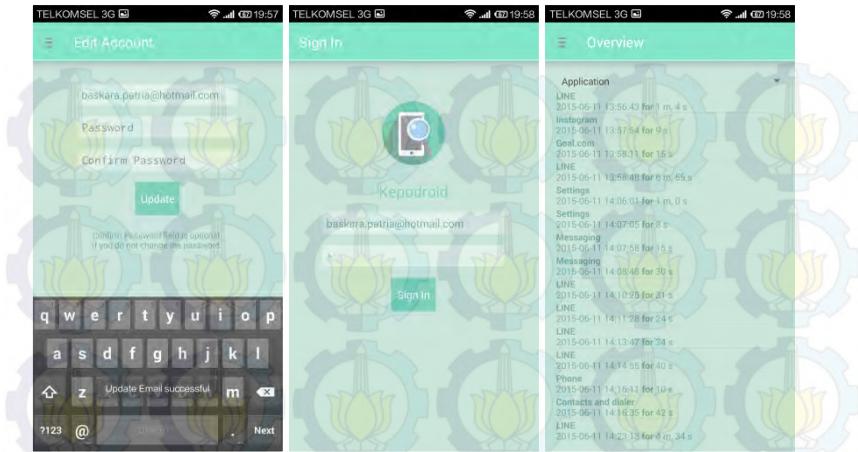
Pengujian ini dilakukan untuk memeriksa apakah fungsi yang menangani proses ubah akun dapat berjalan dengan baik atau tidak. Ada dua skenario pengujian yang dilakukan. Pertama, pengguna hanya mengubah alamat *email* saja tanpa mengubah *password*. Kedua, pengguna mengubah *password* saja tanpa mengubah alamat *email*. Ketiga, pengguna mengubah alamat *email* beserta *password*. Untuk memeriksa apakah aplikasi sudah menangani proses ini dengan baik, pengguna akan menggunakan alamat *email* atau *password* barunya untuk melakukan proses *login* ulang. Berikut penjelasan dari skenario pertama.

Tabel 5. 17 Rincian Pengujian Mengubah Alamat Email

Nomor	PA-16
Nama	Mengubah alamat <i>email</i>
Tujuan	Memeriksa fungsi untuk mengubah alamat <i>email</i> dapat berjalan dengan baik
Kondisi awal	Aplikasi pemantau berjalan di latar belakang

Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna memilih <i>edit account</i> pada navigasi aplikasi 4. Pengguna memasukkan alamat <i>email</i> baru 5. Pengguna menekan tombol <i>update</i> 6. Pengguna menutup aplikasi pemantau 7. Pengguna membuka kembali aplikasi pemantau 8. Pengguna melakukan proses <i>login</i> dengan alamat <i>email</i> yang baru
Masukan	<ol style="list-style-type: none"> 1. Alamat <i>email</i> dan <i>password</i> untuk <i>login</i> pertama 2. Alamat <i>email</i> pengganti 3. Alamat <i>email</i> dan <i>password</i> untuk <i>login</i> kedua
Keluaran yang diharapkan	Halaman <i>overview</i>
Hasil pengujian	Berhasil

Berikut *screenshot* pengujian dengan skenario di atas.



Gambar 5. 16 Hasil Pengujian PA-16

Berikut penjelasan skenario kedua dimana pengujian dilakukan dengan mengubah *password* akun.

Tabel 5. 18 Rincian Pengujian Mengubah Password

Nomor	PA-17
Nama	Mengubah <i>password</i>
Tujuan	Memeriksa fungsi untuk mengubah <i>password</i> dapat berjalan dengan baik
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna memilih <i>edit account</i> pada navigasi aplikasi 4. Pengguna memasukkan <i>password</i> baru 5. Pengguna menekan tombol <i>update</i> 6. Pengguna menutup aplikasi pemantau 7. Pengguna membuka kembali aplikasi pemantau

	8. Pengguna melakukan proses <i>login</i> dengan alamat <i>email</i> yang baru
Masukan	<ol style="list-style-type: none"> 1. Alamat <i>email</i> dan <i>password</i> untuk <i>login</i> pertama 2. <i>Password</i> pengganti 3. Alamat <i>email</i> dan <i>password</i> untuk <i>login</i> kedua
Keluaran yang diharapkan	Halaman <i>overview</i>
Hasil pengujian	Berhasil

Berikut *screenshot* hasil pengujian di atas.



Gambar 5. 17 Hasil Pengujian PA-17

Berikut penjelasan skenario terakhir dimana pengujian dilakukan dengan mengubah alamat *email* beserta *password* akun.

Tabel 5. 19 Rincian Pengujian Mengubah Alamat Email dan Password

Nomor	PA-18
Nama	Mengubah alamat <i>email</i> dan <i>password</i>

Tujuan	Memeriksa fungsi untuk mengubah alamat <i>email</i> beserta <i>password</i> dapat berjalan dengan baik
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna memilih <i>edit account</i> pada navigasi aplikasi 4. Pengguna memasukkan alamat <i>email</i> dan <i>password</i> baru 5. Pengguna menekan tombol <i>update</i> 6. Pengguna menutup aplikasi pemantau 7. Pengguna membuka kembali aplikasi pemantau 8. Pengguna melakukan proses <i>login</i> dengan alamat <i>email</i> yang baru
Masukan	<ol style="list-style-type: none"> 1. Alamat <i>email</i> dan <i>password</i> untuk <i>login</i> pertama 2. Alamat <i>email</i> dan <i>password</i> pengganti 3. Alamat <i>email</i> dan <i>password</i> untuk <i>login</i> kedua
Keluaran yang diharapkan	Halaman <i>overview</i>
Hasil pengujian	Berhasil

Berikut *screenshot* hasil pengujian di atas.



Gambar 5. 18 Hasil Pengujian PA-18

5.2.11. Pengujian Atur Jadwal dan Pengiriman *Record*

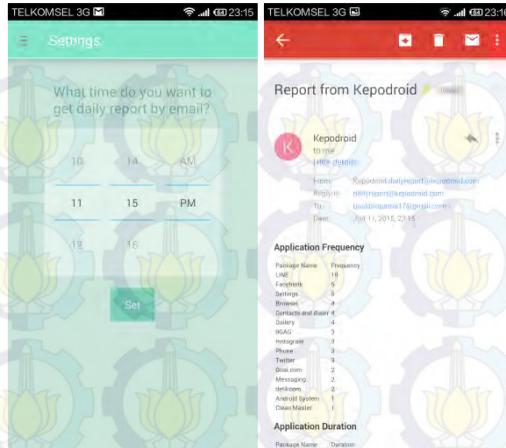
Pengujian ini dilakukan untuk memeriksa apakah fungsi pengaturan *alarm* pada aplikasi dapat berjalan dengan baik. Hal ini ditandai dengan terkirimnya *record* ke *webservice*. Oleh karena *webservice* bersifat hanya meneruskan data yang dikirim ke dirinya, maka untuk memeriksa apakah pengujian kali ini berhasil maka indikatornya adalah *record* terkirim ke alamat *email* tujuan. Skenario pengujian pengiriman *record* kali ini dibagi menjadi tiga jenis. Skenario pertama adalah pengujian pengaturan jadwal pengiriman *record* ketika pada jadwal yang ditentukan, perangkat sedang memiliki koneksi ke internet. Berikut penjelasan dari skenario pertama.

Tabel 5. 20 Rincian Pengujian Pengiriman Record Saat Terhubung Internet

Nomor	PA-19
Nama	Mengirimkan <i>record</i> ketika perangkat terhubung dengan internet
Tujuan	Memeriksa fungsi untuk mengatur jadwal

	pengiriman <i>record</i> dan fungsi untuk mengirim <i>record</i>
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna memilih <i>setting</i> pada navigasi aplikasi 4. Pengguna memasukkan waktu yang ditentukan 5. Pengguna menekan tombol <i>set</i>
Masukan	<ol style="list-style-type: none"> 1. Alamat <i>email</i> dan <i>password</i> untuk <i>login</i> 2. Jam dan menit pengiriman <i>record</i>
Keluaran yang diharapkan	<i>Email</i> terkirim ke akun <i>email</i> tujuan sesuai jadwal
Hasil pengujian	Berhasil

Pada skenario di atas, *record* terkirim ke alamat *email* tujuan dengan benar. Berikut *screenshot* saat pengaturan waktu pengiriman dan saat *email* masuk.



Gambar 5. 19 Hasil Pengujian PA-19

Skenario kedua adalah pengujian pengaturan jadwal pengiriman *record* ketika pada saat jadwal yang ditentukan, koneksi perangkat ke internet sedang dimatikan. Hal ini dilakukan untuk memeriksa apakah fungsi mengaktifkan dan menonaktifkan koneksi internet menggunakan paket *mobile* data bekerja dengan baik. Berikut skenario pengujian kali ini.

Tabel 5. 21 Rincian Pengujian Pengiriman Record Saat Tidak Terhubung Internet

Nomor	PA-20
Nama	Mengirimkan <i>record</i> ketika perangkat tidak terhubung dengan internet
Tujuan	Memeriksa fungsi untuk mengatur jadwal pengiriman <i>record</i> dan fungsi untuk mengirim <i>record</i> ketika perangkat sedang tidak terhubung dengan internet
Kondisi awal	Aplikasi pemantau berjalan di latar belakang dan perangkat tidak sedang terhubung dengan internet

Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna memilih <i>setting</i> pada navigasi aplikasi 4. Pengguna memasukkan waktu yang ditentukan 5. Pengguna menekan tombol <i>set</i>
Masukan	<ol style="list-style-type: none"> 1. Alamat <i>email</i> dan <i>password</i> untuk <i>login</i> 2. Jam dan menit pengiriman <i>record</i>
Keluaran yang diharapkan	<i>Email</i> terkirim ke akun <i>email</i> tujuan sesuai jadwal, koneksi internet terputus
Hasil pengujian	Berhasil

Skenario terakhir adalah pengujian pengaturan jadwal pengiriman *report* ketika pada saat jadwal yang ditentukan, perangkat sedang dalam mode pesawat terbang. Hasil yang diharapkan adalah aplikasi tidak mengirimkan *report* dan menundanya. Berikut penjelasan skenario pengujian ini.

Tabel 5. 22 Rincian Pengujian Pengiriman Record Saat Mode Pesawat Terbang

Nomor	PA-21
Nama	Mengirimkan <i>record</i> ketika perangkat sedang dalam mode pesawat terbang
Tujuan	Memeriksa fungsi untuk mengatur jadwal pengiriman <i>record</i> dan fungsi untuk mengirim <i>record</i> ketika perangkat sedang dalam mode pesawat terbang
Kondisi awal	Aplikasi pemantau berjalan di latar belakang dan perangkat sedang dalam mode pesawat terbang
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna memilih <i>setting</i> pada navigasi aplikasi 4. Pengguna memasukkan waktu yang

	ditentukan 5. Pengguna menekan tombol <i>set</i>
Masukan	1. Alamat <i>email</i> dan <i>password</i> untuk <i>login</i> 2. Jam dan menit pengiriman <i>record</i>
Keluaran yang diharapkan	Perangkat tetap dalam mode pesawat terbang, <i>report</i> tidak dikirimkan
Hasil pengujian	Berhasil

Pada skenario terakhir ini, *report* tidak dikirimkan oleh aplikasi pemantau dan posisi perangkat tetap pada mode pesawat terbang. Selain itu, aplikasi pemantau juga tidak mengaktifkan koneksi internet pada perangkat. Namun setelah 30 menit dan perangkat telah terhubung dengan internet, *report* terkirim dengan normal.

5.2.12. Pengujian Penambahan Aplikasi Favorit

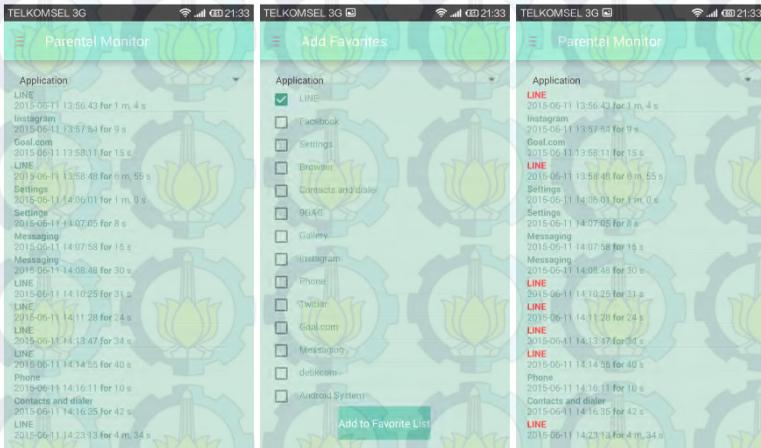
Pengujian ini dilakukan untuk memeriksa apakah fungsi penambahan aplikasi favorit bekerja dan aplikasi dapat membedakan aplikasi favorit dengan aplikasi lainnya. Hal ini ditandai ketika aplikasi yang masuk ke daftar favorit akan ditampilkan dengan warna merah pada halaman *overview*. Berikut penjelasan skenario pengujian kali ini.

Tabel 5. 23 Rincian Pengujian Menambahkan Aplikasi Favorit

Nomor	PA-22
Nama	Menambahkan aplikasi favorit
Tujuan	Memeriksa fungsi untuk menambahkan dan menampilkan aplikasi favorit
Kondisi awal	Aplikasi pemantau berjalan di latar belakang dan telah mencatat beberapa pemakaian aplikasi lain
Skenario	1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna memilih <i>add favorites</i> pada navigasi aplikasi 4. Pengguna memilih aplikasi yang

	<p>ditentukan</p> <p>5. Pengguna menekan tombol <i>add to favorites</i></p>
Masukan	<p>1. Alamat <i>email</i> dan <i>password</i> untuk <i>login</i></p> <p>2. Aplikasi yang ingin dimasukkan ke daftar favorit</p>
Keluaran yang diharapkan	Halaman <i>overview</i> menampilkan aplikasi favorit ditulis dengan warna merah
Hasil pengujian	Berhasil

Berikut *screenshot* proses pengujian skenario di atas.



Gambar 5. 20 Hasil Pengujian PA-22

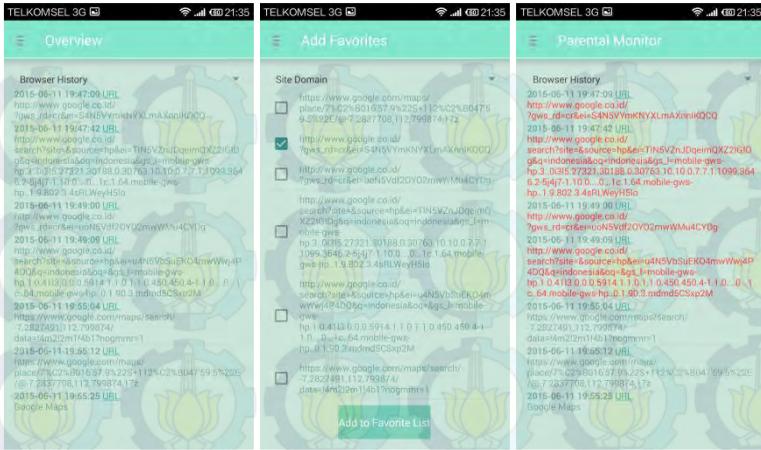
5.2.13. Pengujian Penambahan *Domain* Favorit

Pengujian ini dilakukan untuk memeriksa apakah fungsi penambahan *domain* favorit bekerja dan aplikasi dapat membedakan situs dengan *domain* favorit dan situs dengan *domain* lainnya. Hal ini ditandai ketika situs yang memiliki *domain* yang masuk ke daftar favorit akan ditampilkan dengan warna merah pada halaman *overview*. Berikut penjelasan skenario pengujian kali ini.

Tabel 5. 24 Rincian Pengujian Menambahkan Domain Favorit

Nomor	PA-23
Nama	Menambahkan <i>domain</i> favorit
Tujuan	Memeriksa fungsi untuk menambahkan dan menampilkan situs dengan <i>domain</i> favorit
Kondisi awal	Aplikasi pemantau berjalan di latar belakang dan telah mencatat beberapa kunjungan ke beberapa situs dengan <i>domain</i> berbeda-beda
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna memilih <i>add favorites</i> pada navigasi aplikasi 4. Pengguna memilih <i>site domain</i> pada <i>spinner</i> 5. Pengguna memilih situs yang ditentukan 6. Pengguna menekan tombol <i>add to favorites</i> 7. Pengguna memilih <i>browser history</i> pada <i>spinner</i> di halaman <i>overview</i>
Masukan	<ol style="list-style-type: none"> 1. Alamat <i>email</i> dan <i>password</i> untuk <i>login</i> 2. Situs dengan <i>domain</i> yang ingin dimasukkan ke daftar favorit
Keluaran yang diharapkan	Halaman <i>overview</i> menampilkan situs dengan <i>domain</i> favorit ditulis dengan warna merah
Hasil pengujian	Berhasil

Berikut *screenshot* proses pengujian dengan skenario di atas.



Gambar 5. 21 Hasil Pengujian PA-23

5.2.14. Pengujian Penghapusan Aplikasi Favorit

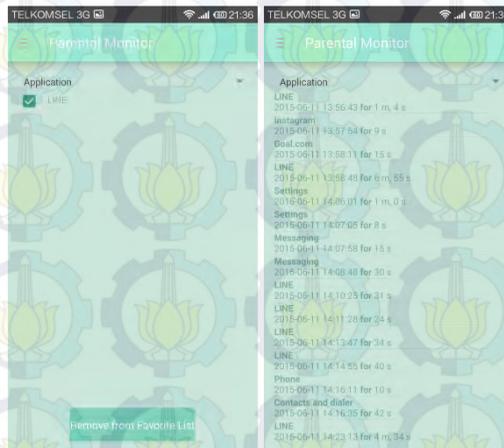
Pengujian ini dilakukan untuk memeriksa apakah fungsi penghapusan aplikasi dari daftar favorit bekerja dengan baik. Hal ini ditandai ketika aplikasi yang sebelumnya termasuk ke dalam daftar aplikasi favorit dan ditampilkan dengan warna merah pada halaman *overview*, kembali ditampilkan dengan warna *default*. Berikut penjelasan dari skenario pengujian kali ini.

Tabel 5. 25 Rincian Pengujian Menghapus Aplikasi Favorit

Nomor	PA-24
Nama	Menghapus aplikasi favorit
Tujuan	Memeriksa fungsi untuk menghapus aplikasi favorit
Kondisi awal	Aplikasi pemantau berjalan di latar belakang dan telah memiliki beberapa aplikasi yang termasuk ke dalam daftar favorit
Skenario	1. pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i>

	<ol style="list-style-type: none"> 3. Pengguna memilih <i>remove favorites</i> pada navigasi aplikasi 4. Pengguna memilih aplikasi yang ditentukan 5. Pengguna menekan tombol <i>remove from favorites</i>
Masukan	<ol style="list-style-type: none"> 1. Alamat <i>email</i> dan <i>password</i> untuk <i>login</i> 2. Aplikasi yang ingin dihilangkan dari daftar favorit
Keluaran yang diharapkan	Halaman <i>overview</i> menampilkan aplikasi yang baru saja dihilangkan dari daftar favorit ditulis dengan warna <i>default</i>
Hasil pengujian	Berhasil

Berikut *screenshot* proses pengujian dengan skenario di atas.



Gambar 5. 22 Hasil Pengujian PA-24

5.2.15. Pengujian Penghapusan *Domain* Favorit

Pengujian ini dilakukan untuk memeriksa apakah fungsi penghapusan *domain* dari daftar favorit bekerja dengan baik. Hal ini ditandai ketika situs-situs dengan *domain* yang sebelumnya termasuk ke dalam daftar *domain* favorit dan ditampilkan dengan

warna merah pada halaman *overview*, kembali ditampilkan dengan warna *default*. Berikut penjelasan dari skenario pengujian kali ini.

Tabel 5. 26 Rincian Pengujian Menghapus Domain Favorit

Nomor	PA-25
Nama	Menghapus <i>domain</i> favorit
Tujuan	Memeriksa fungsi untuk menghapus <i>domain</i> favorit
Kondisi awal	Aplikasi pemantau berjalan di latar belakang dan telah memiliki beberapa <i>domain</i> yang termasuk ke dalam daftar favorit
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna memilih <i>remove favorites</i> pada navigasi aplikasi 4. Pengguna memilih <i>site domain</i> pada <i>spinner</i> 5. Pengguna memilih <i>domain</i> yang akan dihilangkan 6. Pengguna menekan tombol <i>remove from favorites</i> 7. Pengguna memilih <i>browser history</i> pada <i>spinner</i> di halaman <i>overview</i>
Masukan	<ol style="list-style-type: none"> 1. Alamat <i>email</i> dan <i>password</i> untuk <i>login</i> 2. <i>Domain</i> yang ingin dihilangkan dari daftar favorit
Keluaran yang diharapkan	Halaman <i>overview</i> menampilkan situs dengan <i>domain</i> yang baru saja dihilangkan dari daftar favorit ditulis dengan warna <i>default</i>
Hasil pengujian	Berhasil

Berikut *screenshot* proses pengujian dengan skenario di atas.



Gambar 5. 23 Hasil Pengujian PA-25

5.3. Pengujian Non Fungsionalitas

Pada subbab ini dibahas pengujian non fungsionalitas yang dilakukan pada aplikasi pemantau pada perangkat bergerak. Pada tahap ini aplikasi akan dilihat perilakunya ketika pengguna melakukan pembersihan memori melalui *task manager*, pembersihan memori melalui aplikasi *third party*, menghentikan servis melalui *application manager*, dan memantau konsumsi memori oleh aplikasi.

5.3.1. Pembersihan Memori Melalui *Task Manager*

Pengujian kali ini dilakukan untuk mengetahui perilaku aplikasi ketika pengguna memutuskan untuk melakukan pembersihan memori melalui *task manager*. Normalnya, aplikasi akan tertutup dan servis-servis yang berjalan pada pada latar belakang perangkat akan berhenti berjalan. Android sebenarnya memberikan fasilitas *START_STICKY*¹¹ yang dapat ditambahkan pada kode sumber servis yang dibuat. Menurut dokumentasi dari situs *developer.android.com* fitur ini mengakibatkan sistem

¹¹ http://developer.android.com/reference/android/app/Service.html#START_STICKY

menyalakan kembali servis yang telah dimatikan oleh sistem ketika selesai melakukan pembersihan memori. Namun ketika dicoba untuk diimplementasikan, hasilnya negatif. Oleh sebab itu dilakukan pengujian terhadap metode *broadcast* yang diimplementasikan pada aplikasi pemantau kali ini.

Skenario yang dilakukan dimulai dengan menyalakan servis aplikasi pemantau supaya berjalan pada latar belakang. Setelah itu, pengguna melakukan pembersihan memori melalui *task manager*. Hasil yang diharapkan adalah servis pada aplikasi berjalan kembali. Berikut penjelasan skenario pengujian ini.

Tabel 5. 27 Rincian Pengujian Pembersihan Memori Melalui Task Manager

Nomor	PB-1
Nama	Membersihkan memori melalui <i>task manager</i>
Tujuan	Memeriksa fungsi untuk menyalakan servis kembali setelah dihentikan oleh sistem melalui <i>task manager</i>
Kondisi awal	Aplikasi pemantau berjalan di latar belakang dan <i>permission</i> untuk <i>auto start</i> telah diaktifkan
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna menutup aplikasi pemantau 4. Pengguna memeriksa adanya aplikasi pemantau pada daftar <i>task manager</i> 5. Pengguna menekan <i>clear memory</i> 6. Pengguna membuka menu <i>application manager</i> pada menu <i>setting</i> di perangkat 7. Pengguna memilih aplikasi pemantau pada menu <i>running application</i>
Masukan	1. Alamat <i>email</i> dan <i>password</i> untuk <i>login</i>
Keluaran yang	Servis yang terdapat pada aplikasi pemantau

diharapkan	berjalan
Hasil pengujian	Berhasil

Berikut tampilan layar ketika pengguna akan melakukan proses *clear memory* dari *task manager*.



Gambar 5. 24 Proses Pembersihan Memori Dari Task Manager

Setelah pengguna menekan tombol bergambar kuas, maka muncul notifikasi pada layar yang menunjukkan ada *foreground service* yang baru saja berjalan.



Gambar 5. 25 Notifikasi Foreground Service

Ketika notifikasi dibuka, muncul logo aplikasi pemantau yang menandakan bahwa servis milik aplikasi ini tengah berjalan di latar belakang.



Gambar 5. 26 Tampilan Notifikasi Foreground Service Berjalan

Meskipun pada kenyataannya servis milik aplikasi pemantau dinyalakan ulang, ternyata pada menu *application manager*, terlihat durasi servis milik aplikasi pemantau yang tidak di-reset.



Gambar 5. 27 Tampilan Menu Application Manager

5.3.2. Pembersihan Memori Melalui Aplikasi Luar

Pengujian ini dilakukan untuk memeriksa perilaku aplikasi pemantau ketika pengguna melakukan pembersihan memori menggunakan aplikasi luar. Pada kasus ini, dilakukan pengujian menggunakan aplikasi *Clean Master*. Berikut skenario pengujian kali ini.

Tabel 5. 28 Rincian Pengujian Pembersihan Memori Melalui Aplikasi Luar

Nomor	PB-2
Nama	Membersihkan memori melalui aplikasi luar
Tujuan	Memeriksa fungsi untuk menyalakan servis kembali setelah dihentikan oleh sistem melalui aplikasi luar
Kondisi awal	Aplikasi pemantau berjalan di latar belakang dan <i>permission</i> untuk <i>auto start</i> telah diaktifkan
Skenario	1. Pengguna membuka aplikasi pembersih memori

	<ol style="list-style-type: none"> 2. Pengguna memeriksa adanya aplikasi pemantau pada daftar yang akan dibersihkan 3. Pengguna menekan <i>boost</i> 4. Pengguna menutup aplikasi pembersih memori 5. Pengguna membuka menu <i>application manager</i> pada menu <i>setting</i> di perangkat 6. Pengguna memilih aplikasi pemantau pada menu <i>running application</i>
Masukan	-
Keluaran yang diharapkan	Servis yang terdapat pada aplikasi pemantau berjalan
Hasil pengujian	Berhasil

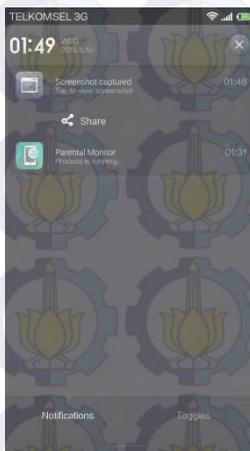
Berikut *screenshot* proses pengujian pembersihan memori melalui aplikasi luar.



Gambar 5. 28 Tampilan Clear Master

Pada tampilan di atas menunjukkan bahwa terdapat memori sebesar 25 MB yang bisa dibersihkan yang berasal dari pemakaian aplikasi pemantau. Berikut tampilan notifikasi

foreground service aplikasi pemantau setelah proses pembersihan memori yang dilakukan oleh *clean master* selesai.



Gambar 5. 29 Notifikasi *Foreground Service*

Berikut tampilan daftar servis yang dijalankan oleh aplikasi pemantau melalui menu *application manager*.



Gambar 5. 30 Tampilan Application Manager

Pada pengujian kali ini membuktikan bahwa servis ternyata tidak dihentikan oleh aplikasi *clear master* meskipun pada keterangannya terdapat sekitar 25 MB memori yang dibersihkan berasal dari pemakaian aplikasi pemantau.

5.3.3. Penghentian Servis Melalui *Application Manager*

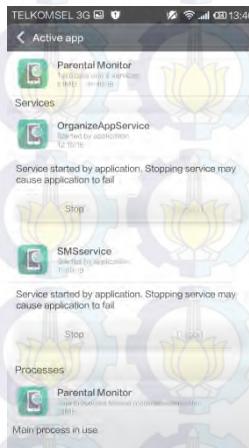
Pengujian ini dilakukan untuk memeriksa perilaku aplikasi pemantau ketika pengguna menghentikan servis yang berjalan pada aplikasi pemantau. Pada kasus ini, dilakukan penghentian melalui menu *application manager* yang ada pada menu *settings* milik perangkat. Berikut skenario pengujian kali ini.

Tabel 5. 29 Rincian Pengujian Penghentian Servis

Nomor	PB-3
Nama	Menghentikan servis melalui <i>application manager</i>
Tujuan	Memeriksa fungsi untuk menyalakan servis kembali setelah dihentikan oleh pengguna
Kondisi awal	Aplikasi pemantau berjalan di latar belakang dan <i>permission</i> untuk <i>auto start</i>

	telah diaktifkan
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna menutup aplikasi pemantau 4. Pengguna membuka menu <i>application manager</i> pada menu <i>setting</i> di perangkat 5. Pengguna memilih aplikasi pemantau pada menu <i>running application</i> 6. Pengguna menekan tombol <i>stop</i> pada servis milik aplikasi pemantau
Masukan	-
Keluaran yang diharapkan	Servis yang terdapat pada aplikasi pemantau berjalan kembali
Hasil pengujian	Berhasil

Berikut tampilan menu *application manager* ketika servis milik aplikasi pemantau belum dihentikan. Di situ terlihat bahwa durasi servis ini telah berjalan selama beberapa jam.



Gambar 5. 31 Tampilan Saat Akan Dihentikan

Berikut tampilan menu *application manager* ketika pengguna baru saja menghentikan kedua servis milik aplikasi

pemantau. Terlihat bahwa durasi servis berjalan menandakan bahwa servis ini baru saja di-*restart* ulang.



Gambar 5. 32 Tampilan Setelah Dihentikan

5.3.4. Konsumsi Memori

Pada subbab ini ditampilkan *screenshot* konsumsi memori aplikasi pemantau dari waktu ke waktu. Berikut *screenshot* konsumsi memori aplikasi pemantau pada saat aplikasi di latar belakang berjalan kurang dari 1 jam.



Gambar 5. 33 Konsumsi Memori < 1 Jam

Pada gambar di atas, angka menunjukkan bahwa aplikasi memakan kapasitas memori sebesar 37 MB. Selanjutnya, konsumsi memori oleh aplikasi terpantau sedikit menurun. Berikut *screenshot* yang menunjukkan angka konsumsi memori oleh aplikasi setelah berjalan selama sekitar 2 jam.



Gambar 5. 34 Konsumsi Memori 2 Jam

Pada gambar di atas, angka konsumsi memori menunjukkan adanya penurunan konsumsi memori menjadi 34 MB. Berikutnya, dilakukan pemeriksaan konsumsi memori setelah aplikasi pemantau berjalan sekitar 7 jam. Berikut *screenshot* hasil pemeriksaan tersebut.



Gambar 5. 35 Konsumsi Memori 7 Jam

Gambar di atas menunjukkan bahwa konsumsi memori oleh aplikasi pemantau sebesar 58 MB. Pada proses pemeriksaan berikutnya, aplikasi terlihat telah berjalan selama hampir 10 jam. Pada waktu ini, konsumsi memori yang dilakukan oleh aplikasi pemantau tercatat sebesar 26 MB. Berikut *screenshot* hasil pemantauan konsumsi memori aplikasi pemantau.



Gambar 5. 36 Konsumsi Memori 9 Jam

5.4. Evaluasi Pengujian

Berdasarkan hasil pengujian fungsionalitas dan pengujian non fungsionalitas, maka dapat dilakukan evaluasi sebagai berikut.

5.4.1. Evaluasi Pengujian Fungsionalitas

Pengujian fungsionalitas yang telah dilakukan memberikan hasil yang sesuai dengan skenario yang telah direncanakan. Evaluasi dari pengujian pada masing-masing fungsionalitas dijelaskan dengan sebagai berikut.

1. Pengujian PA-01 mengeluarkan hasil sesuai harapan dimana proses pendaftaran berjalan dengan sempurna.
2. Pengujian PA-02 mengeluarkan hasil sesuai harapan dimana proses *login* berjalan dengan sempurna.
3. Pengujian PA-03 mengeluarkan hasil sesuai harapan dimana proses pencatatan pemakaian aplikasi ditampilkan dengan benar.

4. Pengujian PA-04 mengeluarkan hasil sesuai harapan dimana pencatatan data pesan singkat yang masuk ditampilkan dengan benar.
5. Pengujian PA-05 mengeluarkan hasil sesuai harapan dimana pencatatan data pesan singkat yang dikirim oleh perangkat ditampilkan dengan benar.
6. Pengujian PA-06 mengeluarkan hasil sesuai harapan dimana pencatatan data panggilan telepon masuk yang diangkat oleh perangkat ditampilkan dengan benar.
7. Pengujian PA-07 mengeluarkan hasil sesuai harapan dimana pencatatan data *missed call* yang masuk ke perangkat ditampilkan dengan benar.
8. Pengujian PA-08 mengeluarkan hasil sesuai harapan dimana proses pencatatan data panggilan telepon yang dilakukan oleh perangkat ditampilkan dengan benar.
9. Pengujian PA-09 mengeluarkan hasil sesuai harapan, namun ada sedikit catatan. Proses pencatatan alamat situs dapat dilakukan dengan benar. Ketika situs dapat *ter-load* dengan sempurna dalam waktu singkat, maka judul situs yang tengah dibuka juga akan ditampilkan dengan tepat. Namun, ketika situs tidak dapat *ter-load* dalam waktu singkat, judul yang ditampilkan adalah alamat situs.
10. Pengujian PA-10 mengeluarkan hasil sesuai harapan namun ada sedikit catatan. Proses pencatatan koordinat lokasi berjalan dengan benar. Ketika perangkat tidak mampu menemukan lokasinya saat itu, maka yang disimpan adalah koordinat lokasi yang tercatat saat proses pencarian lokasi terakhir.
11. Pengujian PA-11 mengeluarkan hasil sesuai harapan dimana statistik aplikasi yang paling sering digunakan ditampilkan dengan benar.
12. Pengujian PA-12 mengeluarkan hasil sesuai harapan dimana statistik nomor telepon yang paling sering menghubungi perangkat ditampilkan dengan benar.

13. Pengujian PA-13 mengeluarkan hasil sesuai harapan dimana statistik nomor telepon yang paling sering dihubungi oleh perangkat ditampilkan dengan benar.
14. Pengujian PA-14 mengeluarkan hasil sesuai harapan dimana nomor telepon yang paling sering mengirimkan pesan singkat ke perangkat ditampilkan dengan benar.
15. Pengujian PA-15 mengeluarkan hasil sesuai harapan dimana nomor telepon yang paling sering menerima pesan singkat dari perangkat ditampilkan dengan benar.
16. Pengujian PA-16 mengeluarkan hasil sesuai harapan dimana alamat *email* berubah tetapi *password* akun tidak berubah.
17. Pengujian PA-17 mengeluarkan hasil sesuai harapan dimana alamat *email* tidak berubah tetapi *password* berubah.
18. Pengujian PA-18 mengeluarkan hasil sesuai harapan dimana alamat *email* dan *password* berubah.
19. Pengujian PA-19 mengeluarkan hasil sesuai harapan dimana *record* terkirim ke alamat *email* tujuan serta data pemakaian perangkat bergerak pada basis data dihapus oleh aplikasi.
20. Pengujian PA-20 mengeluarkan hasil sesuai harapan dimana *record* terkirim ke alamat tujuan dan koneksi internet perangkat dimatikan kembali secara otomatis oleh aplikasi serta data pemakaian perangkat bergerak pada basis data dihapus oleh aplikasi.
21. Pengujian PA-21 mengeluarkan hasil sesuai harapan dimana *record* tidak dikirimkan oleh aplikasi dan data pemakaian perangkat bergerak tidak dihapus oleh aplikasi. Proses pengiriman dilakukan 30 menit setelah jadwal yang seharusnya.
22. Pengujian PA-22 mengeluarkan hasil sesuai harapan dimana aplikasi yang baru saja ditambahkan ke daftar favorit ditampilkand dengan warna merah.

23. Pengujian PA-23 mengeluarkan hasil sesuai harapan dimana situs dengan *domain* yang baru saja dimasukkan ke daftar favorit ditampilkan dengan warna merah.
24. Pengujian PA-24 mengeluarkan hasil sesuai harapan dimana aplikasi yang baru saja dihilangkan dari daftar favorit ditampilkan dengan warna *default*.
25. Pengujian PA-25 mengeluarkan hasil sesuai harapan dimana situs dengan *domain* yang baru saja dihilangkan dari daftar favorit ditampilkan dengan warna *default*.

5.4.2. Evaluasi Pengujian Non Fungsionalitas

Pengujian non fungsionalitas telah dilakukan dengan beberapa skenario yang ada dan memberikan hasil yang sesuai harapan. Evaluasi dari proses-proses pengujian dijelaskan sebagai berikut.

1. Pengujian PB-1 mengeluarkan hasil sesuai harapan dimana cara pengaktifan kembali servis menggunakan *broadcast* mampu menyasati terhentinya proses pemantauan ketika pengguna melakukan pembersihan memori melalui *task manager*.
2. Pengujian PB-2 mengeluarkan hasil sesuai harapan dimana hasil menunjukkan bahwa ternyata aplikasi luar tidak mengganggu kinerja aplikasi pemantau saat melakukan proses pembersihan memori.
3. Pengujian PB-3 mengeluarkan hasil sesuai harapan dimana cara pengaktifan kembali servis menggunakan *broadcast* mampu menangani masalah berhentinya servis saat pengguna menghentikan servis secara paksa melalui menu *application manager*.
4. Pengujian PB-4 mengeluarkan hasil yang relatif sesuai harapan. Jumlah konsumsi memori yang fluktuatif menunjukkan bahwa aplikasi pemantau dapat bertahan dengan baik ketika servis-servis miliknya berjalan dalam waktu yang lama. Hal ini mengindikasikan bahwa proses-proses utama milik aplikasi pemantau tetap bisa

bekerja dengan baik ketika sistem operasi melakukan pembersihan memori secara berkala. Konsumsi maksimal memori yang dilakukan oleh aplikasi pemantau relatif sekitar 60 MB.

Hasil pengujian non fungsionalitas pada aplikasi pemantau sebenarnya bergantung pada spesifikasi dan kondisi ROM dan *firmware* perangkat. Khusus pada perangkat yang digunakan pada pengujian kali ini, pengguna harus mengaktifkan opsi *auto-start* pada bagian *permission list* yang ada pada menu *setting* di perangkat. Jika tidak, secara *default* ROM perangkat-perangkat yang tidak dalam status *root* tidak akan mengizinkan servis-servis pada suatu aplikasi untuk dapat menyala secara otomatis.

DAFTAR PUSTAKA

Anna, L. K. (2011, Oktober 26). *Bahaya Balita Gemar Memainkan iPad*. Dipetik Mei 13, 2015, dari Kompas Health:

<http://health.kompas.com/read/2011/10/26/08073739/Bahaya.Balita.Gemar.Memainkan.iPad>

Hall, B. S. (2015, April 17). *Best Parental-Control Apps 2015*. Dipetik Mei 13, 2015, dari Tom's Guide:

<http://www.tomsguide.com/us/best-parental-control-apps,review-2258.html>

Singhal, M., & Shukla, A. (2012, Mei 20). Implementation of Location based Services in Android using GPS and Web Services. *IJCSI International Journal of Computer Science Issues*, 237-242.

Stenneth, L., Wolfson, O., Yu, P., & Xu, B. (2011). Transportation mode detection using mobile phones and GIS information. *19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 54-63.

BAB VI PENUTUP

Pada bab ini dijelaskan kesimpulan yang dapat diambil dalam pengerjaan tugas akhir dan saran untuk pengembangan lebih lanjut dari sistem pemantau pemakaian perangkat bergerak anak ini.

5.1. Kesimpulan

Dalam proses pengerjaan tugas akhir dari tahap pendahuluan, kajian pustaka, analisis, perancangan, implementasi dan pengujian perangkat diperoleh kesimpulan sebagai berikut.

1. Cara kerja *service* adalah dengan berjalan di belakang layar dan menjalankan proses yang diinginkan sampai ada perintah untuk berhenti. Sedangkan cara kerja *asynchronous task* adalah dengan berjalan di belakang layar dan menjalankan proses lalu berhenti dengan sendirinya ketika proses tersebut selesai. Sementara cara kerja *broadcast* adalah dengan mengirimkan sebuah pesan atau perintah yang dibaca oleh kelas penerimanya untuk melakukan *trigger* tertentu.
2. *Service* pada aplikasi pemantau dapat dibuat menyala kembali dengan seketika saat baru saja dimatikan dengan cara mengirimkan sebuah *broadcast* kepada suatu kelas yang di dalamnya terdapat fungsi untuk menyalakan kembali *service* ketika *service* tersebut dimatikan.
3. Aplikasi pemantau pada perangkat yang dibangun berbasis android dapat memantau aktivitas dan lokasi perangkat bergerak.
4. Aplikasi pemantau pada perangkat yang dibangun berbasis android dapat memproduksi statistik pemakaian perangkat bergerak.

5. Sistem pemantau pemakaian perangkat bergerak mampu mengirimkan data beserta statistik yang diproduksi oleh aplikasi ke alamat *email* tujuan secara terjadwal.

5.2. Saran

Berikut ini merupakan beberapa saran mengenai pengembangan lebih lanjut sistem pemantau pemakaian perangkat bergerak anak berdasarkan hasil rancangan, implementasi dan uji coba yang telah dilakukan.

1. Perbaiki tampilan antarmuka pada aplikasi perangkat bergerak. Tampilan yang elegan dan mengikuti panduan *material design* yang terbaru akan menarik masyarakat untuk menggunakan aplikasi ini.
2. Perbaiki tampilan *report* yang dikirimkan oleh sistem ke alamat *email* tujuan. Tampilan statistik dalam bentuk grafik yang lebih informatif akan memudahkan pengguna membaca informasi di dalamnya.
3. Menambahkan fitur pencatat *traffic* media sosial maupun aplikasi *chatting*. Hal ini untuk menyikapi semakin seringnya penggunaan layanan seperti ini untuk sarana komunikasi antar individu.
4. Aplikasi ini masih memiliki kelemahan pada pencatatan judul situs. Pada saat halaman situs belum terbuka pada kurun waktu tertentu karena koneksi jaringan yang lambat, maka yang dicatat sebagai judul adalah alamat situs tersebut. Untuk itu perlu adanya perbaikan pada pengembangan selanjutnya.
5. Ikon aplikasi ini masih terlihat ketika terpasang pada perangkat. Hal ini memungkinkan anak menghapus aplikasi dari perangkatnya. Diperlukan suatu metode implementasi yang berbeda agar ikon aplikasi tersembunyi sehingga anak tidak tahu bahwa dia sedang dipantau oleh orang tua.



FINAL PROJECT - KI091391

KIDS MOBILE DEVICE USAGE MONITORING SYSTEM

BASKARA NUR PATRIA
NRP 5111 100 097

Advisor
Dr.Tech. Ir. R.V. Hari Ginardi, M.Sc.
Ir. F.X. Arunanto, M.Sc.

INFORMATICS DEPARTMENT
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2015

LEMBAR PENGESAHAN

Sistem Pemantau Pemakaian Perangkat Bergerak Anak.

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Algoritma Pemrograman
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

BASKARA NUR PATRIA

NRP : 5111 100 097

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Dr.Tech. Ir. R.V. Hari Ginardi, M.Sc.

NIP: 196505181992031009

(pembimbing 1)

Ir. F.X. Arunanto, M.Sc.

NIP: 195701011983031004

(pembimbing 2)

SURABAYA

JUNI 2015

Sistem Pemantau Pemakaian Perangkat Bergerak Anak.

Nama Mahasiswa : Baskara Nur Patria
NRP : 5111 100 097
Jurusan : Teknik Informatika FTIf-ITS
Dosen Pembimbing I : Dr.Tech. Ir. R.V. Hari Ginardi, M.Sc.
Dosen Pembimbing II : Ir. F.X. Arunanto, M.Sc.

ABSTRAK

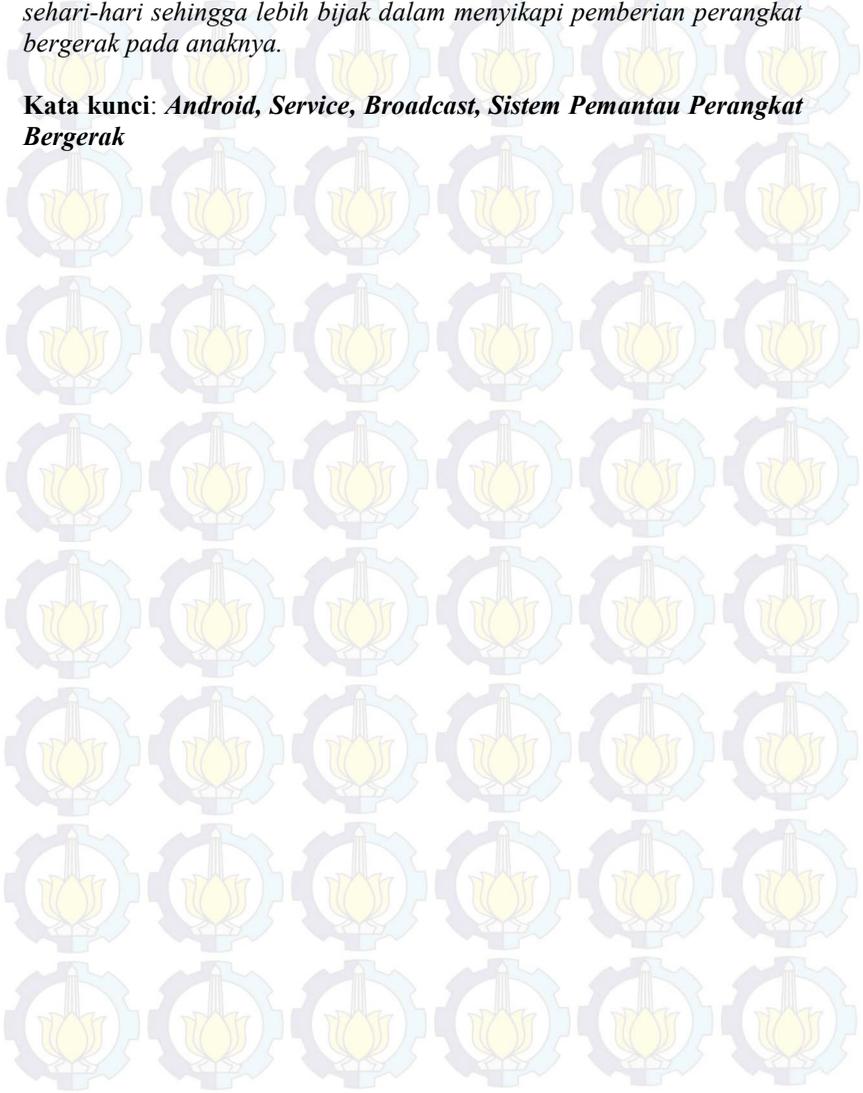
Dewasa ini semakin banyak orang tua yang memberikan perangkat bergerak berupa ponsel cerdas atau tablet kepada anaknya. Ini dilakukan untuk kemudahan komunikasi antara keduanya, maupun sebagai digital babysitter supaya anak lebih tenang. Namun, perlu disadari juga bahwa hal ini dapat membuat anak mengalami ketergantungan terhadap layanan yang diberikan perangkat bergerak tersebut. Layanan-layanan tersebut antara lain video game, pesan singkat, sampai browser. Sebagai orang tua yang memiliki kesibukan dan tidak mungkin mengawasi aktivitas anak dengan perangkat bergeraknya, diperlukan sebuah sistem pemantau perangkat bergerak yang mampu menginformasikan hal ini kepada para orang tua secara rutin.

Sistem pemantau aktivitas perangkat bergerak yang dibuat pada tugas akhir ini terdiri dari sebuah aplikasi perangkat bergerak dan sebuah aplikasi yang berjalan pada server. Aplikasi pada perangkat bergerak ini memanfaatkan layanan service dan broadcast yang disediakan oleh sistem operasi android. Service ini akan berjalan selama perangkat aktif dan berfungsi untuk mencatat seluruh aktivitas perangkat bergerak dan menampilkan hasil record yang telah disimpan. Sementara layanan broadcast digunakan untuk beberapa hal seperti menangkap telepon masuk maupun keluar dan pesan singkat yang masuk ke perangkat. Selain itu, aplikasi ini juga akan mencatat lokasi-lokasi perangkat tersebut berada sehingga orang tua tahu dimana anak beserta perangkatnya berada selama sehari penuh.

Aplikasi perangkat bergerak kemudian mengirimkan data-data yang sudah dicatat ke aplikasi kedua yang berjalan pada server. Nantinya, aplikasi kedua meneruskan data-data tersebut ke alamat

email orang tua. Diharapkan dengan data-data yang dihasilkan oleh sistem ini, orang tua dapat lebih mudah mengevaluasi kegiatan anak sehari-hari sehingga lebih bijak dalam menyikapi pemberian perangkat bergerak pada anaknya.

Kata kunci: *Android, Service, Broadcast, Sistem Pemantau Perangkat Bergerak*



Sistem Pemantau Orang Tua Terhadap Pemakaian Perangkat Bergerak Anak.

Student Name : Baskara Nur Patria
NRP : 5111 100 097
Major : Informatics Engineering FTIf-ITS
Advisor I : Dr.Tech. Ir. R.V. Hari Ginardi, M.Sc.
Advisor II : Ir. F.X. Arunanto, M.Sc.

ABSTRACT

Today, more parents are giving mobile device such as smartphone or tablet to their children. They are doing to facilitate communication between them and as a digital babysitting tool to calm their children. However, we need to realize that this can make children addicted to services or applications that provided by a gadget such as video game, short message service, and browser. As parents who are busy and do not really have time for monitoring children's activities with their gadget, we need a mobile device usage monitoring system that is able to inform their activities to parents regularly.

Mobile device usage monitoring system that developed in this final project consists of a mobile application and a web service application. The mobile application uses system service and broadcast that provided by android. This service will run every time and . Service ini akan berjalan selama perangkat aktif dan serves to record all device's activities and show the records that have been stored. The broadcast service is used for listening some activities such as incoming call, outgoing call, and incoming sms. This application is also able to record the device's location. So, parents will know their children position all day long.

The mobile application will send records to second application that runs in a server. Later, the second application will forward those records to registered parent's email address. Hopefully, parents can evaluate their children's activities easier with their gadget using records that provided by this system.

Keywords: Android, Service, Broadcast, Gadget Monitoring System

KATA PENGANTAR

Puji syukur kepada Allah Yang Maha Esa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan tugas akhir yang berjudul:

Sistem Pemantau Pemakaian Perangkat Bergerak Anak

Melalui lembar ini, penulis ingin menyampaikan ucapan terima kasih dan penghormatan yang sebesar-besarnya kepada:

1. Allah SWT atas segala nikmat dan rahmat yang telah diberikan selama ini.
2. Ibu penulis yang selalu mencurahkan kasih sayang, dukungan, doa dan selalu menjadi motivasi bagi penulis untuk selalu berusaha sebaik-baiknya.
3. Saudara kandung, saudara ipar, dan keponakan penulis yang selalu menjadi motivasi kepada penulis dalam menuntut ilmu dan menyelesaikan buku ini.
4. Bapak Dr.tech. Ir. R. V. Hari Ginardi, M.Sc. dan Bapak Ir. F.X. Arunanto, M.Sc. selaku dosen pembimbing yang telah memberikan arahan, ilmu, motivasi, dan meluangkan waktu untuk membantu pengerjaan tugas akhir ini.
5. Ibu Umi Laili Yuhana selaku dosen wali penulis yang telah meluangkan waktu untuk konsultasi mengenai masalah-masalah akademik.
6. Bapak dan Ibu dosen yang telah membina dan memberikan ilmu kepada penulis selama menempuh studi di Teknik Informatika ITS.
7. Teman-teman yang telah membantu penulis ketika sedang mengalami hambatan teknis dalam pengerjaan

tugas akhir dan hal lainnya: Ali, Indra, Galih, Wilik, Andy, Atnis, Tania, dan Novandi.

8. Teman-teman yang telah memberikan referensi buku tugas akhir kepada penulis: Ghani, Monika, Helmy, dan Fahmi.
9. Teman-teman yang pernah satu kelompok pada tugas kuliah di setiap semester.
10. Keluarga besar angkatan TC 2011 yang menemani keseharian penulis dalam kehidupan kampus.
11. Pihak-pihak lain yang namanya tidak dapat penulis sebutkan satu -persatu.

Bagaimanapun juga penulis telah berusaha semaksimal mungkin dalam menyusun tugas akhir ini, namun penulis mohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, Juni 2015

Baskara Nur Patria

DAFTAR ISI

ABSTRAK	i
ABSTRACT	iii
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Permasalahan	2
1.3 Batasan Permasalahan	3
1.4 Tujuan dan Manfaat	3
1.5 Metodologi	4
1.6 Sistematika	5
BAB II TINJAUAN PUSTAKA	7
2.1 Sistem Pemantau Pemakaian <i>Gadget</i>	7
2.2 Android SDK	8
2.3 MPAndroidChart	9
2.4 <i>Reverse Geocoding</i>	9
2.5 Basis Data SQLite	9
2.6 <i>JavaScript Object Notation</i>	10
2.7 <i>Content Provider</i>	10
2.8 <i>Location Based Services</i>	11

2.9	<i>Webservice</i>	11
BAB III ANALISIS DAN PERANCANGAN.....		13
3.1.	Analisis.....	13
3.1.1.	Analisis Permasalahan.....	13
3.1.2.	Deskripsi Umum Perangkat Lunak.....	14
3.1.3.	Kebutuhan Fungsional Aplikasi Android.....	15
3.1.4.	Kebutuhan Fungsional Aplikasi <i>Webservice</i>	23
3.2.	Perancangan.....	25
3.2.1.	Perancangan Umum Arsitektur Sistem.....	25
3.2.2.	Perancangan Basis Data.....	26
3.2.3.	Perancangan Algoritma.....	33
3.2.4.	Perancangan Antarmuka Sistem.....	36
BAB IV IMPLEMENTASI.....		49
4.1.	Lingkungan Implementasi.....	49
4.2.	Implementasi Antarmuka.....	49
4.3.	Implementasi Proses Pada Aplikasi Perangkat Bergerak.....	59
4.3.1.	Mencatat Aplikasi yang Dibuka.....	59
4.3.2.	Mencatat Lokasi Pengguna.....	62
4.3.3.	Mencatat Panggilan Telepon.....	63
4.3.4.	Mencatat Pesan Singkat Masuk.....	65
4.3.5.	Mencatat Pesan Singkat Keluar.....	65
4.3.6.	Mencatat Alamat Situs yang Dikunjungi.....	67
4.3.7.	Menampilkan <i>Record</i>	68
4.3.8.	Mengatur Waktu Pengiriman <i>Record</i>	69

4.3.9. Mengirim <i>Record</i>	69
4.3.9.1. Memeriksa Jaringan	69
4.3.9.2. Membuat Struktur JSON	70
4.3.9.3. Mengirimkan JSON.....	71
4.3.10. Mengubah Akun.....	72
4.3.11. Mendaftarkan dan <i>Login</i> Akun.....	72
4.3.12. Menghitung Statistik	73
4.3.12.1. Aplikasi	73
4.3.12.2. Panggilan Masuk dan Keluar.....	74
4.3.12.3. Pesan Singkat Masuk dan Keluar	74
4.3.12.4. Durasi	74
4.3.13. Menampilkan Statistik.....	75
4.3.14. Menambahkan Favorit.....	76
4.3.15. Menghapus Favorit.....	77
4.4. Implementasi Proses Pada Aplikasi <i>Webservice</i>	77
4.5. Implementasi Tampilan <i>Report</i>	79
BAB V PENGUJIAN DAN EVALUASI	81
5.1. Lingkungan Pelaksanaan Pengujian	81
5.2. Pengujian Fungsionalitas.....	81
5.2.1. Pengujian Melakukan Pendaftaran dan <i>Login</i> Akun	82
5.2.2. Pengujian Pencatatan Aplikasi	85
5.2.3. Pengujian Pencatatan Pesan Singkat Masuk	86
5.2.4. Pengujian Pencatatan Pesan Singkat Keluar	88
5.2.5. Pengujian Pencatatan Panggilan Telepon Masuk.....	89

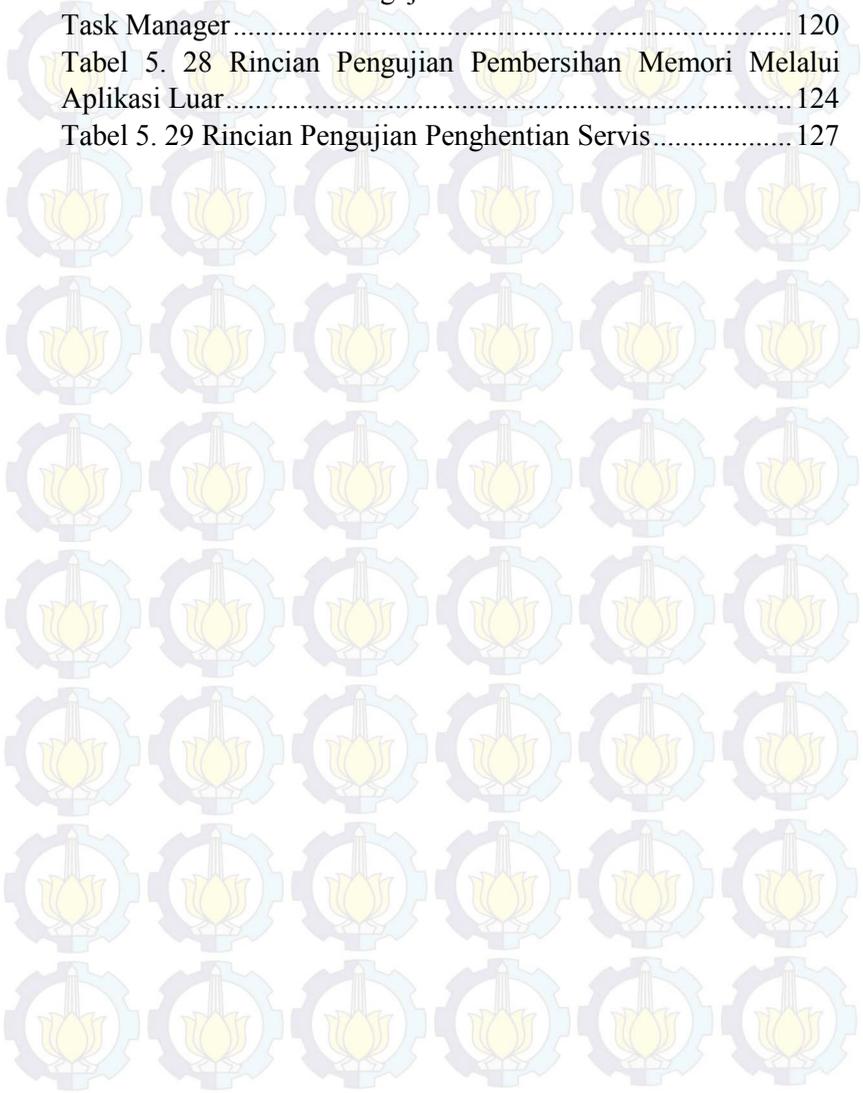
5.2.6. Pengujian Pencatatan Panggilan Telepon Keluar	92
5.2.7. Pengujian Pencatatan Alamat Situs	94
5.2.8. Pengujian Pencatatan Lokasi	95
5.2.9. Pengujian Menampilkan Statistik	97
5.2.10. Pengujian Ubah Akun	104
5.2.11. Pengujian Atur Jadwal dan Pengiriman <i>Record</i>	109
5.2.12. Pengujian Penambahan Aplikasi Favorit	113
5.2.13. Pengujian Penambahan <i>Domain</i> Favorit	114
5.2.14. Pengujian Penghapusan Aplikasi Favorit	116
5.2.15. Pengujian Penghapusan <i>Domain</i> Favorit	117
5.3. Pengujian Non Fungsionalitas	119
5.3.1. Pembersihan Memori Melalui <i>Task Manager</i>	119
5.3.2. Pembersihan Memori Melalui Aplikasi Luar	124
5.3.3. Penghentian Servis Melalui <i>Application Manager</i>	127
5.3.4. Konsumsi Memori	129
5.4. Evaluasi Pengujian	133
5.4.1. Evaluasi Pengujian Fungsionalitas	133
5.4.2. Evaluasi Pengujian Non Fungsionalitas	136
BAB VI PENUTUP	139
5.1. Kesimpulan	139
5.2. Saran	140
DAFTAR PUSTAKA	141
BIODATA PENULIS	143

DAFTAR TABEL

Tabel 3. 1 Daftar Kasus Penggunaan Aplikasi Perangkat Bergerak	19
Tabel 3. 2 Daftar Kasus Penggunaan Aplikasi <i>Webservice</i>	24
Tabel 3. 3 Rincian Tabel Pengguna.....	26
Tabel 3. 4 Rincian Tabel SMS Masuk	27
Tabel 3. 5 Rincian Tabel SMS Keluar	27
Tabel 3. 6 Rincian Tabel Telepon Masuk	28
Tabel 3. 7 Rincian Tabel Telepon Keluar	29
Tabel 3. 8 Rincian Tabel Aplikasi.....	29
Tabel 3. 9 Rincian Tabel Lokasi	30
Tabel 3. 10 Rincian Tabel Situs	30
Tabel 3. 11 Rincian Tabel Aplikasi Favorit	31
Tabel 3. 12 Rincian Tabel Situs Favorit.....	31
Tabel 3. 13 Rincian Tabel Alarm	32
Tabel 3. 14 Rincian Tabel SMS Sementara.....	32
Tabel 3. 15 Keterangan Halaman Antarmuka Daftar	36
Tabel 3. 16 Keterangan Halaman Antarmuka Login.....	38
Tabel 3. 17 Keterangan Halaman Antarmuka Overview	39
Tabel 3. 18 Keterangan Halaman Antarmuka Edit Akun.....	41
Tabel 3. 19 Keterangan Halaman Antarmuka Statistik	43
Tabel 3. 20 Keterangan Halaman Antarmuka Atur Jadwal.....	44
Tabel 3. 21 Keterangan Halaman Antarmuka Menambahkan Favorit	46
Tabel 3. 22 Keterangan Halaman Antarmuka Menghapus Favorit	47
Tabel 5. 1 Spesifikasi Perangkat Bergerak.....	81
Tabel 5. 2 Rincian Pengujian Melakukan Pendaftaran	82
Tabel 5. 3 Rincian Pengujian Melakukan Login	84
Tabel 5. 4 Rincian Pengujian Melakukan Pencatatan Aplikasi... ..	85

Tabel 5. 5 Rincian Pengujian Melakukan Pencatatan SMS Masuk	87
Tabel 5. 6 Rincian Pengujian Melakukan Pencatatan SMS Keluar	88
Tabel 5. 7 Rincian Pengujian Pencatatan Panggilan Masuk dan Diangkat	90
Tabel 5. 8 Rincian Pengujian Pencatatan Missed Call	91
Tabel 5. 9 Rincian Pengujian Pencatatan Panggilan Keluar.....	92
Tabel 5. 10 Rincian Pengujian Pencatatan Alamat Situs.....	94
Tabel 5. 11 Rincian Pengujian Melakukan Pencatatan Lokasi....	96
Tabel 5. 12 Rincian Pengujian Menampilkan Statistik Aplikasi.....	97
Tabel 5. 13 Rincian Pengujian Menampilkan Statistik Panggilan Masuk	98
Tabel 5. 14 Rincian Pengujian Menampilkan Statistik Panggilan Keluar.....	100
Tabel 5. 15 Rincian Pengujian Menampilkan Statistik SMS Masuk	101
Tabel 5. 16 Rincian Pengujian Menampilkan Statistik SMS Keluar.....	103
Tabel 5. 17 Rincian Pengujian Mengubah Alamat Email	104
Tabel 5. 18 Rincian Pengujian Mengubah Password	106
Tabel 5. 19 Rincian Pengujian Mengubah Alamat Email dan Password.....	107
Tabel 5. 20 Rincian Pengujian Pengiriman Record Saat Terhubung Internet.....	109
Tabel 5. 21 Rincian Pengujian Pengiriman Record Saat Tidak Terhubung Internet	111
Tabel 5. 22 Rincian Pengujian Pengiriman Record Saat Mode Pesawat Terbang.....	112
Tabel 5. 23 Rincian Pengujian Menambahkan Aplikasi Favorit	113
Tabel 5. 24 Rincian Pengujian Menambahkan Domain Favorit.....	115
Tabel 5. 25 Rincian Pengujian Menghapus Aplikasi Favorit.....	116

Tabel 5. 26 Rincian Pengujian Menghapus Domain Favorit 118
Tabel 5. 27 Rincian Pengujian Pembersihan Memori Melalui Task Manager..... 120
Tabel 5. 28 Rincian Pengujian Pembersihan Memori Melalui Aplikasi Luar..... 124
Tabel 5. 29 Rincian Pengujian Penghentian Servis..... 127

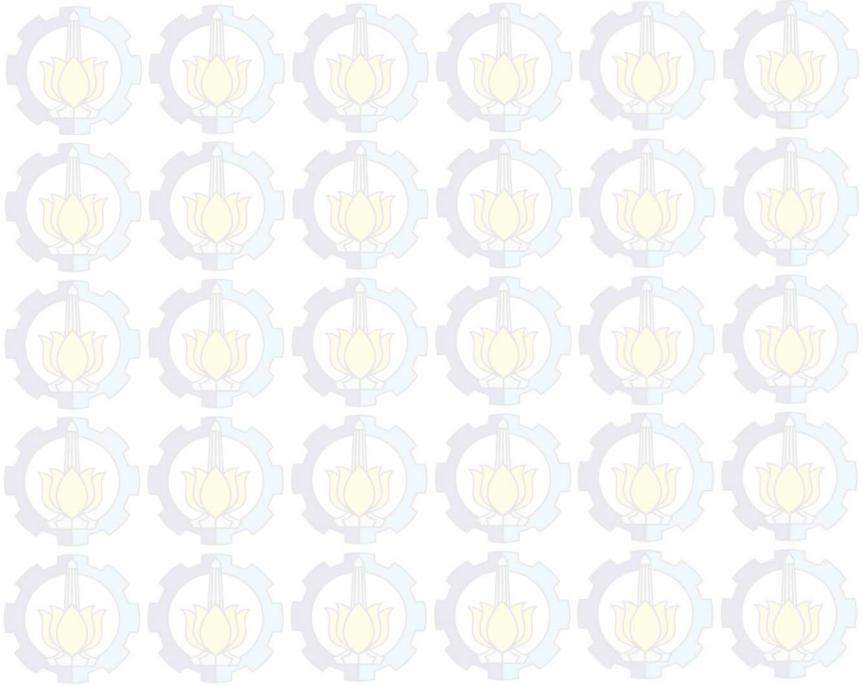


DAFTAR GAMBAR

Gambar 3. 1 Diagram Modul Aplikasi Perangkat Bergerak	17
Gambar 3. 2 Diagram Kasus Penggunaan Aplikasi Perangkat Bergerak	18
Gambar 3. 3 Diagram Kasus Penggunaan Aplikasi Webservice	24
Gambar 3. 4 Arsitektur Umum Sistem	26
Gambar 3. 5 Alur Kerja Aplikasi Perangkat Bergerak	33
Gambar 3. 6 Alur Kerja Servis Utama	34
Gambar 3. 7 Alur Kerja Proses Pengiriman Data	35
Gambar 3. 8 Rancangan Antarmuka Halaman Daftar	36
Gambar 3. 9 Rancangan Antarmuka Halaman Login	38
Gambar 3. 10 Rancangan Antarmuka Halaman Overview	39
Gambar 3. 11 Rancangan Antarmuka Halaman Edit Akun	41
Gambar 3. 12 Rancangan Antarmuka Halaman Statistik	42
Gambar 3. 13 Rancangan Antarmuka Halaman Atur Jadwal	44
Gambar 3. 14 Rancangan Antarmuka Halaman Menambahkan Favorit	45
Gambar 3. 15 Rancangan Antarmuka Halaman Menghapus Favorit	47
Gambar 4. 1 Antarmuka Halaman Daftar	50
Gambar 4. 2 Antarmuka Halaman Login	51
Gambar 4. 3 Antarmuka Halaman Overview	52
Gambar 4. 4 Antarmuka Halaman Overview – Aplikasi	52
Gambar 4. 5 Antarmuka Halaman Overview - Lokasi	53
Gambar 4. 6 Antarmuka Halaman Overview - Telepon Masuk	53
Gambar 4. 7 Antarmuka Halaman Overview - Telepon Keluar	54
Gambar 4. 8 Antarmuka Halaman Overview - Pesan Masuk	54
Gambar 4. 9 Antarmuka Halaman Overview - Pesan Keluar	55
Gambar 4. 10 Antarmuka Halaman Overview - Browser	55
Gambar 4. 11 Antarmuka Halaman Edit Akun	56

Gambar 4. 12 Antarmuka Halaman Statistik.....	56
Gambar 4. 13 Antarmuka Halaman Atur Jadwal.....	57
Gambar 4. 14 Antarmuka Halaman Menambah Favorit	58
Gambar 4. 15 Antarmuka Halaman Menghapus Favorit.....	58
Gambar 4. 16 Implementasi <i>Report</i>	80
Gambar 5. 1 Hasil Pengujian PA-01	83
Gambar 5. 2 Hasil Pengujian PA-02	84
Gambar 5. 3 Hasil Pengujian PA-03	86
Gambar 5. 4 Hasil Pengujian PA-04	88
Gambar 5. 5 Hasil Pengujian PA-05	89
Gambar 5. 6 Hasil Pengujian PA-06	91
Gambar 5. 7 Hasil Pengujian PA-07	92
Gambar 5. 8 Hasil Pengujian PA-08	94
Gambar 5. 9 Hasil Pengujian PA-09	95
Gambar 5. 10 Hasil Pengujian PA-10	97
Gambar 5. 11 Hasil Pengujian PA-11	98
Gambar 5. 12 Hasil Pengujian PA-12	100
Gambar 5. 13 Hasil Pengujian PA-13	101
Gambar 5. 14 Hasil Pengujian PA-14	102
Gambar 5. 15 Hasil Pengujian PA-15	104
Gambar 5. 16 Hasil Pengujian PA-16	106
Gambar 5. 17 Hasil Pengujian PA-17	107
Gambar 5. 18 Hasil Pengujian PA-18	109
Gambar 5. 19 Hasil Pengujian PA-19	111
Gambar 5. 20 Hasil Pengujian PA-22	114
Gambar 5. 21 Hasil Pengujian PA-23	116
Gambar 5. 22 Hasil Pengujian PA-24	117
Gambar 5. 23 Hasil Pengujian PA-25	119
Gambar 5. 24 Proses Pembersihan Memori Dari Task Manager	121
Gambar 5. 25 Notifikasi Foreground Service.....	122

Gambar 5. 26 Tampilan Notifikasi Foreground Service Berjalan	123
Gambar 5. 27 Tampilan Menu Application Manager	124
Gambar 5. 28 Tampilan Clear Master	125
Gambar 5. 29 Notifikasi Foreground Service.....	126
Gambar 5. 30 Tampilan Application Manager.....	127
Gambar 5. 31 Tampilan Saat Akan Dihentikan.....	128
Gambar 5. 32 Tampilan Setelah Dihentikan	129
Gambar 5. 33 Konsumsi Memori < 1 Jam	130
Gambar 5. 34 Konsumsi Memori 2 Jam.....	131
Gambar 5. 35 Konsumsi Memori 7 Jam.....	132
Gambar 5. 36 Konsumsi Memori 9 Jam.....	133



BIODATA PENULIS

Penulis lahir di Kediri, 17 Agustus 1993. Sebelum menempuh jenjang pendidikan sarjana di Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya, penulis menempuh pendidikan di SD Pawyatan Daha 2, SMP Negeri 1 Kediri, dan SMA Negeri 1 Kediri. Dalam menyelesaikan pendidikan S1, penulis mengambil bidang minat algoritma pemrograman. Selama kuliah penulis aktif dalam beberapa kegiatan organisasi.

Pada bidang nonakademik, penulis magang sebagai staff di departemen minat dan bakat Himpunan Mahasiswa Teknik Computer Informatika ITS (HMTC) pada periode kepengurusan 2011-2012. Selama 2 tahun berikutnya penulis aktif sebagai staf dan staf ahli departemen pengembangan sumber daya mahasiswa di HMTC. Penulis juga sempat menjabat wakil koordinator REEVA Schematics 2013. Penulis memiliki ketertarikan pada pembuatan aplikasi perangkat bergerak khususnya berbasis android dan ios, dan sistem *enterprise*. Pada bidang non-teknis penulis memiliki ketertarikan pada bisnis *start-up*, sepak bola, dan travelling. Penulis dapat dihubungi melalui alamat *email* [baskarapatrila17@gmail\[dot\]com](mailto:baskarapatrila17@gmail.com).

BAB I PENDAHULUAN

1.1 Latar Belakang

Saat ini semakin banyak orang tua yang memberikan perangkat bergerak berupa *smartphone* maupun *tablet* kepada anaknya pada usia dini. Hal ini pada awalnya dilakukan untuk kemudahan komunikasi ketika orang tua sedang tidak bersama anaknya. Seiring berjalannya waktu, situs *theAsianparent Indonesia* menuliskan bahwa penggunaan perangkat bergerak pada anak yang berlebihan memiliki dampak negatif sehingga perlu dibatasi. Diantaranya gangguan tidur, terlambat bicara bagi balita, sikap pasif, gangguan bagi perkembangan mata, sulit konsentrasi, dan kecanduan¹. Oleh karena itu, orang tua harus tetap memberikan pengawasan terhadap penggunaan perangkat bergerak milik anaknya sehingga mereka tahu frekuensi kegiatan yang digunakan anak terhadap perangkat bergerak.

Perkembangan seorang anak bergantung sikap orang tua sejak dini, salah satunya adalah dalam memberikan, membimbing, dan mengawasi anak dalam penggunaan media informasi dan teknologi secara terarah dan terpadu. Untuk menentukan sikap orang tua terhadap anaknya secara tepat, diperlukan sebuah *report* pemakaian perangkat bergerak oleh anak secara berkala dan teratur. Dengan *report* ini, orang tua dapat lebih memahami kegiatan anak ketika mengoperasikan perangkat bergerak miliknya sehingga orang tua diharapkan tidak salah dalam mengambil keputusan terhadap perilaku anaknya. Oleh sebab itu, diperlukan aplikasi yang berjalan sebagai *background service* pada sebuah perangkat bergerak yang dapat mencatat seluruh aktivitas pengguna dengan perangkat tersebut. Aplikasi ini harus selalu berjalan menggunakan sumber daya yang

¹ <http://id.theasianparent.com/baikkah-memberikan-tablet-untuk-anak-balita/>

dimiliki perangkat dengan efektif sehingga tidak mengganggu kinerja perangkat ketika digunakan untuk aktivitas lain.

Dalam tugas akhir ini, dibuat aplikasi yang memberikan data serta statistik pemakaian perangkat bergerak untuk mengevaluasi pemakaian perangkat oleh anak. Data-data yang akan diproduksi adalah catatan waktu pemakaian telepon (baik panggilan masuk maupun keluar) beserta nomor yang dituju, pesan singkat masuk dan keluar beserta nomor yang dituju, aplikasi yang dibuka beserta waktu dan lokasinya, dan situs-situs yang dikunjungi oleh *browser* perangkat tersebut. Sementara statistik yang akan dihasilkan dari data yang ada adalah nomor telepon yang paling sering dihubungi dan menghubungi anak melalui panggilan maupun pesan singkat serta aplikasi-aplikasi yang paling sering dibuka oleh anak sehingga orang tua dapat mengawasi apakah anaknya kecanduan terhadap suatu aplikasi. Seluruh *record* beserta statistik pemakaian perangkat yang dihasilkan akan dikirimkan secara berkala dan terjadwal ke alamat *email* yang dimasukkan oleh orang tua. Diharapkan dengan adanya data yang dihasilkan oleh aplikasi ini, orang tua akan lebih mudah menentukan sikap terhadap perilaku anak dalam menggunakan perangkat bergerak.

1.2 Rumusan Permasalahan

Rumusan masalah yang diangkat dalam tugas akhir ini dipaparkan sebagai berikut.

1. Bagaimana cara kerja *service*, *broadcast*, dan *asynchronous task* pada perangkat bergerak.
2. Bagaimana cara untuk membuat *service* selalu berjalan pada latar belakang sistem.
3. Bagaimana cara merancang dan membangun sistem yang mampu memantau aktivitas perangkat bergerak.
4. Bagaimana cara merancang dan membangun sistem yang mampu memproduksi statistik pemakaian perangkat bergerak.

1.3 Batasan Permasalahan

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, di antaranya sebagai berikut.

1. Aplikasi dibangun khusus untuk perangkat bergerak dengan sistem operasi Android dengan versi 4.x.x.
2. Aplikasi membutuhkan koneksi internet untuk mengirimkan *record*.
3. Aplikasi memanfaatkan koneksi internet untuk memperoleh posisi lintang dan bujur.
4. API Google *Geocoding* yang dipakai merupakan versi tidak berbayar.
5. Data alamat situs yang dicatat merupakan alamat situs yang dikunjungi menggunakan *browser native* perangkat.

1.4 Tujuan dan Manfaat

Tujuan dari pembuatan tugas akhir ini adalah sebagai berikut.

1. Mengetahui cara kerja *service*, *broadcast*, dan *asynchronous task* pada android.
2. Merancang cara untuk membuat *service* selalu berjalan di latar belakang sistem.
3. Merancang dan membangun sistem yang mampu memantau aktivitas perangkat bergerak.
4. Merancang dan membangun sistem yang mampu memproduksi statistik pemakaian perangkat bergerak.

Sementara manfaat dari pembuatan tugas akhir ini antara lain sebagai berikut.

1. Memberikan data aktivitas perangkat bergerak secara teratur.
2. Membantu orang tua dalam mengevaluasi kegiatan anak selama menggunakan perangkat bergerak berdasarkan *record* serta hasil statisik yang dikirim ke alamat *email* orang tua.

1.5 Metodologi

Langkah yang ditempuh dalam pengerjaan tugas akhir ini adalah sebagai berikut.

1. Studi Literatur

Tahap ini merupakan tahap pengumpulan informasi dan pembelajaran yang akan digunakan pada tugas akhir ini. Studi literatur meliputi diskusi dan pemahaman mengenai topik tugas akhir ini, diantaranya mengenai:

- a. *Service*, *Broadcast*, dan *Asynchronous Task* pada android.
- b. *Content Provider* pada android.
- c. Proses komunikasi data antara aplikasi perangkat perangkat bergerak yang berbasis android dengan *webservice* yang dibangun menggunakan PHP.
- d. Eksplorasi *library* untuk pembuatan diagram pada android.

2. Perancangan Sistem

Tahap ini merupakan tahap perancangan sistem dengan menggunakan studi literatur dan mempelajari konsep aplikasi yang akan dibuat. Dengan berbekal teori, metode, dan informasi yang sudah terkumpul pada tahap sebelumnya diharapkan akan membantu dalam proses perancangan sistem.

3. Implementasi

Tahap ini dilakukan pembuatan perangkat lunak berdasarkan rancangan yang telah dibuat pada proses sebelumnya. Rincian pada tahap ini sebagai berikut.

- a. Implementasi rancangan basis data.
- b. Implementasi *webservice* yang berfungsi untuk menangani komunikasi aplikasi perangkat bergerak dengan *server*.
- c. Implementasi rancangan aplikasi perangkat bergerak.

4. Pengujian dan Evaluasi

Pengujian aplikasi ini dilakukan dengan cara sebagai berikut.

- a. Uji coba pemakaian pada perangkat ketika terhubung dan tidak terhubung dengan internet.
- b. Uji coba penggunaan aplikasi pada lokasi yang berbeda.
- c. Uji coba penggunaan aplikasi pada kondisi *airplane mode*.
- d. Melakukan pengecekan kebenaran data serta statistik yang dihasilkan oleh aplikasi dengan kejadian yang sebenarnya terjadi.
- e. Pengecekan konsumsi aplikasi terhadap memori karena jika berlebihan akan sangat mempengaruhi performa perangkat.

5. Penyusunan Buku Tugas Akhir

Pada tahap ini ditulis buku yang bertujuan untuk mendokumentasikan seluruh konsep, rancangan, dasar teori, literatur, proses yang dilakukan dan hasil yang diperoleh selama pengerjaan tugas akhir. Buku yang ditulis bertujuan untuk memberikan gambaran dari pengerjaan tugas akhir dan berguna untuk pembaca yang tertarik untuk melakukan pengembangan sistem lebih lanjut.

1.6 Sistematika

Buku tugas akhir ini disusun dengan sistematika yang terdiri dari beberapa bagian, yaitu:

Bab I Pendahuluan

Bab ini membahas latar belakang masalah, tujuan pembuatan tugas akhir, rumusan permasalahan, batasan permasalahan, metodologi yang digunakan, dan sistematika penyusunan tugas akhir.

Bab II Tinjauan Pustaka

Bab ini membahas teori pendukung dan literatur yang berkaitan dengan bahasan dan mendasari pembuatan Tugas Akhir ini.

Bab III Analisis dan Perancangan

Bab ini membahas tentang desain dan rancangan dari perangkat lunak.

Bab IV Implementasi

Bab ini membahas tentang implementasi hasil analisis dan perancangan dalam bentuk *pseudocode*. Bab ini membahas proses pembangunan perangkat lunak.

Bab V Pengujian dan Evaluasi

Bab ini membahas tentang pengujian aplikasi berdasarkan skenario yang telah ditentukan. Mengevaluasi fitur aplikasi apakah telah memenuhi kebutuhan fungsional.

Bab V Kesimpulan dan Saran

Bab ini berisi kesimpulan baik dari proses pengembangan perangkat lunak dan hasil uji coba.

Daftar Pustaka

Merupakan daftar referensi yang digunakan dalam pembuatan Tugas Akhir.

BAB II

TINJAUAN PUSTAKA

Pada bab ini dibahas mengenai dasar teori dan literatur yang menjadi dasar pembuatan tugas akhir. Dasar teori yang dibahas di dalam bab ini adalah kajian mengenai sistem operasi android dan *library* MPAndroidChart. Selain itu juga dibahas mengenai basis data SQLite dan struktur data JSON yang digunakan untuk pertukaran data serta beberapa hal lainnya.

2.1 Sistem Pemantau Pemakaian Gadget

Aplikasi pemantau penggunaan perangkat bergerak merupakan aplikasi yang digunakan untuk membantu pengguna memantau pemakaian perangkat bergerak. Bagi orang tua, dengan aplikasi pemantau berarti mengetahui aktivitas anak tanpa memblokir hak aksesnya.

Saat ini ada beberapa aplikasi yang dapat digunakan untuk memantau maupun mengatur pemakaian perangkat bergerak. Situs *tomsguide*, telah merilis beberapa aplikasi pengatur dan pemantau perangkat bergerak anak oleh orang tua yang dianggap terbaik saat ini, antara lain: PhoneSheriff, Qustodio, Net Nanny, My Mobile Watchdog dan Mobile Spy (Hall, 2015). Aplikasi-aplikasi ini memiliki fitur-fitur yang berbeda satu sama lain. Sebagai aplikasi yang dianggap terbaik, PhoneSheriff mampu memantau aplikasi apa saja yang ada di perangkat, pesan singkat yang masuk dan keluar, situs internet yang dikunjungi, serta lokasi anak. Karena konsepnya merupakan aplikasi pengatur oleh orang tua, maka aplikasi ini memungkinkan orang tua untuk memblokir aplikasi lain di perangkat milik anak dan memblokir nomor telepon. Sementara aplikasi lain dari yang disebutkan di atas rata-rata memiliki beberapa fitur yang dimiliki oleh PhoneSheriff, namun tidak selengkap PhoneSheriff.

Aplikasi-aplikasi tersebut tidak dirancang secara khusus untuk memantau aktivitas anak dalam menggunakan perangkat bergerak. Hal ini tercermin pada tidak mampunya aplikasi-

aplikasi tersebut mencatat aplikasi-aplikasi apa saja yang dibuka oleh anak ketika menggunakan perangkat bergerak miliknya.

2.2 Android SDK

Android SDK merupakan perangkat lunak yang digunakan oleh para pengembang untuk membangun aplikasi pada *platform* android. Android SDK menyediakan API yang dibutuhkan oleh pengembang perangkat lunak untuk membuat aplikasi android.

Pada aplikasi yang menjadi tugas akhir ini, eksplorasi dilakukan pada penggunaan *service* dimana kelas yang mengekstensi kelas abstrak *Service* akan menjalankan proses secara terus menerus di balik layar². Normalnya ketika memori pada perangkat tersisa sedikit, sistem operasi akan secara otomatis mematikan servis-servis tersebut. Oleh karena aplikasi pada tugas akhir ini membutuhkan suatu servis yang harus berjalan terus menerus, diperlukan suatu sistem atau cara yang mencegah servis tersebut dihentikan secara paksa oleh sistem operasi ketika memori perangkat tersisa sedikit.

Android memiliki sistem *broadcast* yang dapat digunakan oleh pengembang ingin suatu proses atau komputasi ketika perangkat dalam *state* tertentu³. Untuk mendapatkan suatu *broadcast* dari sistem, pengembang harus mengimplementasikan fungsi khusus pada suatu kelas sehingga kelas tersebut akan menerima suatu *trigger* dari *broadcast* yang dikeluarkan oleh sistem. Perbedaan paling mendasar kelas ini dengan kelas-kelas yang mengekstensi kelas lainnya adalah cara pemanggilannya.

Android juga memiliki sistem yang membuat suatu kelas abstrak bernama *AsyncTask* dimana kelas yang mengekstensi kelas ini akan bersifat seperti servis, namun dirancang agar digunakan untuk mengeksekusi atau melakukan komputasi sebanyak satu kali.

² <http://developer.android.com/reference/android/app/Service.html>

³ <http://developer.android.com/reference/android/content/BroadcastReceiver.html>

2.3 MPAndroidChart

MPAndroidChart⁴ merupakan *library* yang dikembangkan oleh Philipp Jahoda untuk membuat tampilan grafis berupa diagram pada perangkat android. Jenis diagram yang dapat dibuat menggunakan *library* ini antara lain diagram garis, batang, pencar, *candlestick*, pai atau lingkaran, dan jaring laba-laba atau radar. *Library* ini juga mendukung animasi pada diagram, kustomisasi warna dan ukuran sesuai keinginan pengembang aplikasi. Semua aplikasi yang berjalan pada android versi 2.2 ke atas dapat menggunakan fitur-fitur *library* ini.

2.4 Reverse Geocoding

Geocoding merupakan proses melakukan konversi sebuah alamat (misal 1600 Amphitheatre Parkway, Mountain View, CA) menjadi data koordinat geografis (misal lintang 37.423021 dan bujur -122.083739). Sementara *reverse geocoding* merupakan kebalikannya, dimana koordinat geografis diterima sebagai input dan dihitung alamat fisik terdekat dari koordinat tersebut (Stenneth, Wolfson, Yu, & Xu, 2011). Fitur *reverse geocoding* pada tugas akhir ini menggunakan API milik Google dan diimplementasikan pada *webservice* sistem.

2.5 Basis Data SQLite

Basis data SQLite⁵ merupakan basis data yang berukuran kecil, berdiri sendiri, bukan berupa basis data *client-server*, tanpa konfigurasi khusus, namun memiliki fitur penuh perintah SQL. SQLite merupakan produk *public domain*, yang artinya setiap orang dapat menggunakannya secara bebas. Setiap android memakai basis data SQLite, dimana fitur ini terdapat pada *package* android.database.sqlite. Perangkat berbasis android akan memakai SQLite versi 3.4.0.

⁴ <https://github.com/PhilJay/MPAndroidChart>

⁵ <http://www.sqlite.org/about.html>

2.6 *JavaScript Object Notation*

JSON (*JavaScript Object Notation*)⁶ adalah format pertukaran data yang ringan. JSON mudah untuk dibaca dan ditulis oleh manusia dan memudahkan mesin untuk mengurai dan menciptakan. Format ini berdasar dari himpunan bagian bahasa pemrograman JavaScript standar ECMA-262 edisi ke-3, Desember 1999. JSON merupakan format teks yang benar – benar berbeda atau independen namun tetap menggunakan konvensi yang mudah dikenali bagi *programmer* yang sudah terbiasa dengan bahasa C, C++, C#, Java, JavaScript, Perl, Python dan bahasa lainnya. Sifat JSON yang seperti ini menjadikannya format pertukaran data yang ideal.

2.7 *Content Provider*

*Content Provider*⁷ pada android merupakan suatu layanan yang mengatur akses suatu aplikasi ke suatu set data data yang terstruktur. Mereka mengenkapsulasi data tersebut dan membuat suatu mekanisme untuk mengakses isinya agar keamanan data terjaga. *Content Provider* merupakan antarmuka standar yang dapat digunakan oleh pengembang suatu aplikasi untuk melakukan proses mengakses data yang terdapat pada proses lain di luar aplikasi tempat proses itu berjalan.

Ketika pengembang ingin mengakses data pada *content provider*, pengembang dapat menggunakan objek *content resolver* pada aplikasinya untuk melakukan pengiriman data dengan *content provider*. Nantinya, aplikasi akan dianggap sebagai *client* dan *content provider* akan bertindak seperti *server*. *Content provider* ini nanti akan menerima permintaan dari *client*, lalu akan melakukan proses sesuai yang diminta, dan akan mengembalikan hasilnya kepada *client*. Tentunya untuk membaca isi dari *content provider*, pengembang perlu mencantumkan izin apa saja yang dibutuhkan oleh aplikasinya ke dalam *file manifest* milik android.

⁶ <http://json.org/json-id.html>

⁷ <http://developer.android.com/guide/topics/providers/content-providers.html>

2.8 *Location Based Services*

Location Based Services atau layanan berbasis lokasi adalah sebuah layanan yang dapat diakses oleh perangkat bergerak melalui untuk mendapatkan posisi perangkat secara geografis (Singhal & Shukla, 2012). Layanan berbasis lokasi memanfaatkan lokasi geografis perangkat untuk menyediakan informasi-informasi yang dibutuhkan. Ada beberapa teknologi yang digunakan layanan ini untuk mendapatkan lokasi geografis sebuah perangkat bergerak, salah satunya *Global Positioning System* atau GPS.

GPS merupakan salah satu teknologi yang terkenal akurat untuk digunakan dalam proses pencarian informasi lokasi suatu perangkat. GPS memanfaatkan sinyal dari beberapa satelit untuk melakukan pendeteksian lokasi. Hal ini menyebabkan adanya kelemahan pada teknologi GPS dimana ketika suatu perangkat berada di dalam ruangan, teknologi ini tidak akan bekerja secara maksimal. Hal ini ditandai dengan lamanya suatu perangkat mendapatkan koordinat lintang dan bujurnya ketika perangkat tersebut menggunakan GPS di dalam ruangan.

Pada perangkat android yang memiliki fitur GPS, pengembang umumnya memiliki dua pilihan untuk mendapatkan lokasi perangkat. Yang pertama menggunakan GPS, kedua menggunakan jaringan internet. Seperti yang telah dijelaskan di awal, GPS menghasilkan koordinat lintang dan bujur yang lebih akurat daripada ketika menggunakan jaringan internet⁸. Namun melihat kelemahan GPS ketika digunakan dalam pendeteksian lokasi di dalam ruangan, pencarian lokasi menggunakan jaringan internet lebih sesuai dengan kebutuhan aplikasi tugas akhir ini.

2.9 *Webservice*

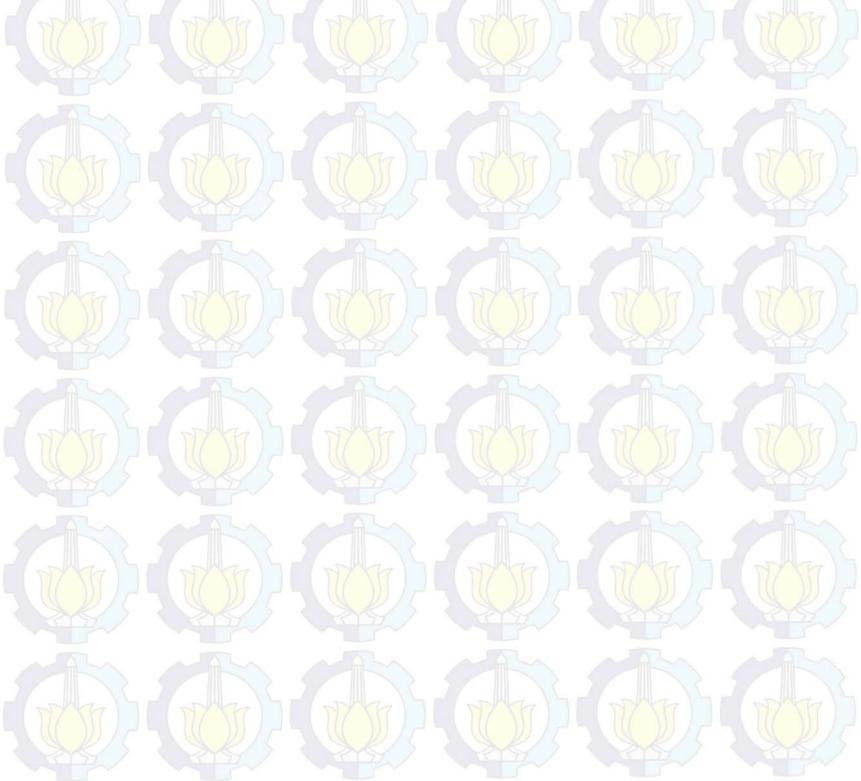
*Webservice*⁹ adalah mekanisme paling standar yang menghubungkan sebuah perangkat lunak dengan perangkat lunak

⁸ <http://stackoverflow.com/questions/1513485/how-do-i-get-the-current-gps-location-programmatically-in-android>

⁹ http://en.wikipedia.org/wiki/Web_service

lain melalui jaringan internet. *Webservice* menggunakan XML untuk melakukan proses *encoding* data melalui protokol HTTP sehingga hampir seluruh *platform* dapat menerima data yang disediakan oleh *webservice*. Selain HTTP dan XML, *webservice* biasanya dibangun dengan dua *platform* tambahan yaitu WSDL dan SOAP.

Dalam tugas akhir ini, *webservice* digunakan untuk menerima data dari *client* yang dikirimkan oleh aplikasi pada perangkat android. Data tersebut diterima dalam bentuk JSON, lalu diolah menjadi tampilan berbentuk tabel. Hasil data olahan ini nantinya dikirimkan ke alamat *email* orang tua.



BAB III

ANALISIS DAN PERANCANGAN

Pada bab ini dibahas analisis kebutuhan, proses, dan desain dari perangkat lunak yang dibangun dalam tugas akhir ini. Bagian awal bab akan dibahas tentang analisis permasalahan dan kebutuhan calon pengguna. Berikutnya dibahas fungsional sistem yang berdasarkan hasil analisis kebutuhan. Bagian berikutnya akan dibahas rancangan perangkat lunak yang ditujukan untuk memberikan gambaran tentang perangkat lunak yang dibuat.

3.1. Analisis

Pada sub bab berikut akan dijelaskan analisis pembuatan aplikasi pemantau pemakaian perangkat bergerak oleh anak. Analisis yang dilakukan meliputi analisis permasalahan, deskripsi umum sistem, dan kebutuhan fungsional sistem.

3.1.1. Analisis Permasalahan

Dewasa ini, semakin banyak orang tua yang memberikan perangkat bergerak kepada anaknya baik berupa ponsel pintar maupun tablet, tidak peduli berapa usia anak. Untuk anak usia balita yang masih belum mengenyam jenjang pendidikan, hal ini dilakukan untuk penenang anak atau semacam *digital babysitter* (Anna, 2011). Sedangkan untuk anak yang telah menginjak jenjang pendidikan formal, hal tersebut dilakukan untuk kemudahan berkomunikasi dengan orang tua maupun teman anak sampai penunjang kegiatan akademik anak.

Namun, dengan semakin banyaknya hiburan serta akses internet nyaris tanpa batas yang ditawarkan oleh perangkat-perangkat bergerak, tidak dapat dipungkiri bahwa perangkat bergerak memiliki ancaman-ancaman kepada anak seperti ketergantungan terhadap *video game*, ketergantungan terhadap media sosial, bahkan memungkinkan anak terlibat dengan tindakan kriminal seperti peredaran narkoba, konsumsi minuman keras, seks bebas, maupun *bullying* yang semestinya bisa dicegah

jika orang tua bisa memantau aktivitas sosial anak dan aktivitas anak ketika menggunakan perangkat bergerak.

3.1.2. Deskripsi Umum Perangkat Lunak

Aplikasi yang dibangun pada tugas akhir ini akan menghasilkan luaran berupa data. Data ini nantinya dapat diolah menjadi hal-hal yang perlu dievaluasi terkait kegiatan anak selama menggunakan perangkat bergerak. Aplikasi ini memiliki kemampuan untuk mencatat aktivitas anak dengan perangkat bergerak miliknya lalu mengirimkan *record* tersebut ke alamat *email* orang tua.

Dalam implementasinya, perangkat lunak ini terdiri dari dua aplikasi. Aplikasi pertama adalah aplikasi yang berjalan di perangkat bergerak android yang nantinya akan dijalankan di perangkat bergerak milik anak. Sedangkan aplikasi kedua merupakan aplikasi *webservice* yang akan menerima data yang dikirim oleh aplikasi pertama, lalu mengolahnya untuk diteruskan ke alamat *email* orang tua. Cara kerja aplikasi pertama ini adalah orang tua cukup melakukan instalasi pada perangkat milik anak, lalu melakukan registrasi menggunakan alamat *email* yang nantinya dijadikan alamat tujuan pengiriman *record* dan menetapkan *password* untuk mengakses aplikasi ini pada perangkat milik anak. Kemudian, orang tua mengatur jam berapa aplikasi akan mengirimkan *email* yang berisi *record* serta statistik pemakaian perangkat oleh anak.

Setiap 24 jam sekali sesuai waktu yang diatur oleh orang tua, aplikasi pertama mengirimkan data-data berupa JSON ke aplikasi kedua. Ketika aplikasi kedua menerima data tersebut, aplikasi kedua akan mengirimkan data-data di JSON tadi ke alamat *email* orang tua. Kebutuhan umum perangkat lunak aplikasi perangkat bergerak apabila diringkas adalah sebagai berikut.

1. Dapat mencatat lokasi perangkat.
2. Dapat mencatat waktu panggilan masuk beserta nomor telepon yang menghubungi.
3. Dapat mencatat waktu panggilan keluar beserta nomor telepon yang dihubungi.

4. Dapat mencatat waktu pesan singkat masuk beserta nomor telepon pengirim dan isi pesan.
5. Dapat mencatat waktu pesan singkat keluar beserta nomor telepon penerima dan isi pesan.
6. Dapat mencatat nama aplikasi yang dibuka beserta waktu saat aplikasi itu dibuka.
7. Dapat mencatat situs *web* yang dikunjungi oleh *browser* perangkat.
8. Dapat menampilkan statistik.
9. Dapat mengubah alamat *email* tujuan penerima *record*.
10. Dapat mengubah *password*.
11. Dapat mengubah waktu *record* dikirimkan ke *webservice*.
12. Dapat mengirimkan *record* ke *webservice*.

Sedangkan aplikasi berbasis *web* sifatnya hanya untuk meneruskan data yang dikirimkan melalui JSON ke alamat *email* tujuan dalam bentuk HTML sehingga dapat dilihat oleh orang tua. Kebutuhan umum perangkat lunak aplikasi *webservice* apabila diringkas adalah sebagai berikut.

1. Dapat menerima data yang dikirimkan oleh perangkat bergerak.
2. Dapat mengirimkan *email* yang berisi *record* perangkat bergerak.

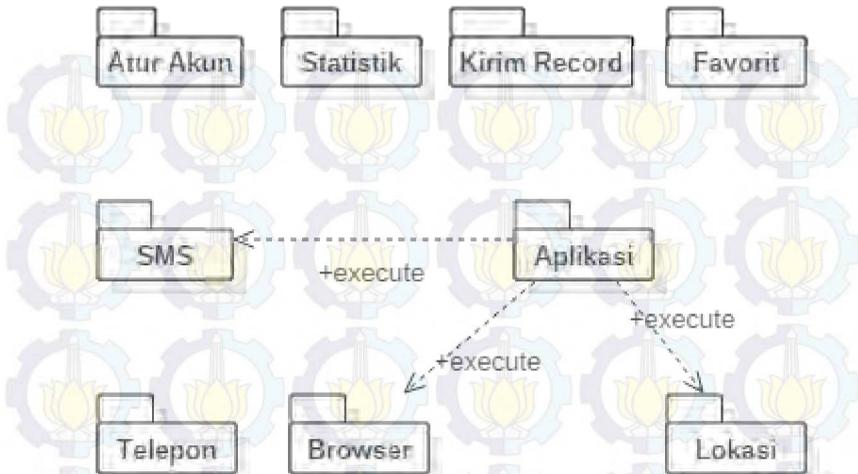
3.1.3. Kebutuhan Fungsional Aplikasi Android

Aplikasi pada perangkat bergerak digunakan untuk menangani proses penyimpanan dan pengolahan data pemakaian perangkat bergerak. Berikut daftar kebutuhan fungsional dari aplikasi pada perangkat bergerak.

1. Menampilkan lokasi perangkat.
2. Menampilkan waktu panggilan masuk.
3. Menampilkan nomor telepon yang menghubungi.
4. Menampilkan waktu panggilan keluar.
5. Menampilkan nomor telepon yang dihubungi.

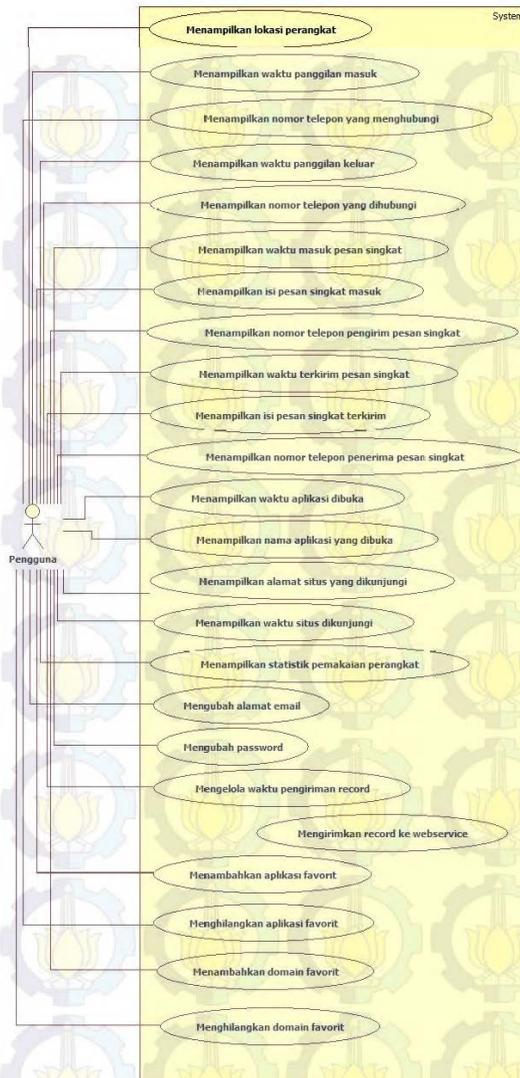
6. Menampilkan waktu masuk pesan singkat.
7. Menampilkan isi pesan singkat masuk.
8. Menampilkan nomor telepon pengirim pesan singkat.
9. Menampilkan waktu terkirim pesan singkat.
10. Menampilkan isi pesan singkat terkirim.
11. Menampilkan nomor telepon penerima pesan singkat.
12. Menampilkan waktu aplikasi dibuka.
13. Menampilkan nama aplikasi yang dibuka.
14. Menampilkan alamat situs yang dikunjungi.
15. Menampilkan waktu situs dikunjungi.
16. Menampilkan statistik pemakaian perangkat.
17. Mengubah alamat *email*.
18. Mengubah *password*.
19. Mengelola waktu pengiriman *record* ke alamat *email* orang tua.
20. Mengirimkan *record* ke *webservice*.
21. Menambahkan aplikasi favorit.
22. Menghilangkan aplikasi favorit.
23. Menambahkan *domain* situs favorit.
24. Menghilangkan *domain* situs favorit.

Dari kebutuhan-kebutuhan fungsional di atas, modul-modul utama yang akan menjadi penyusun aplikasi pada perangkat bergerak adalah sebagai berikut.



Gambar 3. 1 Diagram Modul Aplikasi Perangkat Bergerak

Sementara pada aplikasi *webservice*, hanya terdapat satu modul yaitu modul pengiriman data JSON ke alamat *email*. Sementara penjelasan modul-modul di atas digambarkan lebih spesifik dalam bentuk diagram kasus penggunaan. Terdapat dua diagram kasus penggunaan yang ditampilkan pada sub bab kali ini. Pertama adalah diagram kasus penggunaan aplikasi pada perangkat bergerak. Kedua adalah diagram kasus penggunaan aplikasi pada *webservice*. Diagram kasus penggunaan aplikasi perangkat bergerak ditampilkan sebagai berikut.



Gambar 3. 2 Diagram Kasus Penggunaan Aplikasi Perangkat Bergerak

Hak akses terhadap aplikasi android hanya diberikan kepada orang tua. Penjelasan lengkap mengenai kasus penggunaan pada aplikasi pemantauan perangkat bergerak anak berbasis android ada pada tabel berikut.

Tabel 3. 1 Daftar Kasus Penggunaan Aplikasi Perangkat Bergerak

No	Kode	Nama	Keterangan
1	UC-A01	Menampilkan lokasi perangkat	Orang tua dapat mengetahui lokasi-lokasi perangkat dalam bentuk koordinat lintang bujur.
2	UC-A02	Menampilkan waktu panggilan masuk	Pengguna dapat mengetahui waktu panggilan yang pernah masuk ke perangkat.
3	UC-A03	Menampilkan nomor telepon yang menghubungi	Pengguna dapat mengetahui nomor telepon berapa saja yang menghubungi perangkat milik anak.
4	UC-A04	Menampilkan waktu panggilan keluar	Pengguna dapat mengetahui waktu setiap panggilan keluar yang dilakukan oleh perangkat bergerak.
5	UC-A05	Menampilkan nomor telepon yang dihubungi	Pengguna dapat mengetahui nomor telepon berapa saja yang dihubungi oleh perangkat milik anak.
6	UC-A06	Menampilkan waktu masuk	Pengguna dapat mengetahui waktu setiap

		pesan singkat	pesan singkat yang masuk ke perangkat.
7	UC-A07	Menampilkan isi pesan singkat masuk	Pengguna dapat mengetahui isi setiap pesan singkat yang masuk ke perangkat.
8	UC-A08	Menampilkan nomor telepon pengirim pesan singkat	Pengguna dapat mengetahui nomor telepon pengirim setiap pesan singkat yang masuk ke perangkat.
9	UC-A09	Menampilkan waktu terkirim pesan singkat	Pengguna dapat mengetahui waktu setiap pesan singkat yang dikirim oleh perangkat.
10	UC-A10	Menampilkan isi pesan singkat terkirim	Pengguna dapat mengetahui isi setiap pesan singkat yang dikirim oleh perangkat.
11	UC-A11	Menampilkan nomor telepon penerima pesan singkat	Pengguna dapat mengetahui nomor telepon tujuan setiap pesan singkat yang dikirim oleh perangkat
12	UC-A12	Menampilkan waktu aplikasi dibuka	Pengguna dapat mengetahui waktu setiap aplikasi yang dibuka di perangkat.
13	UC-A13	Menampilkan nama aplikasi	Pengguna dapat mengetahui nama setiap aplikasi yang dibuka di

		yang dibuka	perangkat.
14	UC-A14	Menampilkan alamat situs yang dikunjungi	Pengguna dapat mengetahui alamat situs yang dikunjungi oleh perangkat.
15	UC-A15	Menampilkan waktu situs dikunjungi	Pengguna dapat mengetahui waktu dari alamat situs yang dikunjungi oleh perangkat.
16	UC-A16	Menampilkan statistik pemakaian perangkat	Pengguna dapat mengetahui statistik pemakaian perangkat. Statistik yang dimaksud antara lain: <ul style="list-style-type: none"> • 5 aplikasi yang paling sering dibuka, • 5 nomor telepon yang paling sering dihubungi, • 5 nomor telepon yang paling sering menghubungi perangkat • 5 nomor telepon yang paling sering menerima pesan singkat dari perangkat, dan • 5 nomor telepon yang paling sering mengirim pesan singkat ke perangkat.
17	UC-A17	Mengubah alamat <i>email</i>	Pengguna dapat mengubah alamat <i>email</i> yang dijadikan sebagai sarana

			untuk <i>login</i> dan tujuan pengiriman <i>record</i> pemakaian perangkat.
18	UC-A18	Mengubah <i>password</i>	Pengguna dapat mengubah <i>password</i> yang dijadikan sebagai sarana untuk <i>login</i> .
19	UC-A19	Mengelola waktu pengiriman <i>record</i> ke alamat <i>email</i> orang tua	Pengguna dapat mengatur waktu kapan aplikasi mengirimkan <i>record</i> ke alamat <i>email</i> yang diinginkan.
20	UC-A20	Mengirimkan <i>record</i> ke <i>webservice</i>	Perangkat mampu mengirimkan <i>record</i> ke <i>webservice</i> dalam bentuk JSON.
21	UC-A21	Menambahkan aplikasi favorit	Pengguna dapat menambahkan aplikasi ke dalam daftar aplikasi favorit untuk ditampilkan dengan cara berbeda dari aplikasi lainnya. Hal ini dilakukan bertujuan untuk kemudahan dalam melihat daftar aplikasi yang dibuka pada perangkat.
22	UC-A22	Menghilangkan aplikasi favorit	Pengguna dapat menghapus salah satu aplikasi dari dalam daftar aplikasi favorit.
23	UC-A23	Menambahkan <i>domain</i> situs	Pengguna dapat menambahkan <i>domain</i>

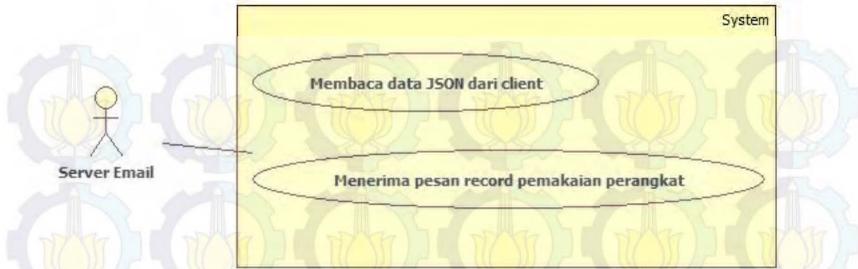
		favorit	situs favorit sehingga setiap situs dengan <i>domain</i> favorit tersebut akan ditampilkan dengan cara berbeda dari situs lainnya. Hal ini juga dilakukan bertujuan untuk kemudahan dalam melihat daftar situs yang dikunjungi perangkat.
24	UC-A24	Menghilangkan <i>domain</i> situs favorit	Pengguna dapat menghapus salah satu <i>domain</i> situs dari dalam daftar <i>domain</i> situs favorit.

3.1.4. Kebutuhan Fungsional Aplikasi *Webservice*

Aplikasi *webservice* ini merupakan aplikasi yang digunakan untuk menerima JSON yang dikirimkan oleh aplikasi pada perangkat bergerak, lalu mengolahnya agar dapat dikirim melalui *email* dan mengirimkannya. Berikut daftar kebutuhan fungsional dari aplikasi *webservice* ini.

1. Membaca data JSON yang dikirim oleh *client*.
2. Mengirim *record* ke alamat *email* yang ditentukan.

Diagram kasus penggunaan aplikasi ini digambarkan sebagai berikut.



Gambar 3. 3 Diagram Kasus Penggunaan Aplikasi Webservice

Oleh karena aplikasi ini sifatnya adalah sebagai *webservice* semata, maka aplikasi ini tidak akan memiliki halaman antarmuka. Berikut penjelasan dari kebutuhan fungsional aplikasi ini.

Tabel 3. 2 Daftar Kasus Penggunaan Aplikasi *Webservice*

No	Kode	Nama	Keterangan
1	UC-W01	Membaca data JSON yang dikirim oleh <i>client</i>	Aplikasi dapat membaca JSON yang dikirim oleh <i>client</i> lalu melakukan <i>decoding</i> agar nilai-nilai dari JSON tersebut dapat diterjemahkan ke dalam bentuk variabel PHP.
2	UC-W02	Mengirim <i>record</i> ke alamat <i>email</i> yang ditentukan	Aplikasi dapat mengirim sebuah halaman HTML yang berisi <i>record</i> pemakaian perangkat bergerak ke alamat <i>email</i> yang telah ditentukan.

3.2. Perancangan

Sub bab berikut membahas tentang perancangan dari aplikasi pemantau penggunaan perangkat bergerak anak. Sub bab berikut membahas rancangan antarmuka, rancangan proses dan rancangan basis data. Pembahasan lebih detail akan dibahas berikut ini.

3.2.1. Perancangan Umum Arsitektur Sistem

Sistem ini dirancang dengan menggunakan *server* dan *client* sebagai pengumpul dan pengolah data. *Server* berupa *webservice* sedangkan *client* berupa perangkat bergerak android. *Client* bertugas mengumpulkan data berupa catatan penggunaan perangkat bergerak tersebut kemudian ketika waktu pengiriman tiba, *client* akan mengirimkan data tersebut ke *server*. *Server* bertugas menerima data tersebut lalu mengirimkan seluruh data tersebut ke alamat *email* yang dituju. Seluruh data pada *client* disimpan pada basis data lokal yang ada pada masing-masing perangkat. Berikut gambar perancangan arsitektur sistem secara umum.



Gambar 3. 4 Arsitektur Umum Sistem

3.2.2. Perancangan Basis Data

Pada sub bab ini akan dijelaskan bagaimana rancangan basis data yang digunakan pada aplikasi pemantau penggunaan perangkat bergerak. Basis data pada sistem yang dibangun pada tugas akhir ini menggunakan sistem manajemen basis data relasional SQLite. SQLite digunakan untuk penyimpanan data pengguna, pesan singkat, telepon, lokasi, aplikasi, dan situs. Oleh karena tabel-tabel yang dirancang tidak berkaitan dan berelasi satu sama lain, maka rancangan basis data akan langsung dijabarkan sebagai berikut.

3.2.1.1. Rancangan Tabel Pengguna (*user*)

Tabel pengguna digunakan untuk menyimpan data pengguna yang digunakan sebagai akses untuk masuk dalam aplikasi pada perangkat android.

Tabel 3. 3 Rincian Tabel Pengguna

Nama Kolom	Type Data	Keterangan
_id	Integer	<i>Primary Key</i> dari

		tabel
email	Text	Alamat pengguna <i>email</i>
password	Text	Kata kunci pengguna

3.2.1.2. Rancangan Tabel SMS Masuk (*incomingSMS*)

Tabel ini digunakan untuk menyimpan data pesan singkat yang masuk ke dalam perangkat.

Tabel 3. 4 Rincian Tabel SMS Masuk

Nama Kolom	Tipe Data	Keterangan
_id	Integer	<i>Primary Key</i> dari tabel
number	Text	Nomor telepon pengirim pesan singkat
time	Text	Waktu pesan singkat masuk
content	Text	Isi pesan singkat
flag	Text	Jenis tabel

3.2.1.3. Rancangan Tabel SMS Keluar (*outgoingSMS*)

Tabel ini digunakan untuk menyimpan data pesan singkat yang dikirim oleh perangkat.

Tabel 3. 5 Rincian Tabel SMS Keluar

Nama Kolom	Tipe Data	Keterangan
------------	-----------	------------

_id	Integer	<i>Primary Key</i> dari tabel
number	Text	Nomor telepon penerima pesan singkat
time	Text	Waktu pesan singkat dikirim
content	Text	Isi pesan singkat
flag	Text	Jenis tabel

3.2.1.4. Rancangan Tabel Telepon Masuk (*incomingCall*)

Tabel ini digunakan untuk menyimpan data panggilan telepon yang masuk ke perangkat.

Tabel 3. 6 Rincian Tabel Telepon Masuk

Nama Kolom	Tipe Data	Keterangan
_id	Integer	<i>Primary Key</i> dari tabel
number	Text	Nomor telepon yang menghubungi perangkat
details	Text	Detail panggilan telepon berupa waktu panggilan masuk sampai waktu panggilan berakhir

3.2.1.5. Rancangan Tabel Telepon Keluar (*outgoingCall*)

Tabel ini digunakan untuk menyimpan data panggilan telepon dari perangkat.

Tabel 3. 7 Rincian Tabel Telepon Keluar

Nama Kolom	Tipe Data	Keterangan
_id	Integer	<i>Primary Key</i> dari tabel
number	Text	Nomor telepon yang dihubungi perangkat
details	Text	Detail panggilan berupa waktu panggilan keluar dimulai sampai waktu panggilan berakhir

3.2.1.6. Rancangan Tabel Aplikasi (*apps*)

Tabel aplikasi digunakan untuk menyimpan data aplikasi yang dibuka oleh perangkat.

Tabel 3. 8 Rincian Tabel Aplikasi

Nama Kolom	Tipe Data	Keterangan
_id	Integer	<i>Primary Key</i> dari tabel
package	Text	Nama aplikasi yang dibuka
time	Text	Waktu aplikasi dibuka

banned	Text	Keterangan apakah aplikasi masuk daftar favorit
duration	Integer	Durasi aplikasi dibuka dalam satuan detik

3.2.1.7. Rancangan Tabel Lokasi (*location*)

Tabel lokasi digunakan untuk menyimpan data lokasi perangkat selama digunakan.

Tabel 3. 9 Rincian Tabel Lokasi

Nama Kolom	Tipe Data	Keterangan
_id	Integer	<i>Primary Key</i> dari tabel
coordinate	Text	Koordinat lokasi berupa lintang dan bujur
time	Text	Waktu pencatatan lokasi
flag	Text	Jenis tabel

3.2.1.8. Rancangan Tabel Situs (*browser*)

Tabel situs digunakan untuk menyimpan situs yang dikunjungi oleh perangkat.

Tabel 3. 10 Rincian Tabel Situs

Nama Kolom	Tipe Data	Keterangan
------------	-----------	------------

_id	Integer	<i>Primary Key</i> dari tabel
time	Text	Waktu dikunjungi situs
url	Text	Alamat situs
title	Text	Judul situs
flag	Text	Jenis tabel

3.2.1.9. Rancangan Tabel Aplikasi Favorit (*bannedApps*)

Tabel ini digunakan untuk mencatat aplikasi-aplikasi apa saja yang dimasukkan ke daftar aplikasi favorit oleh orang tua.

Tabel 3. 11 Rincian Tabel Aplikasi Favorit

Nama Kolom	Tipe Data	Keterangan
_id	Integer	<i>Primary Key</i> dari tabel
package	Text	Nama aplikasi

3.2.1.10. Rancangan Tabel Situs Favorit (*bannedLink*)

Tabel ini digunakan untuk mencatat *domain* situs apa saja yang dimasukkan ke daftar *domain* favorit oleh orang tua.

Tabel 3. 12 Rincian Tabel Situs Favorit

Nama Kolom	Tipe Data	Keterangan
_id	Integer	<i>Primary Key</i> dari tabel

url	Text	Alamat <i>domain</i> situs
-----	------	----------------------------

3.2.1.11. Rancangan Tabel Alarm (*alarm*)

Tabel ini digunakan untuk menyimpan waktu pengiriman *record* ke *webservice*.

Tabel 3. 13 Rincian Tabel Alarm

Nama Kolom	Tipe Data	Keterangan
<u>_id</u>	Integer	<i>Primary Key</i> dari tabel
hour	Text	Jam pengiriman <i>record</i> ke <i>webservice</i>
minute	Text	Menit pengiriman <i>record</i> ke <i>webservice</i>

3.2.1.12. Rancangan Tabel SMS Sementara (*temporarySMS*)

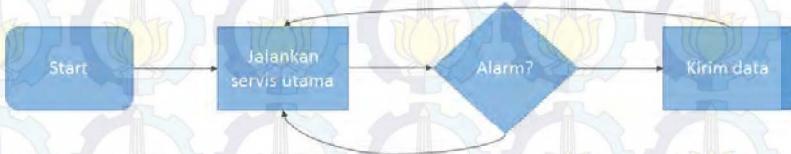
Tabel ini digunakan untuk menyimpan pesan terkirim terakhir dari perangkat. Pesan singkat terkirim terakhir perangkat tidak disimpan pada variabel khusus untuk menghindari terhapusnya isi variabel tersebut oleh sistem.

Tabel 3. 14 Rincian Tabel SMS Sementara

Nama Kolom	Tipe Data	Keterangan
<u>_id</u>	Integer	<i>Primary Key</i> dari tabel
sms	Text	Isi pesan beserta nomor tujuan penerima pesan

3.2.3. Perancangan Algoritma

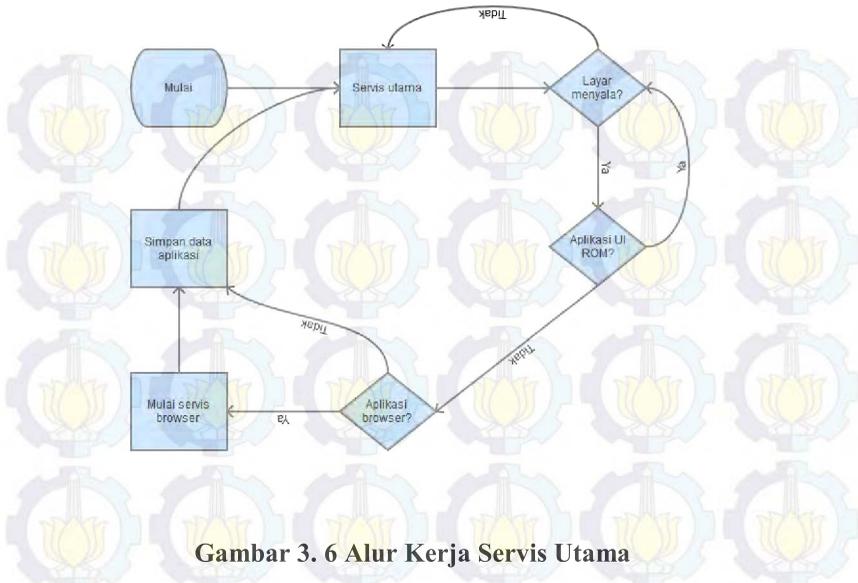
Perancangan algoritma merupakan tahap untuk membentuk alur proses yang berjalan pada sistem. Berikut perancangan alur kerja aplikasi pada perangkat bergerak secara garis besar.



Gambar 3. 5 Alur Kerja Aplikasi Perangkat Bergerak

3.2.3.1 Servis Utama

Servis utama ini merupakan inti dari proses pencatatan pemakaian perangkat bergerak. Dari penjelasan data-data yang akan dicatat oleh aplikasi perangkat bergerak ini, ada beberapa jenis data yang bisa didapatkan melalui *broadcast* yang dikirimkan oleh sistem operasi namun ada juga data yang harus didapatkan secara manual dengan membaca isi *content provider*. Data-data yang harus dibaca secara manual melalui isi *content provider* inilah yang harus dikerjakan oleh *service* khusus yang harus dibuat. Data-data tersebut antara lain data aplikasi yang sedang dibuka, data pesan yang dikirimkan, data situs yang dikunjungi, dan data lokasi perangkat berada. Secara garis besar berikut diagram alur servis utama yang berjalan di latar belakang perangkat android.



Gambar 3. 6 Alur Kerja Servis Utama

Servis *SMS* dan servis *browser* sendiri berisi sebuah *content resolver* sesuai kebutuhan data yang ingin didapatkan. Cara kerjanya sendiri dengan menyimpan setiap adanya perubahan data pada *content provider* yang diakses tersebut. Perubahan data tersebut menandakan adanya pesan keluar yang baru saja terkirim atau situs yang baru saja dikunjungi oleh perangkat. Servis utama ini tidak boleh berhenti menyala karena kerja servis-servis yang lain bergantung *trigger* yang dibuat oleh servis utama. Maka dari itu, untuk mencegah servis ini dimatikan oleh sistem, servis ini harus dijadikan sebagai *foreground service*¹⁰ agar sistem operasi memprioritaskan servis ini sebagai servis terakhir yang akan dimatikan ketika sistem operasi kehabisan memori.

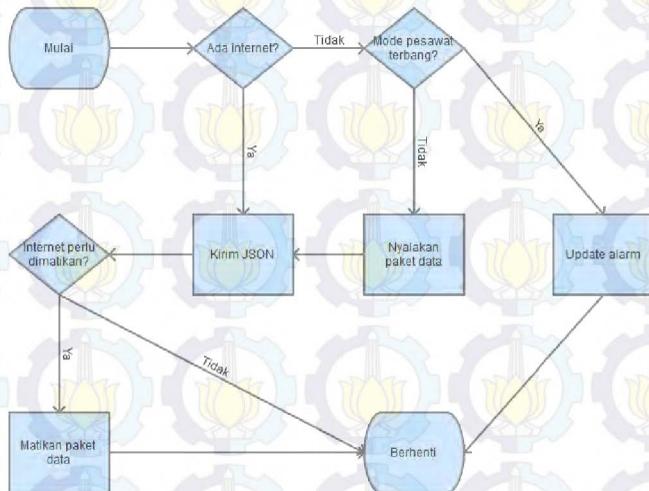
Pada perkembangannya, keterbatasan kapasitas memori yang dimiliki oleh perangkat android menjadi kendala untuk servis *browser* secara terus menerus. Oleh sebab itu, ketika aplikasi yang dibuka tidak berhubungan dengan *browser* maka servis

¹⁰ <http://developer.android.com/guide/components/services.html#Foreground>

utama akan menghentikan servis *browser* jika servis tersebut sedang berjalan. Hal ini ditujukan untuk mengantisipasi terjadinya *crash* karena aplikasi menjadi rakus memori.

3.2.3.2 Pengiriman Data

Pada proses pengiriman data, hal yang perlu dipertimbangkan ada dua hal. Pertama, untuk menyiasati ketika anak mematikan akses internet perangkat bergerak saat aplikasi akan mengirimkan JSON ke *webservice*. Kedua, jika perangkat bergerak sedang berada dalam penerbangan pesawat dan menyalakan fitur mode pesawat, aplikasi seharusnya menunda pengiriman JSON ke *webservice*. Oleh sebab itu, aplikasi dirancang agar mengetahui situasi perangkat bergerak tempatnya bernaung. Berikut diagram alur pengiriman data ke *webservice* dengan mempertimbangkan kendala di atas.



Gambar 3. 7 Alur Kerja Proses Pengiriman Data

3.2.4. Perancangan Antarmuka Sistem

Pada subbab ini akan dibahas secara mendetail rancangan antarmuka hanya pada aplikasi perangkat bergerak.

3.2.4.1. Antarmuka Halaman Daftar

Halaman ini berfungsi untuk menampung pengguna yang belum pernah terdaftar pada aplikasi ini. Berikut tampilan antarmuka halaman daftar.



Gambar 3. 8 Rancangan Antarmuka Halaman Daftar

Komponen-komponen penyusun halaman di atas terdapat pada tabel berikut.

Tabel 3. 15 Keterangan Halaman Antarmuka Daftar

Nomor	Nama Komponen	Keterangan
1	ImageView	Digunakan untuk menampilkan ikon aplikasi.
2	TextView1	Digunakan untuk menampilkan

		nama aplikasi.
3	EditText1	Digunakan sebagai tempat pengisian alamat <i>email</i> pengguna untuk dapat masuk ke aplikasi.
4	EditText2	Digunakan sebagai tempat pengisian <i>password</i> pengguna.
5	EditText3	Digunakan sebagai tempat pengisian ulang <i>password</i> pengguna.
6	Button	Digunakan untuk melakukan proses registrasi ke basis data aplikasi.
7	TextView2	Digunakan untuk menampilkan tulisan pemberitahuan kepada pengguna bahwa masukan alamat <i>email</i> harus benar.

3.2.4.2. Antarmuka Halaman Masuk

Halaman masuk ini berfungsi menangani pengguna yang ingin mengakses aplikasi ketika aplikasi ini memiliki data akun seorang pengguna di dalam basis. Berikut tampilan antarmuka halaman masuk.



Gambar 3. 9 Rancangan Antarmuka Halaman Login

Komponen-komponen penyusun halaman di atas adalah sebagai berikut.

Tabel 3. 16 Keterangan Halaman Antarmuka Login

Nomor	Nama Komponen	Keterangan
1	ImageView	Digunakan untuk menampilkan ikon aplikasi.
2	TextView	Digunakan untuk menampilkan nama aplikasi.
3	EditText1	Digunakan sebagai tempat pengisian alamat <i>email</i> pengguna untuk dapat masuk ke aplikasi.
4	EditText2	Digunakan sebagai tempat pengisian <i>password</i> pengguna.
5	Button	Digunakan untuk melakukan proses pengecekan <i>username</i> dan

		<i>password</i> yang dimasukkan pengguna ke basis data aplikasi.
--	--	--

3.2.4.3. Antarmuka Halaman *Overview*

Halaman *overview* merupakan halaman yang menampilkan seluruh hasil pencatatan yang dilakukan terhadap aktivitas yang dilakukan dengan perangkat. Berikut tampilan antarmuka halaman *overview*.



Gambar 3. 10 Rancangan Antarmuka Halaman Overview

Halaman ini terdiri dari sebuah *spinner* dan sebuah *listview*. Berikut komponen-komponen penyusun halaman *overview*.

Tabel 3. 17 Keterangan Halaman Antarmuka Overview

Nomor	Nama Komponen	Keterangan
1	Spinner	Digunakan untuk menampilkan

		pilihan data apa saja yang ingin ditampilkan pada <i>listview</i> .
2	ListView	Digunakan untuk menampilkan hasil <i>record</i> yang telah tersimpan pada basis data aplikasi. Pada <i>listview</i> ini berisi sebuah <i>textview</i> dimana <i>textview</i> ini nantinya dijadikan sebagai media untuk menampilkan hasil <i>record</i> .

3.2.4.4. Antarmuka Halaman *Edit Akun*

Halaman *edit* akun merupakan halaman yang digunakan pengguna untuk mengelola akun yang digunakan untuk masuk ke dalam aplikasi. Pada halaman ini terdiri dari kolom isian alamat *email* dan *password* baru yang dikehendaki oleh pengguna jika ingin melakukan perubahan. Berikut tampilan halaman *edit* akun.



Gambar 3. 11 Rancangan Antarmuka Halaman Edit Akun

Berikut komponen-komponen penyusun halaman *edit* akun.

Tabel 3. 18 Keterangan Halaman Antarmuka Edit Akun

Nomor	Nama Komponen	Keterangan
1	EditText1	Digunakan sebagai tempat pengisian alamat <i>email</i> baru pengguna untuk dapat masuk ke aplikasi.
2	EditText2	Digunakan sebagai tempat pengisian <i>password</i> baru pengguna.
3	EditText3	Digunakan sebagai tempat konfirmasi pengisian <i>password</i> baru.
4	Button	Digunakan untuk melakukan

		proses pengubahan alamat <i>email</i> dan <i>password</i> baru pada basis data.
5	TextView	Digunakan untuk menampilkan keterangan pada halaman.

3.2.4.5. Antarmuka Halaman Statistik

Halaman statistik merupakan halaman yang digunakan untuk menampilkan statistik pemakaian perangkat bergerak dalam bentuk diagram. Diagram yang ditampilkan merupakan diagram pai. Berikut rancangan antarmuka halaman statistik.



Gambar 3. 12 Rancangan Antarmuka Halaman Statistik

Komponen-komponen penyusun halaman statistik ada dalam tabel berikut.

Tabel 3. 19 Keterangan Halaman Antarmuka Statistik

Nomor	Nama Komponen	Keterangan
1	Spinner	Digunakan untuk menampilkan pilihan data statistik apa saja yang ingin ditunjukkan pada <i>piechart</i> .
2	PieChart	Digunakan untuk menampilkan statistik pemakaian perangkat bergerak. Bagian ini terdiri dari sebuah gambar diagram dan sebuah legenda yang menjelaskan maksud dari warna-warna yang ada pada diagram pai tersebut.

3.2.4.6. Antarmuka Halaman Atur Jadwal

Halaman atur jadwal merupakan halaman yang digunakan untuk menangani kebutuhan pengguna dalam mengatur jadwal kapan *email* yang berisi *record* pemakaian perangkat bergerak dikirimkan ke alamat *email* orang tua. Berikut rancangan antarmuka halaman atur jadwal.



Gambar 3. 13 Rancangan Antarmuka Halaman Atur Jadwal

Komponen-komponen penyusun halaman ini antara lain sebagai berikut.

Tabel 3. 20 Keterangan Halaman Antarmuka Atur Jadwal

Nomor	Nama Komponen	Keterangan
1	TextView	Digunakan untuk menampilkan tulisan yang menerangkan tentang fungsi halaman.
2	DatePicker	Digunakan sebagai media pengguna untuk menentukan jam berapa <i>record</i> dikirimkan oleh aplikasi ke alamat <i>email</i> pengguna.
3	Button	Digunakan sebagai proses untuk menyimpan waktu pengiriman <i>record</i> .

3.2.4.7. Antarmuka Halaman Menambahkan Favorit

Halaman ini merupakan halaman yang digunakan untuk menangani kebutuhan pengguna dalam menandai aplikasi-aplikasi serta *domain-domain* situs mana saja yang perlu ditandai menjadi aplikasi dan *domain* favorit. Halaman ini nantinya menampilkan seluruh aplikasi dan situs yang telah diakses oleh anak sehingga orang tua tinggal memilih aplikasi dan situs dengan *domain* apa yang dianggap perlu diperhatikan. Berikut rancangan antarmuka halaman ini.



Gambar 3. 14 Rancangan Antarmuka Halaman Menambahkan Favorit

Komponen-komponen penyusun halaman ini antara lain sebagai berikut.

Tabel 3. 21 Keterangan Halaman Antarmuka Menambahkan Favorit

Nomor	Nama Komponen	Keterangan
1	Spinner	Digunakan untuk menampilkan pilihan jenis favorit yang ingin ditambahkan. Nantinya <i>spinner</i> ini jika ditekan akan memunculkan sebuah fragmen <i>popup</i> yang berisi pilihan aplikasi dan <i>domain</i> .
2	ListView	Digunakan sebagai media untuk menampilkan daftar calon aplikasi atau situs yang bisa ditambahkan ke dalam objek favorit oleh pengguna. <i>Listview</i> ini nantinya berisi gabungan <i>checkbox</i> dan <i>textview</i> .
3	Button	Digunakan sebagai proses untuk menyimpan daftar konten-konten yang ditandai sebagai objek favorit.

3.2.4.8. Antarmuka Halaman Menghapus Favorit

Halaman ini merupakan halaman untuk menangani kebutuhan pengguna dalam menghilangkan aplikasi atau *domain* dari daftar favorit yang dimiliki. Berikut tampilan rancangan antarmuka dari halaman menghapus favorit.



Gambar 3. 15 Rancangan Antarmuka Halaman Menghapus Favorit

Komponen-komponen penyusun antarmuka halaman ini antara lain sebagai berikut.

Tabel 3. 22 Keterangan Halaman Antarmuka Menghapus Favorit

Nomor	Nama Komponen	Keterangan
1	Spinner	Digunakan untuk menampilkan pilihan jenis favorit yang ingin dihapus dari daftar. Nantinya <i>spinner</i> ini jika ditekan akan memunculkan sebuah fragmen <i>popup</i> yang berisi pilihan aplikasi dan <i>domain</i> .
2	ListView	Digunakan sebagai media untuk menampilkan daftar calon aplikasi atau <i>domain</i> situs yang bisa dihapus dari dalam daftar objek

		favorit oleh pengguna. <i>Listview</i> ini nantinya berisi gabungan <i>checkbox</i> dan <i>textview</i> .
3	Button	Digunakan sebagai proses untuk menyimpan perubahan isi dari daftar konten yang ditandai sebagai objek favorit.

BAB IV IMPLEMENTASI

Pada bab ini dijelaskan mengenai implementasi dari perancangan perangkat lunak. Implementasi yang dijelaskan meliputi lingkungan pembangunan perangkat lunak, implementasi antarmuka pengguna, dan implementasi proses-proses yang terjadi pada perangkat lunak. Implementasi sistem mengacu pada perancangan yang ditulis pada Bab III.

4.1. Lingkungan Implementasi

Dalam merancang perangkat lunak ini digunakan beberapa perangkat pendukung. Perangkat keras yang digunakan dalam pengembangan sistem ini berupa komputer dengan spesifikasi AMD Phenom II X6 1090T 3,20 Ghz dengan memori 8GB. Sementara spesifikasi perangkat lunak yang digunakan untuk pengembangan sistem adalah sebagai berikut :

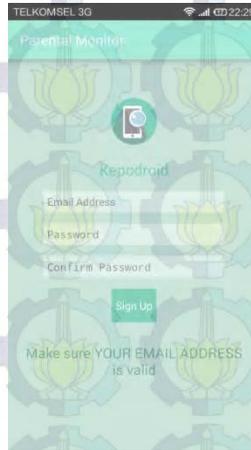
- Microsoft Windows 7 64bit sebagai sistem operasi.
- Android Studio 1.1.0 sebagai IDE untuk implementasi aplikasi.
- Android SDK versi 24.1.2 sebagai kerangka aplikasi perangkat bergerak.
- Apache versi 2.4.12 dan PHP versi 5.4.41 sebagai lingkungan web server aplikasi *webservice*.

4.2. Implementasi Antarmuka

Pada subbab ini akan dibahas implementasi antarmuka berdasarkan rancangan antarmuka yang telah dibahas pada bab sebelumnya. Antarmuka yang akan dibahas merupakan antarmuka pada aplikasi perangkat bergerak. Berikut rincian rancangan antarmuka pada aplikasi perangkat bergerak.

- a. **Halaman daftar** merupakan halaman yang akan terbuka pertama kali ketika pengguna baru saja melakukan instalasi

aplikasi ke perangkat bergerak. Halaman ini digunakan oleh pengguna untuk mendaftarkan alamat *email* beserta *password* yang diinginkan untuk digunakan sebagai alat masuk ke aplikasi. Berikut implementasi dari halaman daftar pada aplikasi perangkat bergerak.



Gambar 4. 1 Antarmuka Halaman Daftar

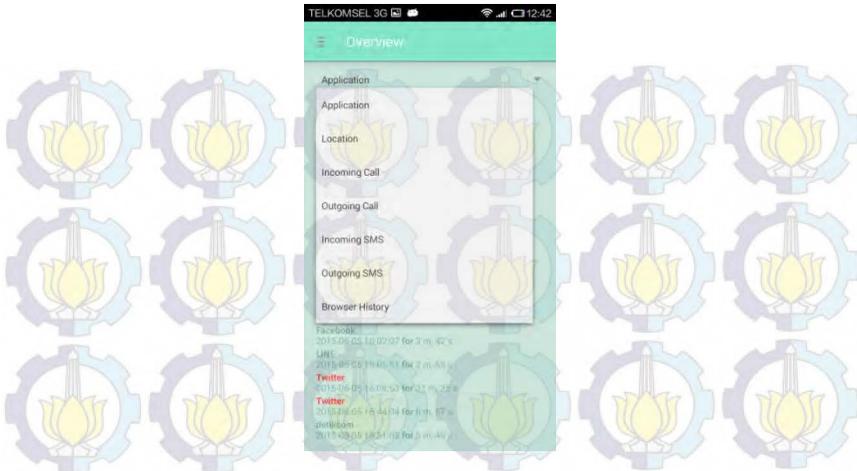
- b. **Halaman masuk** merupakan halaman yang akan terbuka pertama kali ketika pengguna membuka aplikasi, lalu aplikasi mendeteksi bahwa telah ada pengguna yang terdaftar sebelumnya. Berikut implementasi dari halaman masuk pada aplikasi perangkat bergerak.



Gambar 4.2 Antarmuka Halaman Login

Pada gambar di atas, pengguna diminta untuk memasukkan alamat *email* dan *password* untuk masuk ke aplikasi. Jika terjadi kesalahan, maka aplikasi akan menampilkan *toast* yang menginformasikan bahwa alamat *email* atau *password* yang dimasukkan salah.

- c. **Halaman *overview*** merupakan halaman yang akan terbuka setelah pengguna melewati halaman daftar maupun halaman masuk. Halaman ini berfungsi untuk menampilkan hasil pemantauan terhadap aktivitas perangkat bergerak. Pada gambar di bawah ditampilkan bahwa tampilan *default* dari antarmuka ini adalah tampilan daftar aplikasi yang dibuka pada perangkat bergerak. Daftar *record* ini ditampilkan pada sebuah *listview* yang berisi sekumpulan *textview*. Di atas *listview* ini, ada sebuah *spinner* yang menampilkan pilihan jenis *record* yang bisa dipilih pengguna untuk ditampilkan pada halaman. Berikut tampilan antarmuka halaman *overview*.



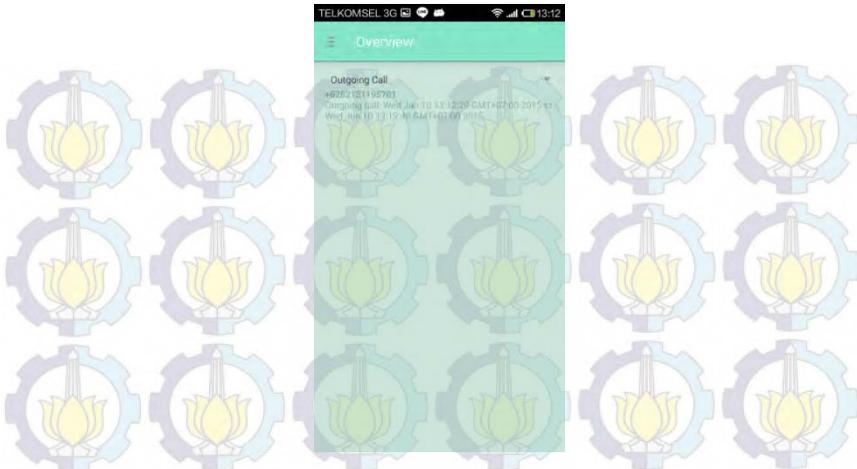
Gambar 4. 3 Antarmuka Halaman Overview

- Halaman antarmuka berikut adalah tampilan ketika pengguna memilih *application* pada *spinner*.



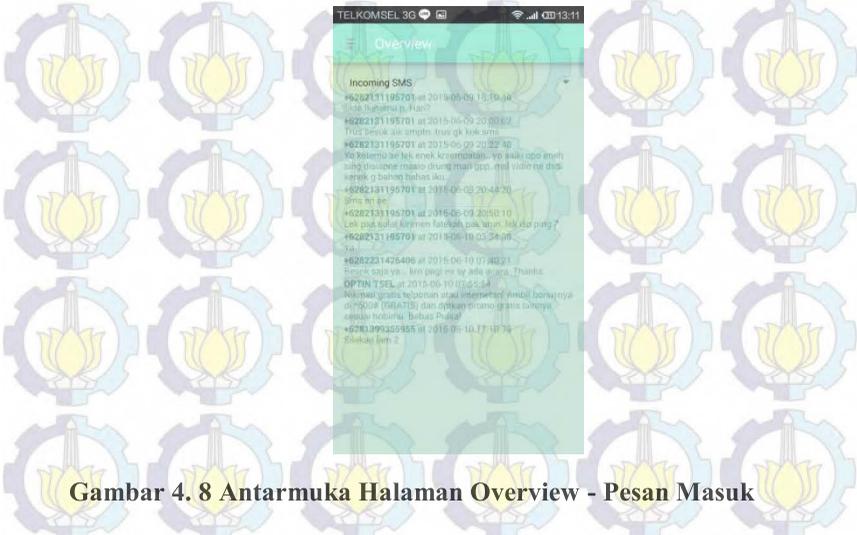
Gambar 4. 4 Antarmuka Halaman Overview – Aplikasi

- Halaman antarmuka berikut adalah tampilan ketika pengguna memilih *location* pada *spinner*.



Gambar 4.7 Antarmuka Halaman Overview - Telepon Keluar

- Halaman antarmuka berikut adalah tampilan ketika pengguna memilih *incoming SMS* pada *spinner*.



Gambar 4.8 Antarmuka Halaman Overview - Pesan Masuk

- Halaman antarmuka berikut adalah tampilan ketika pengguna memilih *outgoing SMS* pada *spinner*.



Gambar 4. 9 Antarmuka Halaman Overview - Pesan Keluar

- Halaman antarmuka berikut adalah tampilan ketika pengguna memilih *browser history* pada *spinner*.



Gambar 4. 10 Antarmuka Halaman Overview - Browser

d. Halaman *edit* akun merupakan halaman yang digunakan pengguna jika mereka ingin melakukan perubahan alamat

email maupun *password*. Berikut tampilan antarmuka halaman *edit* akun.



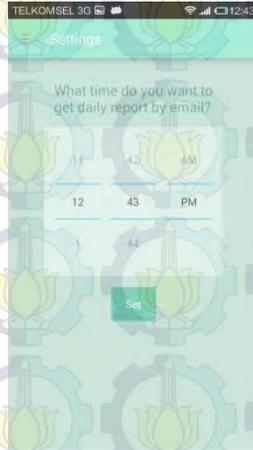
Gambar 4. 11 Antarmuka Halaman Edit Akun



Gambar 4. 12 Antarmuka Halaman Statistik

- e. **Halaman statistik** merupakan halaman yang menampilkan hasil statistik aktivitas perangkat bergerak. Diagram pai yang

ada di antarmuka ini memakai *library* dari MPAndroidChart. Tampilan antarmuka halaman statistik terdapat pada gambar 4.12.



Gambar 4. 13 Antarmuka Halaman Atur Jadwal

f. **Halaman atur jadwal** merupakan halaman yang digunakan pengguna untuk mengatur waktu pengiriman *record* ke alamat *email* yang terdaftar. tampilan antarmuka halaman atur jadwal terdapat pada gambar 4.13.

g. **Halaman menambahkan favorit** merupakan halaman yang digunakan pengguna untuk memilih aplikasi atau *domain* apa saja yang ingin ditambahkan ke daftar favorit. Aplikasi maupun *domain* yang ditambahkan ke daftar favorit akan ditampilkan dengan warna merah pada halaman *overview*. Berikut tampilan antarmuka halaman menambahkan favorit.



Gambar 4. 14 Antarmuka Halaman Menambah Favorit



Gambar 4. 15 Antarmuka Halaman Menghapus Favorit

- h. **Halaman menghapus favorit** merupakan halaman yang digunakan untuk menghilangkan aplikasi atau *domain* yang sebelumnya ada pada daftar favorit. Ketika aplikasi atau *domain* ini dihilangkan dari daftar favorit, maka aplikasi dan

situs dengan *domain* tadi akan ditampilkan dengan warna normal. Tampilan antarmuka halaman menghapus favorit ditunjukkan pada gambar 4.15.

4.3. Implementasi Proses Pada Aplikasi Perangkat Bergerak

Pada subbab ini akan dijabarkan implementasi dari proses-proses yang ada pada aplikasi yang berjalan pada perangkat bergerak. Proses yang ada meliputi pencatatan aktivitas, penampilan *record*, pembuatan dan penampilan statistik, sampai pengiriman *record* ke *webservice*.

4.3.1. Mencatat Aplikasi yang Dibuka

Berikut ini merupakan implementasi proses pencatatan aplikasi yang dibuka oleh perangkat bergerak. Proses ini memanfaatkan kelas bertipe *service* yang berjalan terus menerus di latar belakang perangkat. Ketika pertama kali *service* ini dijalankan, secara otomatis fungsi *onCreate()* dipanggil dan dilanjutkan ke fungsi *onStartCommand()*. Fungsi-fungsi ini merupakan *override* dari kelas *Service*. Berikut *pseudocode* implementasi dari fungsi *onCreate()*.

```
PUBLIC VOID onCreate() {
    inialisasi ActivityManager;
    inialisasi databaseConnector;
}
```

ActivityManager di atas diinisialisasi untuk digunakan dalam berinteraksi dengan status global dari perangkat. Sedangkan *database connector* merupakan kelas yang berfungsi untuk menghubungkan aplikasi dengan basis data SQLite. Secara otomatis, ketika fungsi di atas selesai, maka proses akan berlanjut ke fungsi *onStartCommand()*. Berikut *pseudocode* implementasi dari fungsi tersebut.

```
PUBLIC INT onStartCommand(Intent intent, int flag, int
startId) {
    START servis lokasi
    IF servis sms tidak nyala THEN
        START servis sms
    END IF
```

```

REGISTER screen listener
SET launcherName
STOP servis lokasi
START foreground service
RETURN START_STICKY
}

```

Kelas servis yang dibuat mengimplementasikan kelas *interface Runnable* sehingga ketika servis akan mengeksekusi fungsi *run()* secara otomatis. Pada fungsi ini, proses dilakukan dengan cara mengecek apakah layar perangkat menyala atau tidak. Jika tidak, maka proses pengecekan aplikasi yang sedang berjalan tidak dilakukan. Jika iya, maka sebaliknya. Setelah itu, proses mengecek apakah yang sedang terbuka merupakan *launcher* perangkat bergerak. Jika iya, maka *looping* akan dilanjutkan ke iterasi berikutnya. Kemudian, proses akan mengecek apakah nama *package* aplikasi yang sedang dibuka tidak sama dengan nama *package* aplikasi yang terakhir dibuka ketika proses mencatat pada iterasi sebelumnya. Jika iya, maka data aplikasi yang terakhir dimasukkan dahulu ke basis data. Lalu, data nama dan waktu dibukanya aplikasi yang saat ini terbuka disimpan pada sebuah variabel yang nantinya dipanggil ketika akan dimasukkan ke basis data. Selanjutnya, proses akan memanggil servis lokasi untuk mendapatkan lokasi perangkat pada saat ini. Berikut *pseudocode* dari implementasi fungsi *run()*.

```

PUBLIC VOID run() {
    WHILE isRunning = true
        SLEEP 1 detik
        IF screenOn = true THEN
            nowApp = nama package top activity
            IF nowApp = launcherName THEN
                IF lastApp != launcherName THEN
                    lastApp = launcherName
                    insertApps ()
                END IF
            CONTINUE
        END IF
        IF nowApp != nama package app ini THEN
            IF nowApp != lastApp THEN
                insertApps ()
            END IF
        END IF
    END WHILE
}

```

```

        lastApp = nowApp
        IF nowApp = browser THEN
            START servis browser
        ELSE
            url = ""
            IF servis browser berjalan THEN
                STOP servis browser
            END IF
        END IF
        appName = getLabel(nowApp)
        appTime = getTimeStamp()
        IF bisa akses lokasi THEN
            START servis lokasi
            SLEEP 1 detik
            simpan lokasi ke basis data
            STOP servis lokasi
        END IF
    END IF
END IF
END IF
END WHILE
}

```

Berikut *pseudocode* dari fungsi *insertApps()* yang digunakan untuk memasukkan data aplikasi ke basis data.

```

PUBLIC VOID insertApps() THROWS ParseException {
    IF appName != null && appTime != null THEN
        now = getCurrentTimeStamp()
        format = new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss")
        date1 = format.parse(appTime)
        date2 = format.parse(now)
        difference = date2.getTime() - date1.getTime()
        duration = difference / 1000
        IF lastAppTime != appTime THEN
            lastAppTime = appTime
            databaseConnector.addApps(appName, appTime,
duration)
        END IF
    END IF
}
}

```

Setiap melakukan pengisian ke basis data, dilakukan pengecekan apakah data aplikasi yang akan dimasukkan termasuk aplikasi yang ada dalam daftar favorit. Berikut *query* beserta fungsi yang dijalankan untuk memeriksa apakah aplikasi termasuk ke daftar aplikasi favorit.

```

PUBLIC BOOLEAN cekBannedApps(String name) {
    sql = "SELECT COUNT(*) FROM " + TABLE_BANNEDAPPS + "
WHERE package = '" + name + "'"
    cursor = database.rawQuery(sql, null)
    cursor.moveToFirst()
    count = cursor.getInt(0)
    cursor.close()
    IF count > 0 THEN
        RETURN TRUE
    ELSE
        RETURN FALSE
    END IF
}

```

Oleh karena proses ini seharusnya berjalan terus menerus, maka ketika pengguna dengan sengaja mematikan servis ini, servis akan mengirimkan *broadcast* agar servis ini dinyalakan kembali. Berikut implementasi dari proses pengiriman *broadcast* yang dilakukan oleh fungsi *onDestroy()*.

```

PUBLIC VOID onDestroy() {
    UNREGISTER screen listener
    isRunning = false
    intent = new Intent("restartApps")
    SEND BROADCAST intent
}

```

Broadcast “restartApps” akan diterima oleh kelas lain yang mengekstensi *BroadcastReceiver* lalu kelas tersebut akan memulai kelas servis utama ini lagi.

4.3.2. Mencatat Lokasi Pengguna

Proses pencatatan lokasi pengguna dilakukan dengan menggunakan sebuah kelas yang berjalan sebagai servis dan mengimplementasikan kelas *interface LocationListener*. Berikut *pseudocode* dari fungsi *onStartCommand()* yang dijalankan pertama setelah servis lokasi ini dimulai.

```

PUBLIC INT onStartCommand(Intent intent, int flag, int
startId) {
    inisialisasi locationManager
    SET tipe request NETWORK_PROVIDER
    RETURN START_NOT_STICKY
}

```

Kemudian dengan fungsi *onLocationChanged()* yang diimplementasi dari kelas *interface*, setiap perubahan lokasi yang dideteksi oleh perangkat akan disimpan ke variabel global lintang dan bujur. Hal ini dilakukan karena proses pengisian data lokasi dilakukan pada servis utama. Berikut *pseudocode* fungsi *onLocationChanged()*.

```
PUBLIC VOID onLocationChanged(Location location) {
    latitude = location.getLatitude()
    longitude = location.getLongitude()
    REMOVE update lokasi
}
```

4.3.3. Mencatat Panggilan Telepon

Proses pencatatan panggilan telepon ini diimplementasikan dengan cara membuat sebuah kelas abstrak yang mengekstensi *BroadcastReceiver*. Kelas ini memiliki sebuah fungsi *onRecieve()* dimana pada fungsi ini setiap proses *state* telepon yang masuk atau keluar akan dicatat. Berikut *pseudocode* dari fungsi *onReceive()*.

```
PUBLIC VOID onReceive(Context context, Intent intent) {
    IF broadcast diterima = "NEW_OUTGOING_CALL" THEN
        simpan nomor telepon
    ELSE
        stateStr = GET extra state
        number = GET extra number
        int state = 0;
        IF stateStr = "EXTRA_STATE_IDLE" THEN
            state = "CALL_STATE_IDLE"
        ELSE IF stateStr = "EXTRA_STATE_OFFHOOK" THEN
            state = "CALL_STATE_OFFHOOK"
        ELSE IF stateStr = "EXTRA_STATE_RINGING" THEN
            state = "CALL_STATE_RINGING"
        END IF
        onCallStateChanged(context, state, number)
    END IF
}
```

Kemudian pada fungsi *onCallStateChanged()* dilakukan pengecekan apakah jenis perubahan *state* yang terjadi termasuk telepon masuk atau keluar. Jika telepon masuk, apakah jenisnya

missed call atau telepon terangkat. Berikut *pseudocode* dari fungsi *onCallStateChanged()*.

```

PUBLIC VOID onCallStateChanged(Context context, int state,
String number) {
    IF lastState = state THEN
        RETURN
    END IF
    CASE state OF
        CALL_STATE_RINGING:
            isIncoming = true
            time = new Date()
            savedNumber = number
            onIncomingCallStarted(context, number, time)
        CALL_STATE_OFFHOOK:
            IF lastState != "CALL_STATE_RINGING" THEN
                isIncoming = false
                time = new Date()
                onOutgoingCallStarted(context,
savedNumber, time)
            END IF
        CALL_STATE_IDLE:
            IF lastState = "CALL STATE_RINGING" THEN
                onMissedCall(context, savedNumber, time)
            ELSE IF isIncoming = true THEN
                onIncomingCallEnded(context,
savedNumber, time, new Date())
            ELSE
                onOutgoingCallEnded(context,
savedNumber, time, new Date())
            END IF
    END CASE
    lastState = state
}

```

Ketika proses mendeteksi perubahan *state* dari RINGING menjadi IDLE, berarti terjadi *missed call* yang masuk ke perangkat. Jika ada perubahan *state* dari RINGING ke OFFHOOK lalu menjadi IDLE, berarti telepon yang masuk ke perangkat diangkat oleh pengguna. Selain itu, *state* yang masuk dianggap sebagai aksi panggilan keluar. Setiap *state* IDLE maka fungsi di atas memanggil fungsi untuk melakukan proses pengisian ke basis data berdasarkan jenis panggilan yang terjadi.

4.3.4. Mencatat Pesan Singkat Masuk

Proses mencatat pesan singkat masuk memanfaatkan *broadcast* yang dikeluarkan sistem operasi ketika ada pesan singkat masuk. Kelas ini mengekstensi kelas abstrak *BroadcastReceiver* sehingga untuk mendapatkan *broadcast* pesan masuk, cukup mengimplementasikan fungsi *onReceive()*. Berikut *pseudocode* fungsi *onReceive()*.

```
PUBLIC VOID onReceive(Context context, Intent intent) {
    IF intent.getAction() = "SMS_RECEIVED" THEN
        handleMessage(intent.getExtras(), context)
    END IF
}
```

Pada fungsi *handleMessage()* dilakukan proses pemilahan isi dari *intent* yang dikirimkan. *Intent* ini diterima dalam bentuk *Bundle*, lalu dalam *bundle* tersebut diambil objek PDU di dalamnya. PDU merupakan singkatan dari *Protocol Data Unit*, berupa suatu format objek yang digunakan untuk mengirimkan data pesan singkat melalui jaringan seluler. Dari objek PDU tersebut, nantinya didapatkan informasi nomer asal pesan singkat dan isi pesan singkat yang baru saja masuk. Berikut *pseudocode* implementasi fungsi *handleMessage()*.

```
PUBLIC VOID handleMessage(Bundle bundle, Context context) {
    inisialisasi databaseConnector
    IF bundle != null THEN
        pdu = (Object[]) bundle.get("pdus")
        FOR Object pdu IN pdu
            msg = SmsMessage.createFromPdu((byte[]) pdu)
            number = msg.getOriginatingAddress()
            body = msg.getMessageBody()
            simpan ke basis data
        END FOR
    END IF
}
```

4.3.5. Mencatat Pesan Singkat Keluar

Pada proses pencatatan data pesan singkat keluar, dibuat sebuah servis yang berjalan terus menerus. Hal ini dikarenakan untuk menangkap pesan singkat yang dikirimkan oleh perangkat,

sistem operasi tidak menyediakan *broadcast* yang bisa ditangkap oleh aplikasi eksternal. Servis yang dibuat ini dimulai dengan fungsi *onStartCommand()* yang berisi *register Content Observer* untuk membaca isi *Content Provider*. Berikut implementasi dari fungsi ini.

```
PUBLIC INT onStartCommand(Intent intent, int flag, int
startId) {
    inialisasi databaseConnector
    inialisasi smsObserver
    contentResolver = getContentResolver()
    registerContentObserver(Uri.parse
("content://sms/sent"), true, smsObserver)
    RETURN START_STICKY
}
```

SMSObserver merupakan kelas yang mengekstensi *ContentObserver* sehingga memiliki sebuah fungsi *onChange()*. Fungsi ini akan dipanggil setiap ada perubahan isi dari *Content Provider* yang telah di-*register* melalui *Content Resolver*. Berikut *pseudocode* fungsi *onChange()* pada kelas *SMSObserver*.

```
PUBLIC VOID onChange(booleam selfChange) {
    uri = Uri.parse("content://sms/sent")
    cur = getContentResolver().query(uri, null,
null, null, null)
    cur.moveToNext()
    content = cur.getString(kolom "body")
    smsNumber = cur.getString(kolom "address")
    smsDate = cur.getString(kolom "date")
    IF smsNumber = null || smsNumber.length < 1 THEN
        smsNumber = "Unknown"
    END IF
    cur.close()
    IF sms telah disimpan ke basis data THEN
        simpan ke basis data
    END IF
}
```

Sebelum dilakukan penyimpanan ke basis data, perlu dicek apakah data pesan singkat yang akan disimpan telah disimpan sebelumnya. Hal ini dilakukan karena pada kenyataannya, fungsi *onChange()* ini kadang terpanggil lebih dari satu kali ketika terdeteksi adanya perubahan isi dari *Content Provider*. Servis ini dijalankan secara terus menerus, sehingga ketika proses

mendeteksi bahwa servis dimatikan secara paksa oleh pengguna, maka proses akan mengirim *broadcast* agar servis ini dinyalakan kembali oleh sistem. Proses pengiriman *broadcast* hampir sama dengan yang dilakukan oleh servis utama.

4.3.6. Mencatat Alamat Situs yang Dikunjungi

Proses pencatatan alamat situs yang dikunjungi relatif sama dengan proses pencatatan pesan singkat keluar, yaitu memanfaatkan sebuah servis yang mengolah data dari *Content Observer*. Bedanya, servis ini dijalankan hanya jika servis utama mendeteksi bahwa pengguna sedang membuka aplikasi *browser native* milik perangkat. Berikut implementasi *pseudocode* dari fungsi *onStartCommand()*.

```
PUBLIC INT onStartCommand(Intent intent, int flag, int
startId) {
    inisialisasi databaseConnector
    inisialisasi browserObserver
    registerContentObserver(Browser.BOOKMARKS_URI, true,
browserObserver)
    RETURN START_STICKY
}
```

Kelas *BrowserObserver* memiliki sebuah fungsi implementasi yaitu fungsi *onChange()* yang dipanggil setiap ada perubahan pada isi *Content Provider* yang telah di-*register*. Berikut *pseudocode* fungsi *onChange()*.

```
PUBLIC VOID onChange(boolean selfChange) {
    proj = new String[] {
    Browser.BookmarkColumns.TITLE, Browser.BookmarkColumns.URL,
    Browser.BookmarkColumns.DATE }
    selection = Browser.BookmarkColumns.BOOKMARK + "
= 0"
    cur =
getContentResolver().query(Browser.BOOKMARKS_URI, proj,
selection, null, null)
    cur.moveToLast()
    url = cur.getString(1)
    date = cur.getString(2)
    IF url belum dimasukkan basis data THEN
        SLEEP 1 detik
        title = cur.getString(0)
        simpan ke basis data
```

```

        END IF
        cur.close()
    }

```

Sama seperti setiap fungsi *onChange()* sebelumnya, fungsi ini kadang dipanggil lebih dari satu kali ketika sistem mendeteksi adanya perubahan isi dari *Content Provider*. Sehingga diperlukan pengecekan apakah situs yang baru saja terdeteksi sudah dimasukkan ke dalam basis data atau belum sebelum melakukan penyimpanan. Pada saat melakukan proses penyimpanan ke basis data, ketika halaman belum ter-load sepenuhnya, judul situs yang tersimpan adalah alamat situs tersebut.

4.3.7. Menampilkan *Record*

Pada proses menampilkan *record*, proses akan membaca isi dari basis data, lalu menyimpannya pada sebuah *String*, dan menampilkannya menjadi *TextView*. Khusus data aplikasi, untuk menampilkan durasi pemakaian suatu aplikasi diperlukan sebuah fungsi untuk mengkonversi data dengan satuan detik menjadi satuan waktu dengan format jam, menit, dan detik. Berikut *pseudocode* untuk melakukan konversi durasi.

```

STRING convertDuration(int seconds) {
    String duration
    IF seconds >= 60 THEN
        sec = seconds % 60
        IF seconds >= 3600 THEN
            hour = seconds / 3600;
            min = (seconds - (hour * 3600)) / 60
            duration = hour + " h, " + min + " m, " +
sec + " s"
        ELSE
            min = seconds / 60
            duration = min + " m, " + sec + " s"
        END IF
    ELSE
        duration = seconds + " s"
    END IF
    RETURN duration
}

```

4.3.8. Mengatur Waktu Pengiriman *Record*

Untuk proses pengaturan waktu pengiriman *record*, dilakukan dua tahap. Pertama adalah membaca isi *timepicker* yang ada pada antarmuka ketika tombol ditekan. Berikut *pseudocode* proses tersebut.

```
hour = timePicker.getCurrentHour()
minute = timePicker.getCurrentMinute()
simpan waktu alarm ke basis data
alarm.setAlarm()
```

4.3.9. Mengirim *Record*

Proses pengiriman *record* dilakukan ketika kelas yang mengekstensi *BroadcastReceiver* menerima *broadcast* berupa semacam *alarm*. Proses ini dibagi menjadi beberapa tahap, pertama aplikasi akan memeriksa status ketersediaan koneksi internet. Setelah koneksi internet dipastikan tersedia, proses dilanjutkan dengan penyusunan struktur JSON untuk dikirimkan ke *webservice*. Setelah objek JSON selesai disusun, baru data tersebut dikirim ke *webservice*. Setelah objek dikirim, aplikasi akan menghapus isi *record* di basis data dan mengatur ulang waktu pengiriman *record* menjadi 24 jam berikutnya.

4.3.9.1. Memeriksa Jaringan

Proses pemeriksaan jaringan dilakukan dengan cara pertama memeriksa apakah perangkat bisa terhubung dengan internet. Jika iya, maka proses berikutnya dilakukan. Jika tidak, aplikasi akan memeriksa apakah status perangkat saat itu sedang dalam mode pesawat terbang. Jika iya, maka pengiriman *report* ditunda menjadi 30 menit berikutnya. Jika tidak, maka aplikasi akan menyalakan koneksi internet perangkat lalu mengirimkan *record* untuk kemudian dimatikan lagi setelah beberapa saat. Berikut *pseudocode* proses pemeriksaan jaringan.

```
IF jam sekarang = jam alarm & menit sekarang = menit alarm
THEN
    flagMobileData = 0
    IF globalVariable.getFlagEmailSent() = 0 THEN
```

```

globalVariable.setFlagEmailSent(1)
IF internetAvailable() THEN
    START alarm
ELSE
    IF cek mode airplane = true THEN
        delay alarm 30 menit
    ELSE
        nyalakan mobile data
        SLEEP 10 detik
        IF internetAvailable() THEN
            flagMobileData = 1
            START alarm
        ELSE
            delay alarm 24 jam
        END IF
    END IF
END IF
ELSE
    globalVariable.setFlagEmailSent(0)
END IF

```

4.3.9.2. Membuat Struktur JSON

Setelah dipastikan bahwa *record* dapat dikirim saat itu juga, maka aplikasi akan menyusun objek JSON dari data yang ada di basis data. Berikut *pseudocode* proses pembuatan objek JSON sesuai struktur yang dibutuhkan.

```

PRIVATE STRING createJsonObject (Cursor cursor) {
    inisialisasi jsonArray
    cursor.moveToFirst()
    WHILE posisi cursor < ukuran cursor
        inisialisasi jsonObject
        IF cursor != null THEN
            FOR i=0 TO jumlah kolom cursor
                IF cursor.getColumnname(i) != null &
                cursor.getString(i) != null THEN
                    col = cursor.getColumnname(i)
                    IF col = "count()" THEN
                        col = "total"
                    END IF
                    jsonObject.put(col,
                    cursor.getString(i))
                END IF
            END FOR
        END IF
    END WHILE
}

```

```

        jsonArray.put(jsonObject);
        cursor.moveToNext()
    END WHILE
    CLOSE cursor
    RETURN jsonArray.toString()
}

```

4.3.9.3. Mengirimkan JSON

Proses pengiriman JSON dilakukan dengan sederhana. Setelah semua data *record* yang ingin dikirim ke *webservice* terkumpul menjadi sebuah kumpulan JSON, data yang berupa *String* ini dikirim sebagai *Object* ke sebuah kelas yang mengekstensi *AsyncTask*. Pada kelas ini, diimplementasikan sebuah fungsi *doInBackground()* yang berjalan di samping proses lainnya. Berikut *pseudocode* fungsi *doInBackground()*.

```

PROTECTED OBJECT doInBackground(Object... args) {
    inisialisasi httpClient
    inisialisasi httpPost ke alamat webservice
    hapus brackets dari String email
    POST args.toString()
    RETURN null
}

```

Selanjutnya, untuk menunggu proses pengiriman selesai, aplikasi akan melakukan *sleep* selama 20 detik. Hal ini dilakukan untuk berjaga-jaga jika koneksi internet yang dimiliki perangkat tidak cepat. Kemudian, proses akan memeriksa apakah sebelumnya aplikasi menyalakan koneksi internet dengan sendirinya. Jika iya, maka aplikasi akan mematikan koneksi internet yang ada, lalu mengosongkan isi basis data. Berikut *pseudocode* proses ini secara keseluruhan.

```

GET alamat email dari basis data
GET objek JSON yang akan dikirimkan
POST JSON
SET alarm menjadi 24 jam berikutnya
SLEEP 20 detik
IF flagMobileData = 1 THEN
    matikan koneksi internet
END IF
EMPTY basis data

```

4.3.10. Mengubah Akun

Proses perubahan akun diimplementasikan dengan cara membaca *field editText* pada antarmuka, kemudian dilakukan pengecekan isian data. Terdapat fungsi *updateEmail()* dan *updatePassword()* yang melakukan *query* ke basis data dan akan mengembalikan nilai 1 jika bernilai benar, nilai 0 jika tidak benar. Berikut *pseudocode* untuk menjalankan proses perubahan akun.

```

successUpdateEmail = updateEmail(newEmail)
IF oldPassword = newPassword OR editText is empty THEN
  IF successUpdateEmail = 1 THEN
    TOAST update email sukses
  ELSE
    TOAST update email gagal
  END IF
ELSE
  IF confirmNewPassword != newPassword THEN
    TOAST password tidak sama
  ELSE
    successUpdatePassword = updatePassword(newPassword)
    IF successUpdateEmail = 1 AND successUpdatePassword
= 1 THEN
      TOAST update email & password berhasil
    ELSE
      TOAST update email & password gagal
    END IF
  END IF
END IF

```

4.3.11. Mendaftarkan dan *Login* Akun

Proses untuk masuk ke aplikasi ditangani oleh satu *fragment*. *Fragment* ini pula yang menangani apakah antarmuka yang ditampilkan berupa antarmuka untuk mendaftar atau *login*. Berikut *pseudocode* untuk memeriksa antarmuka mana yang harus ditampilkan.

```

IF ada user di basis data THEN
  isLogin = true
  confirmPassText visibility = GONE
  SET teks pada tombol menjadi "Sign In"
  SET judul menjadi "Sign In"

```

```
ELSE
    tampilkan activity tutorial
END IF
```

Setelah mengetahui antarmuka seperti apa yang akan ditampilkan, jika pengguna menekan tombol yang ada di antarmuka, maka proses berikut yang akan dijalankan.

```
IF akan login THEN
    IF field tidak kosong THEN
        bolehLogin = cek email dan password
        IF bolehLogin = true THEN
            tampilkan halaman overview
        ELSE
            TOAST email atau password salah
        END IF
    ELSE
        TOAST field kosong
    END IF
ELSE
    IF field tidak kosong THEN
        IF password dan confirmPassword sama THEN
            simpan user dan password ke basis data
            tampilkan halaman overview
        ELSE
            TOAST password tidak sama
        END IF
    ELSE
        TOAST field kosong
    END IF
END IF
```

4.3.12. Menghitung Statistik

Proses penghitungan statistik dilakukan melalui eksekusi *query* ke basis data. Berikut implementasi *query* untuk melakukan penghitungan masing-masing statistik.

4.3.12.1. Aplikasi

Berikut proses penghitungan frekuensi aplikasi diurutkan mulai dari yang paling banyak.

```
database.query(TABLE_APPS, new String[]{"package",
    "count()"}, null, null, "package", null, "count() DESC")
```

4.3.12.2. Panggilan Masuk dan Keluar

Berikut proses penghitungan frekuensi nomor yang melakukan panggilan masuk ke perangkat diurutkan mulai dari yang paling banyak.

```
database.query(TABLE_INCOMINGCALL, new String[]{"number",
"count()"}, null, null, "number", null, "count() DESC")
```

Berikut proses penghitungan frekuensi nomor yang paling sering dihubungi oleh perangkat dan diurutkan mulai dari yang paling banyak.

```
database.query(TABLE_OUTGOINGCALL, new String[]{"number",
"count()"}, null, null, "number", null, "count() DESC")
```

4.3.12.3. Pesan Singkat Masuk dan Keluar

Berikut proses penghitungan frekuensi nomor yang paling sering mengirim pesan singkat ke perangkat diurutkan berdasarkan yang paling banyak.

```
database.query(TABLE_INCOMINGSMS, new String[]{"number",
"count()"}, null, null, "number", null, "count() DESC")
```

Berikut proses penghitungan frekuensi nomor yang paling sering menerima pesan singkat dari perangkat diurutkan dari yang paling banyak.

```
database.query(TABLE_OUTGOINGSMS, new String[]{"number",
"count()"}, null, null, "number", null, "count() DESC")
```

4.3.12.4. Durasi

Berikut proses penghitungan durasi pemakaian masing-masing aplikasi diurutkan mulai dari yang terlama.

```
database.rawQuery("SELECT package, sum(duration) as total
FROM " + TABLE_APPS + " GROUP BY package ORDER BY
sum(duration) DESC", null)
```

4.3.13. Menampilkan Statistik

Proses menampilkan statistik diawali dengan eksekusi *query* statistik yang diinginkan. Selanjutnya, hasil eksekusi yang berupa *Cursor* dikirim ke fungsi *drawchart()* beserta deskripsi diagram yang diinginkan. Mula-mula proses akan menghitung isi data pada *cursor* tersebut. Batas jumlah data yang akan ditampilkan pada diagram maksimal 5 data. Selanjutnya, data jumlah frekuensi yang ada pada *Cursor* beserta namanya dimasukkan ke dalam *array* yang nantinya dijadikan sebagai *dataset* untuk dibaca dan ditampilkan pada diagram. Setelah itu, hal yang perlu dilakukan cukup mengatur beberapa atribut tampilan diagram. Berikut *pseudocode* dari fungsi *drawChart()*.

```

VOID drawChart(Cursor cursor, String description) {
    count = cursor.getCount()
    totalCount = 0
    IF count > 5 THEN
        count = 5
    END IF
    cursor.moveToFirst()
    FOR i=0 TO count
        totalCount += cursor.getInt(1)
        cursor.moveToNext()
    END FOR
    inisialisasi array ENTRY yVal
    inisialisasi array STRING xVal
    SET array warna diagram
    cursor.moveToFirst()
    FOR i=0 TO count
        yVal.add(new Entry(cursor.getInt(1), i))
        xVal.add(cursor.getString(0))
        cursor.moveToNext()
    END FOR
    inisialisasi pieDataSet dengan yVal
    SET slice space diagram
    SET warna diagram
    inisialisasi pieData dengan xVal dan pieDataSet
    SET ukuran teks
    SET penulisan format teks
    SET pieData ke pieChart
    SET deskripsi diagram
    SET waktu animasi
    cursor.close()
    SET posisi legend
    INVALIDATE diagram
}

```

```
}

```

4.3.14. Menambahkan Favorit

Proses penambahan aplikasi maupun *domain* ke daftar favorit dilakukan dengan memanfaatkan sebuah *listview* yang menampilkan sepasang *checkbox* dan *textview*. Ketika sebuah *checkbox* dipilih oleh pengguna, maka nilai *textview* dimasukkan ke dalam sebuah *array* sebagai tempat penampung sementara. Ketika pengguna menekan tombol untuk menambahkan data ke daftar favorit, proses berikut ini dilakukan oleh aplikasi.

```
count = checkedItems.size()
FOR i=0 TO count
    favorite.add(checkedItems.get(i))
END FOR
addToDatabase()
tampilkan fragment overview

```

Pada fungsi *addToDatabase()* dilakukan dua proses. Pertama adalah menambahkan daftar aplikasi atau domain ke dalam tabel BANNEDAPPS atau BANNEDLINK. Kedua melakukan proses *update* nilai *flag* pada setiap aplikasi atau URL yang termasuk ke daftar ini menjadi bernilai 1. Berikut *pseudocode* fungsi *addToDatabase()*.

```
IF spinner = application THEN
    FOR i=0 TO favorite.size()
        ADD daftar aplikasi favorit
        UPDATE flag aplikasi
    END FOR
ELSE
    FOR i=0 TO favorite.size()
        ADD daftar domain favorit
        UPDATE flag URL
    END FOR
END IF
CLEAR array favorite

```

Setelah proses di atas dilakukan, maka tampilan akan berpindah ke halaman *overview*. Di halaman ini, semua aplikasi atau situs dengan *domain* yang baru saja ditambahkan akan ditampilkan dengan warna merah.

4.3.15. Menghapus Favorit

Proses penghapusan aplikasi maupun *domain* dari daftar favorit dilakukan dengan memanfaatkan sebuah *listview* yang menampilkan sepasang *checkbox* dan *textview*. Ketika sebuah *checkbox* dipilih oleh pengguna, maka nilai *textview* dimasukkan ke dalam sebuah *array* sebagai tempat penampung sementara. Ketika pengguna menekan tombol untuk menghapus data dari daftar favorit, proses berikut ini dilakukan oleh aplikasi.

```
count = checkedItems.size()
FOR i=0 TO count
    removeFromDatabase(checkedItems.get(i))
END FOR
tampilkan fragment overview
```

Pada fungsi *removeFromDatabase()* dilakukan dua proses. Pertama adalah menghapus daftar aplikasi atau domain dari dalam tabel BANNEDAPPS atau BANNEDLINK. Kedua melakukan proses *update* nilai *flag* pada setiap aplikasi atau URL yang dari *array* ini menjadi bernilai 0. Berikut *pseudocode* fungsi *removeFromDatabase()*.

```
IF spinner = application THEN
    DELETE data dari tabel BANNEDAPPS
    UPDATE flag dari tabel APPS
ELSE
    DELETE data dari tabel BANNEDLINK
    UPDATE flag dari tabel URL
END IF
```

Setelah proses di atas dilakukan, maka tampilan akan berpindah ke halaman *overview*. Di halaman ini, semua aplikasi atau situs dengan *domain* yang baru saja dihapus dari daftar favorit akan ditampilkan dengan warna *default*.

4.4. Implementasi Proses Pada Aplikasi *Webservice*

Pada aplikasi *webservice*, pertama kali yang dilakukan adalah menerima *post* data dari *client* dan mengubahnya menjadi data berbentuk *array*. Oleh karena data yang dikirim berbentuk JSON, maka diperlukan proses *decoding* JSON supaya terbaca isi objeknya. Berikut proses yang dilakukan pada tahap ini.

```

$app = $this->input->post('apps');
$brow = $this->input->post('browser');
$incs = $this->input->post('incsms');
$out = $this->input->post('outsms');
$incc = $this->input->post('inccall');
$outc = $this->input->post('outcall');
$loca = $this->input->post('loc');
$mail = $this->input->post('email');
$countapps = $this->input->post('capps');
$countincc = $this->input->post('cicall');
$countoutc = $this->input->post('cocall');
$countincs = $this->input->post('cisms');
$countouts = $this->input->post('cosms');
$countdura = $this->input->post('cdura');

$array_content["apps"] = json_decode($app);
$array_content["brow"] = json_decode($brow);
$array_content["incs"] = json_decode($incs);
$array_content["outs"] = json_decode($outs);
$array_content["incc"] = json_decode($incc);
$array_content["outc"] = json_decode($outc);
$array_content["loca"] = json_decode($loca);
$array_content["countapps"] = json_decode($countapps);
$array_content["countincc"] = json_decode($countincc);
$array_content["countoutc"] = json_decode($countoutc);
$array_content["countincs"] = json_decode($countincs);
$array_content["countouts"] = json_decode($countouts);
$array_content["countdura"] = json_decode($countdura);

```

Setelah *array_content* terbentuk, proses berikutnya yang dilakukan adalah mengubah isi data pada kolom durasi menjadi berformat jam, menit, dan detik. Hal ini dilakukan karena data yang ada pada saat ini dalam format detik. Berikut *pseudocode* fungsi untuk merubah data durasi.

```

FUNCTION seconds_to_time($seconds)
{
    inialisasi datetime
    IF seconds > 3600 THEN
        RETURN format jam, menit, detik
    ELSE IF seconds > 60 THEN
        RETURN format menit, detik
    ELSE
        RETURN detik
    END IF
}

```

Selanjutnya, aplikasi melakukan proses *request* alamat dari data koordinat lintang dan bujur yang dimiliki ke server Google. Berikut *pseudocode* proses ini.

```

inisialisasi array addresses
FOR i=0 TO count(array_content["loca"])
    latlng = array_content["loca"][i]->coordinate
    IF (i % 10) = 0 THEN
        SLEEP 1 detik
    END IF
    endpoint = request google API dengan latlng
    address = ambil isi endpoint
    PUSH json_decode(address) ke array addresses
END FOR
array_content["addresses"] = addresses

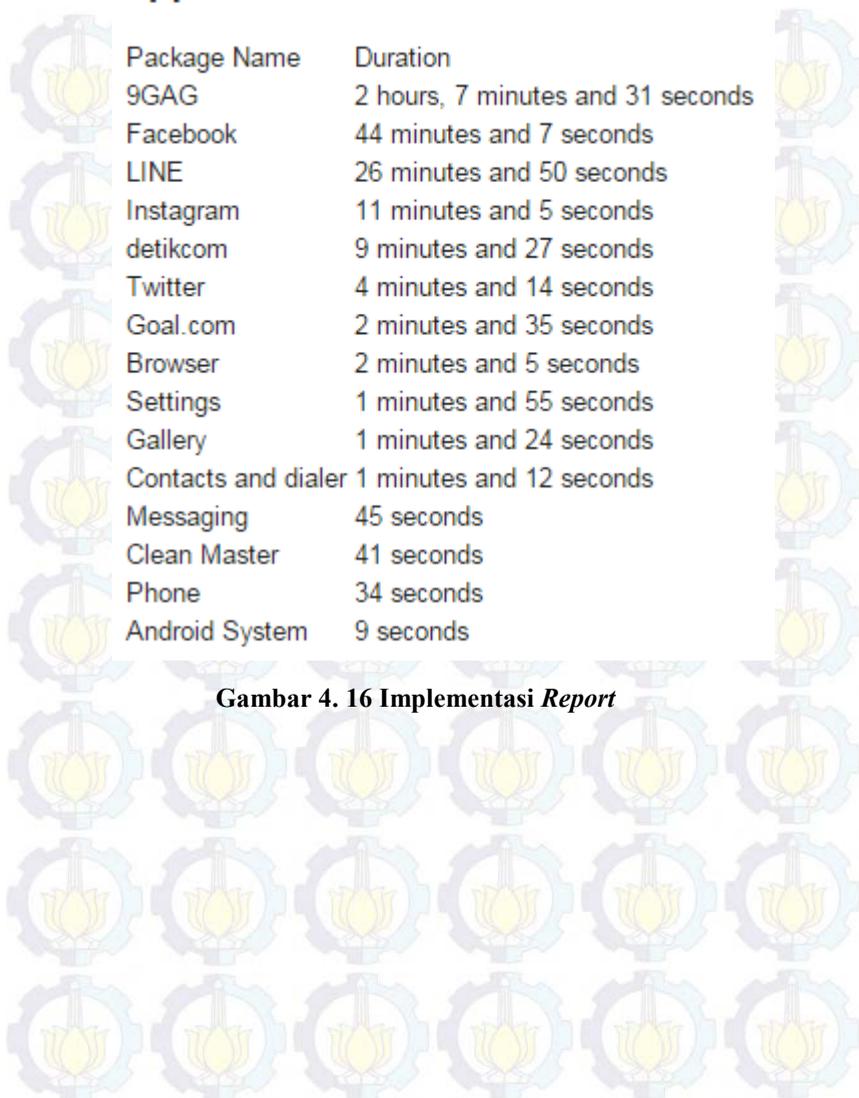
```

Setelah proses-proses di atas dijalankan, variabel *array_content* beserta alamat *email* dikirim ke halaman *view* lalu halaman tersebut dikirim ke alamat *email* yang ingin dituju.

4.5. Implementasi Tampilan *Report*

Pada subbab ini dijabarkan isi dari tampilan *report* yang dikirimkan ke alamat *email* tujuan. Konsep dari tampilan ini hanya dengan membuat tabel-tabel lalu menampilkan isi *array_content* satu per satu ke tabel masing-masing sesuai jenis datanya. Hal ini dilakukan beberapa kali sampai semua data ditampilkan. Berikut contoh implementasi dari pembuatan satu tabel.

Application Duration



Package Name	Duration
9GAG	2 hours, 7 minutes and 31 seconds
Facebook	44 minutes and 7 seconds
LINE	26 minutes and 50 seconds
Instagram	11 minutes and 5 seconds
detikcom	9 minutes and 27 seconds
Twitter	4 minutes and 14 seconds
Goal.com	2 minutes and 35 seconds
Browser	2 minutes and 5 seconds
Settings	1 minutes and 55 seconds
Gallery	1 minutes and 24 seconds
Contacts and dialer	1 minutes and 12 seconds
Messaging	45 seconds
Clean Master	41 seconds
Phone	34 seconds
Android System	9 seconds

Gambar 4. 16 Implementasi *Report*

BAB V PENGUJIAN DAN EVALUASI

Bab ini membahas tentang pengujian dan evaluasi dari perangkat lunak pada tugas akhir ini. Pengujian dilakukan pada fungsionalitas perangkat lunak. Pembahasan tentang pengujian dan evaluasi disampaikan pada subbab-subbab berikut.

5.1. Lingkungan Pelaksanaan Pengujian

Lingkungan pengujian menjelaskan tentang perangkat yang dilibatkan dalam proses pengujian. Lingkungan pengujian ini menggunakan perangkat keras berupa perangkat bergerak berbasis Android yang dapat terhubung dengan. Spesifikasi lingkungan pengujian dijelaskan pada tabel berikut.

Tabel 5. 1 Spesifikasi Perangkat Bergerak

Perangkat Bergerak Android	Processor	Qualcomm 2.5 GHz Quad-core
	RAM	3 GB
	Sistem Operasi	Android versi 4.4.2
	ROM	MIUI
	Status	Tidak <i>root</i>

Sementara spesifikasi server menggunakan *web hosting* milik kuncihost dengan kapasitas *disk* sebesar 500 MB dan *bandwidth* 70 GB.

5.2. Pengujian Fungsionalitas

Subbab ini menjelaskan tentang skenario pengujian fungsionalitas perangkat lunak pada tugas akhir ini. Pengujian didokumentasikan secara sistematis sebagai tolok ukur keberhasilan sistem. Uji coba fungsionalitas terbagi menjadi beberapa skenario berikut.

- Uji coba melakukan pendaftaran akun
- Uji coba melakukan *login* ke aplikasi
- Uji coba pencatatan aplikasi yang dibuka
- Uji coba pencatatan pesan singkat masuk
- Uji coba pencatatan pesan singkat keluar
- Uji coba pencatatan panggilan telepon masuk
- Uji coba pencatatan panggilan telepon keluar
- Uji coba pencatatan alamat situs
- Uji coba pencatatan lokasi
- Uji coba menampilkan statistik
- Uji coba ubah akun
- Uji coba atur jadwal pengiriman *record*
- Uji coba pengiriman *record*
- Uji coba penambahan aplikasi favorit
- Uji coba penambahan *domain* favorit
- Uji coba penghapusan aplikasi favorit
- Uji coba penghapusan *domain* favorit

Khusus pada bagian uji coba pengiriman *record*, hal ini nantinya sekaligus membuktikan apakah *webservice* mampu menangani permintaan pengiriman *record* dengan baik.

5.2.1. Pengujian Melakukan Pendaftaran dan *Login* Akun

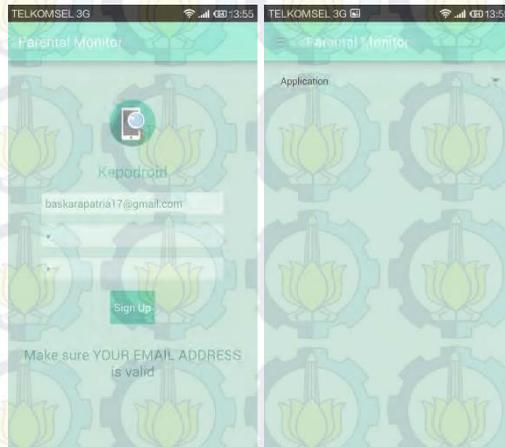
Pengujian melakukan pendaftaran dan *login* akun dilakukan dengan dua tahap. Skenario pertama dijalankan saat pengguna baru saja melakukan instalasi pada perangkat bergerak, sehingga belum ada data pengguna pada basis data. Skenario pengujian pertama dijelaskan pada tabel berikut.

Tabel 5. 2 Rincian Pengujian Melakukan Pendaftaran

Nomor	PA-01
Nama	Melakukan pendaftaran akun
Tujuan	Memeriksa fungsi untuk menangani akun

	baru apakah berfungsi atau tidak
Kondisi awal	Data akun belum ada
Skenario	<ol style="list-style-type: none"> 1. Pengguna melakukan instalasi aplikasi ke perangkat 2. Pengguna memasukkan alamat <i>email</i> dan <i>password</i> beserta konfirmasi <i>password</i> 3. Pengguna menekan tombol <i>sign up</i>
Masukan	Alamat <i>email</i> dan <i>password</i> beserta konfirmasi <i>password</i>
Keluaran yang diharapkan	Halaman <i>overview</i> ditampilkan
Hasil pengujian	Berhasil

Berikut tampilan atarmuka saat akan mendaftar dan halaman *overview* ketika pengguna telah menekan tombol *sign up*.



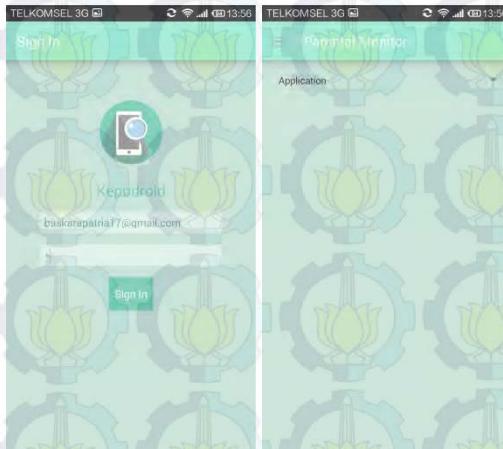
Gambar 5. 1 Hasil Pengujian PA-01

Berikutnya, dengan data pengguna yang didaftarkan pada skenario pertama, dilakukan pengujian *login* akun dengan skenario kedua. Berikut penjelasan skenario pengujian kedua.

Tabel 5. 3 Rincian Pengujian Melakukan Login

Nomor	PA-02
Nama	Melakukan <i>login</i> akun
Tujuan	Memeriksa fungsi untukm menangani <i>login</i> apakah berfungsi atau tidak
Kondisi awal	Data akun sudah ada
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi 2. Pengguna memasukkan alamat <i>email</i> dan <i>password</i> yang sudah didaftarkan 3. Pengguna menekan tombol <i>sign in</i>
Masukan	Alamat <i>email</i> dan <i>password</i>
Keluaran yang diharapkan	Halaman <i>overview</i> ditampilkan
Hasil pengujian	Berhasil

Berikut tampilan layar saat akan melakukan *login* beserta tampilan layar saat pengguna telah melakukan proses *login* dengan berhasil.

**Gambar 5. 2 Hasil Pengujian PA-02**

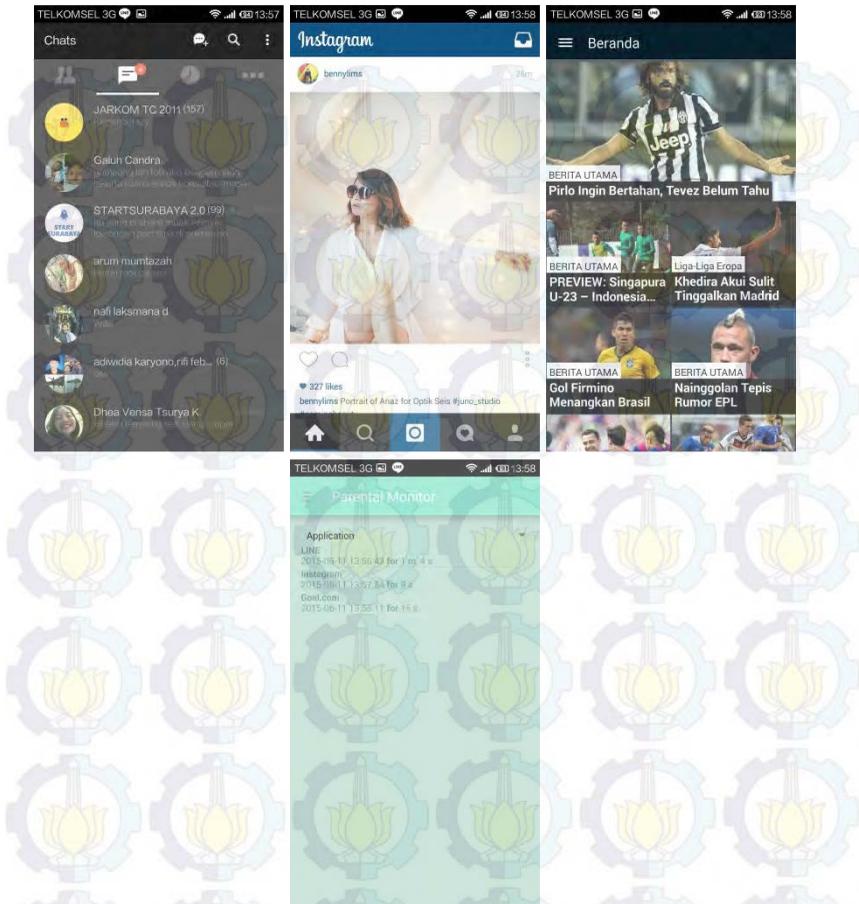
5.2.2. Pengujian Pencatatan Aplikasi

Pengujian ini dilakukan untuk memeriksa apakah fungsi pencatat aktivitas perangkat bergerak yang berjalan pada servis utama bekerja dengan baik. Skenario pengujian dilakukan dengan cara membuka aplikasi apa saja selain aplikasi tugas akhir ini kemudian menutup aplikasi tersebut setelah beberapa saat. Berikut penjelasan skenario pengujian tersebut.

Tabel 5. 4 Rincian Pengujian Melakukan Pencatatan Aplikasi

Nomor	PA-03
Nama	Melakukan pencatatan aplikasi
Tujuan	Memeriksa fungsi untuk mencatat aplikasi yang dibuka oleh perangkat yang berjalan pada servis utama berfungsi atau tidak
Kondisi awal	Servis utama aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi selain aplikasi pemantau 2. Pengguna menutup aplikasi yang baru saja dibuka 3. Pengguna membuka aplikasi pemantau 4. Pengguna melakukan proses <i>login</i>
Masukan	Alamat <i>email</i> dan <i>password</i> untuk <i>login</i>
Keluaran yang diharapkan	Halaman <i>overview</i> ditampilkan, terdapat data aplikasi yang baru saja dibuka sebelumnya
Hasil pengujian	Berhasil

Pada pengujian kali ini, percobaan dilakukan dengan membuka aplikasi LINE, lalu instagram, dan GOAL(dot)com secara berurutan. Berikut tampilan layar saat pengguna membuka beberapa aplikasi lalu tampilan aplikasi pemantau saat menampilkan *record* aplikasi yang baru dibuka.



Gambar 5.3 Hasil Pengujian PA-03

5.2.3. Pengujian Pencatatan Pesan Singkat Masuk

Pengujian ini dilakukan untuk memeriksa apakah penerima *broadcast* pesan singkat masuk yang ada pada aplikasi ini berjalan. Skenario dilakukan dengan cara mengirimkan pesan singkat dari perangkat lain ke perangkat yang memiliki aplikasi

pemantau pemakaian perangkat bergerak. Berikut penjelasan mengenai skenario ini.

Tabel 5. 5 Rincian Pengujian Melakukan Pencatatan SMS Masuk

Nomor	PA-04
Nama	Melakukan pencatatan pesan singkat masuk
Tujuan	Memeriksa fungsi untuk mencatat data pesan singkat yang masuk ke perangkat apakah berfungsi atau tidak
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Pengguna menerima pesan singkat yang dikirim oleh perangkat lain 2. Pengguna membuka aplikasi pemantau 3. Pengguna melakukan proses <i>login</i> 4. Pengguna memilih <i>incoming sms</i> pada <i>spinner</i> di halaman <i>overview</i>
Masukan	Alamat <i>email</i> dan <i>password</i> untuk <i>login</i>
Keluaran yang diharapkan	Halaman <i>overview</i> ditampilkan, terdapat data pesan singkat yang baru saja masuk
Hasil pengujian	Berhasil

Berikut tampilan hasil pengujian pencatatan pesan singkat yang masuk ke perangkat.



Gambar 5. 4 Hasil Pengujian PA-04

5.2.4. Pengujian Pencatatan Pesan Singkat Keluar

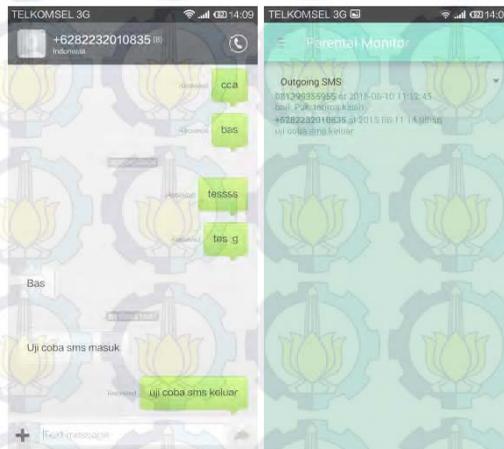
Pengujian ini dilakukan untuk memeriksa apakah servis yang membaca setiap pesan singkat yang dikirimkan oleh perangkat bergerak dapat berfungsi dengan baik. Skenario dilakukan dengan cara mengirim pesan singkat dari perangkat ini ke sebuah nomor telepon. Saat melakukan pengiriman pesan singkat, servis yang menangani proses ini harus dipastikan berjalan terlebih dahulu. Berikut penjelasan mengenai skenario pengujian ini.

Tabel 5. 6 Rincian Pengujian Melakukan Pencatatan SMS Keluar

Nomor	PA-05
Nama	Melakukan pencatatan pesan singkat keluar
Tujuan	Memeriksa fungsi untuk mencatat data pesan singkat yang dikirim oleh perangkat apakah berfungsi atau tidak
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	1. Pengguna mengirim pesan singkat ke suatu nomor telepon

	<ol style="list-style-type: none"> 2. Pengguna memastikan bahwa pesan singkat telah terkirim ke nomor telepon tujuan 3. Pengguna membuka aplikasi pemantau 4. Pengguna melakukan proses <i>login</i> 5. Pengguna memilih <i>outgoing sms</i> pada <i>spinner</i> di halaman <i>overview</i>
Masukan	Alamat <i>email</i> dan <i>password</i> untuk <i>login</i>
Keluaran yang diharapkan	Halaman <i>overview</i> ditampilkan, terdapat data pesan singkat yang baru saja terkirim
Hasil pengujian	Berhasil

Berikut tampilan antarmuka saat proses pengujian pencatatan pesan singkat yang dikirimkan perangkat tengah berlangsung



Gambar 5.5 Hasil Pengujian PA-05

5.2.5. Pengujian Pencatatan Panggilan Telepon Masuk

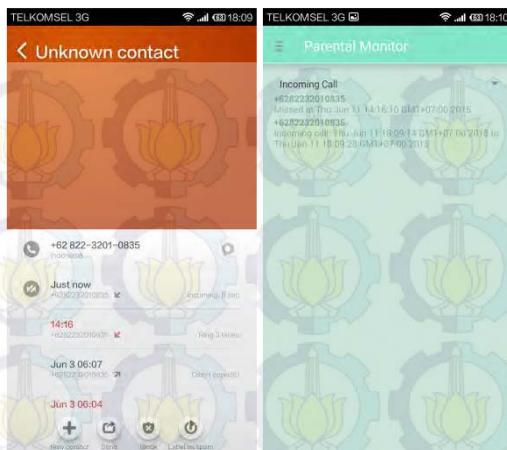
Pengujian ini dilakukan untuk memeriksa apakah aplikasi dapat mencatat data panggilan telepon yang masuk ke perangkat. Selain itu, aplikasi juga harus mampu membedakan jenis panggilan yang masuk, apakah diangkat oleh pemilik perangkat

atau *missed call*. Oleh karena itu, terdapat dua skenario untuk pengujian kali ini. Berikut penjelasan skenario pertama.

Tabel 5. 7 Rincian Pengujian Pencatatan Panggilan Masuk dan Diangkat

Nomor	PA-06
Nama	Melakukan pencatatan panggilan telepon masuk dan diangkat
Tujuan	Memeriksa fungsi untuk mencatat panggilan telepon yang masuk ke perangkat dan diangkat oleh perangkat
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Perangkat yang memiliki aplikasi pemantau dihubungi melalui perangkat lain 2. Pengguna mengangkat telepon yang masuk lalu menutup telepon setelah beberapa saat 3. Pengguna membuka aplikasi pemantau 4. Pengguna melakukan proses <i>login</i> 5. Pengguna memilih <i>incoming call</i> pada <i>spinner</i> di halaman <i>overview</i>
Masukan	Alamat <i>email</i> dan <i>password</i> untuk <i>login</i>
Keluaran yang diharapkan	Halaman <i>overview</i> ditampilkan, terdapat data telepon yang masuk diangkat oleh pengguna
Hasil pengujian	Berhasil

Berikut adalah tampilan antarmuka ketika proses pengujian pencatatan panggilan telepon yang masuk ke perangkat tengah berlangsung.



Gambar 5. 6 Hasil Pengujian PA-06

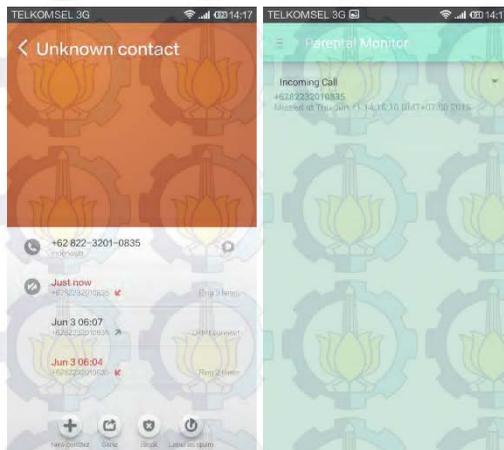
Skenario berikutnya adalah menguji apakah aplikasi pemantau mampu mengetahui bahwa panggilan telepon yang masuk ke perangkat berupa *missed call*. Berikut penjelasan skenario kedua.

Tabel 5. 8 Rincian Pengujian Pencatatan Missed Call

Nomor	PA-07
Nama	Melakukan pencatatan <i>missed call</i>
Tujuan	Memeriksa fungsi untuk mencatat panggilan telepon yang masuk ke perangkat tetapi tidak diangkat
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Perangkat yang memiliki aplikasi pemantau dihubungi melalui perangkat lain 2. Pengguna tidak mengangkat telepon yang masuk 3. Pengguna membuka aplikasi pemantau

	4. Pengguna melakukan proses <i>login</i> 5. Pengguna memilih <i>incoming call</i> pada <i>spinner</i> di halaman <i>overview</i>
Masukan	Alamat <i>email</i> dan <i>password</i> untuk <i>login</i>
Keluaran yang diharapkan	Halaman <i>overview</i> ditampilkan, terdapat data <i>missed call</i> yang baru saja terjadi
Hasil pengujian	Berhasil

Berikut tampilan antarmuka saat proses pengujian pencatatan *missed call* yang masuk ke perangkat sedang berlangsung



Gambar 5. 7 Hasil Pengujian PA-07

5.2.6. Pengujian Pencatatan Panggilan Telepon Keluar

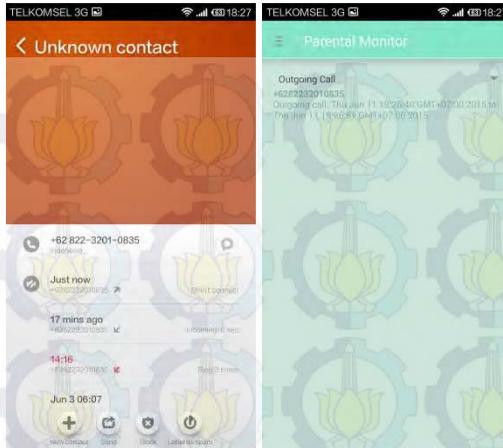
Pengujian pencatatan panggilan telepon keluar dilakukan untuk memeriksa apakah aplikasi dapat menangkap *broadcast* terjadinya panggilan telepon keluar dengan baik. Skenario yang dilakukan untuk pengujian kali ini dijelaskan sebagai berikut.

Tabel 5. 9 Rincian Pengujian Pencatatan Panggilan Keluar

Nomor	PA-08
Nama	Melakukan pencatatan panggilan keluar

Tujuan	Memeriksa fungsi untuk mencatat panggilan telepon yang dilakukan oleh perangkat
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Pengguna menghubungi salah satu nomor telepon 2. Pengguna menutup telepon 3. Pengguna membuka aplikasi pemantau 4. Pengguna melakukan proses <i>login</i> 5. Pengguna memilih <i>outgoing call</i> pada <i>spinner</i> di halaman <i>overview</i>
Masukan	Alamat <i>email</i> dan <i>password</i> untuk <i>login</i>
Keluaran yang diharapkan	Halaman <i>overview</i> ditampilkan, terdapat data panggilan telepon yang baru saja dilakukan
Hasil pengujian	Berhasil

Berikut tampilan antarmuka saat proses pengujian pencatatan panggilan telepon keluar.



Gambar 5. 8 Hasil Pengujian PA-08

5.2.7. Pengujian Pencatatan Alamat Situs

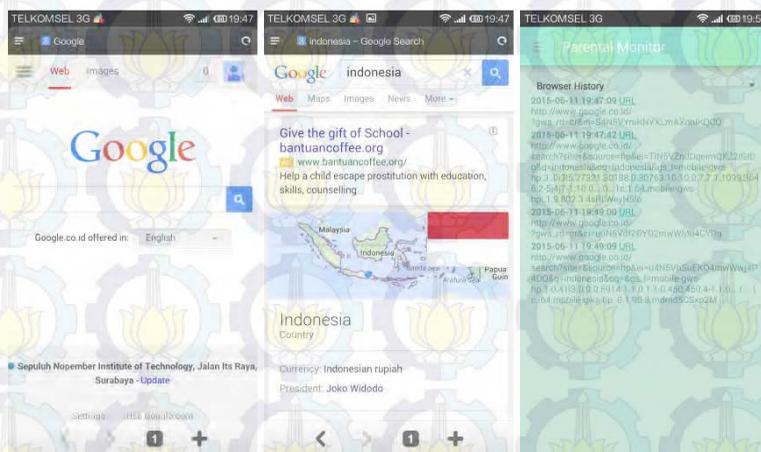
Pengujian ini dilakukan untuk memeriksa apakah servis yang dinyalakan setiap pengguna membuka aplikasi *browser native* perangkat dapat mencatat alamat situs dengan benar. Skenario pengujian dijelaskan pada tabel berikut ini.

Tabel 5. 10 Rincian Pengujian Pencatatan Alamat Situs

Nomor	PA-09
Nama	Melakukan pencatatan alamat situs yang dikunjungi
Tujuan	Memeriksa fungsi untuk mencatat alamat situs yang dikunjungi oleh perangkat
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi <i>browser native</i> pada perangkat 2. Pengguna mengunjungi beberapa situs di internet 3. Pengguna menutup aplikasi <i>browser</i>

	<ol style="list-style-type: none"> 4. Pengguna membuka aplikasi pemantau 5. Pengguna melakukan proses <i>login</i> 6. Pengguna memilih <i>browser history</i> pada <i>spinner</i> di halaman <i>overview</i>
Masukan	Alamat <i>email</i> dan <i>password</i> untuk <i>login</i>
Keluaran yang diharapkan	Halaman <i>overview</i> ditampilkan, terdapat data alamat situs beserta judul situs yang dikunjungi perangkat
Hasil pengujian	Berhasil

Pada pengujian ini, halaman situs masih dalam tahap *loading* setelah satu detik, sehingga judul situs yang tersimpan adalah alamat situsnya. Namun, alamat situs yang tercatat tetap sesuai dengan harapan. Berikut tampilan antarmuka proses pengujian ini.



Gambar 5. 9 Hasil Pengujian PA-09

5.2.8. Pengujian Pencatatan Lokasi

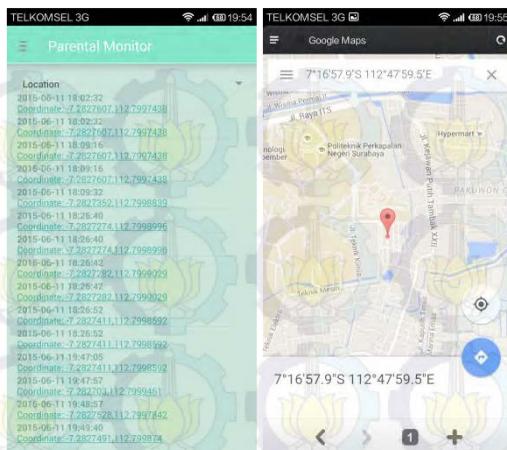
Pengujian pencatatan lokasi dilakukan untuk memeriksa apakah servis yang digunakan untuk mencari posisi koordinat perangkat dapat bekerja dengan baik. Oleh karena proses pencatatan lokasi dilakukan setiap pengguna membuka aplikasi

pada perangkat, skenario pengujian pencatatan lokasi hampir sama dengan skenario pencatatan data aplikasi yang dibuka oleh perangkat. Berikut penjelasan skenario pengujian ini.

Tabel 5. 11 Rincian Pengujian Melakukan Pencatatan Lokasi

Nomor	PA-10
Nama	Melakukan pencatatan lokasi
Tujuan	Memeriksa fungsi untuk mencatat koordinat lokasi perangkat bekerja dengan baik
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi apa saja selain aplikasi pemantau 2. Pengguna menutup aplikasi yang baru saja dibuka 3. Pengguna membuka aplikasi pemantau 4. Pengguna melakukan proses <i>login</i> 5. Pengguna memilih <i>location</i> pada <i>spinner</i> di halaman <i>overview</i> 6. Pengguna menekan koordinat lokasi
Masukan	Alamat <i>email</i> dan <i>password</i> untuk <i>login</i>
Keluaran yang diharapkan	Halaman peta google ditampilkan dengan <i>pinpoint</i> posisi koordinat
Hasil pengujian	Berhasil

Pada pengujian kali ini dilakukan di Jalan Teknik Komputer U-42, Perumahan Dosen ITS Surabaya. Hasil pengujian menunjukkan bahwa koordinat yang dicatat oleh aplikasi pemantau merepresentasikan posisi perangkat dengan cukup baik. Berikut tampilan antarmuka hasil pengujian ini.



Gambar 5. 10 Hasil Pengujian PA-10

5.2.9. Pengujian Menampilkan Statistik

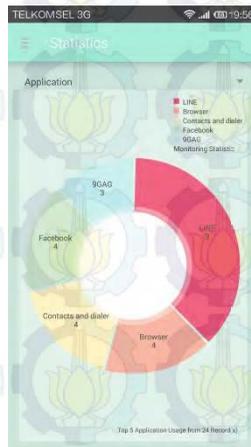
Pengujian ini dilakukan untuk mengetahui apakah *library* MPAndroidChart telah berhasil menampilkan statistik pemakaian perangkat bergerak dengan benar. Pada pengujian kali ini, terdapat lima skenario pengujian. Skenario pertama adalah pengujian untuk menampilkan statistik lima aplikasi yang paling sering dibuka oleh pengguna. Berikut penjelasan dari skenario ini.

Tabel 5. 12 Rincian Pengujian Menampilkan Statistik Aplikasi

Nomor	PA-11
Nama	Menampilkan statistik aplikasi
Tujuan	Memeriksa fungsi untuk menampilkan statistik aplikasi pada perangkat berjalan dengan baik
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna memilih <i>statistics</i> pada

	navigasi aplikasi
Masukan	Alamat <i>email</i> dan <i>password</i> untuk <i>login</i>
Keluaran yang diharapkan	Halaman <i>statistics</i> dengan diagram pai yang menampilkan data lima aplikasi yang paling sering dibuka oleh perangkat
Hasil pengujian	Berhasil

Berikut tampilan antarmuka hasil pengujian statistik aplikasi pada aplikasi pemantau.



Gambar 5. 11 Hasil Pengujian PA-11

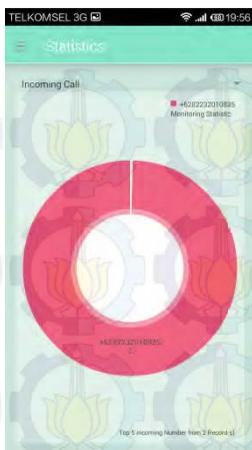
Skenario kedua adalah pengujian untuk menampilkan statistik lima nomor telepon yang paling sering menghubungi perangkat. Berikut penjelasan dari skenario ini.

Tabel 5. 13 Rincian Pengujian Menampilkan Statistik Panggilan Masuk

Nomor	PA-12
Nama	Menampilkan statistik panggilan telepon masuk
Tujuan	Memeriksa fungsi untuk menampilkan

	statistik telepon yang masuk ke perangkat berjalan dengan baik
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna memilih <i>statistics</i> pada navigasi aplikasi 4. Pengguna memilih <i>incoming call</i> pada <i>spinner</i> di halaman <i>statistics</i>
Masukan	Alamat <i>email</i> dan <i>password</i> untuk <i>login</i>
Keluaran yang diharapkan	Halaman <i>statistics</i> dengan diagram pai yang menampilkan data lima nomor telepon yang paling sering menghubungi perangkat
Hasil pengujian	Berhasil

Berikut *screenshot* antarmuka halaman *statistics* yang menampilkan data nomor telepon yang paling sering menghubungi perangkat.



Gambar 5. 12 Hasil Pengujian PA-12

Skenario ketiga adalah pengujian untuk menampilkan statistik lima nomor telepon yang paling sering dihubungi oleh perangkat. Berikut penjelasan dari skenario ini.

Tabel 5. 14 Rincian Pengujian Menampilkan Statistik Panggilan Keluar

Nomor	PA-13
Nama	Menampilkan statistik panggilan telepon keluar
Tujuan	Memeriksa fungsi untuk menampilkan statistik panggilan telepon yang dilakukan oleh perangkat dapat berjalan dengan baik
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna memilih <i>statistics</i> pada navigasi aplikasi 4. Pengguna memilih <i>outgoing call</i> pada

	<i>spinner</i> di halaman <i>statistics</i>
Masukan	Alamat <i>email</i> dan <i>password</i> untuk <i>login</i>
Keluaran yang diharapkan	Halaman <i>statistics</i> dengan diagram pai yang menampilkan data lima nomor telepon yang paling sering dihubungi oleh perangkat
Hasil pengujian	Berhasil

Berikut *screenshot* antarmuka halaman *statistics* yang menampilkan data nomor telepon yang paling sering dihubungi oleh perangkat.



Gambar 5. 13 Hasil Pengujian PA-13

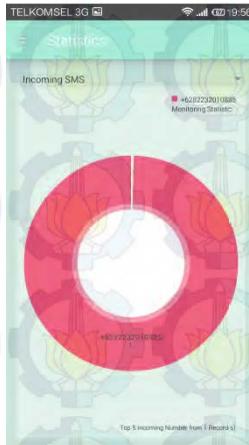
Skenario keempat adalah pengujian untuk menampilkan statistik lima nomor telepon yang paling sering mengirim pesan singkat ke perangkat. Berikut penjelasan dari skenario ini.

Tabel 5. 15 Rincian Pengujian Menampilkan Statistik SMS Masuk

Nomor	PA-14
Nama	Menampilkan statistik pesan singkat masuk
Tujuan	Memeriksa fungsi untuk menampilkan statistik pesan singkat yang masuk ke

	perangkat berjalan dengan baik
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna memilih <i>statistics</i> pada navigasi aplikasi 4. Pengguna memilih <i>incoming sms</i> pada <i>spinner</i> di halaman <i>statistics</i>
Masukan	Alamat <i>email</i> dan <i>password</i> untuk <i>login</i>
Keluaran yang diharapkan	Halaman <i>statistics</i> dengan diagram pai yang menampilkan data lima nomor telepon yang paling sering mengirim pesan singkat ke perangkat
Hasil pengujian	Berhasil

Berikut *screenshot* antarmuka halaman *statistics* yang menampilkan hasil pengujian dengan skenario di atas.



Gambar 5. 14 Hasil Pengujian PA-14

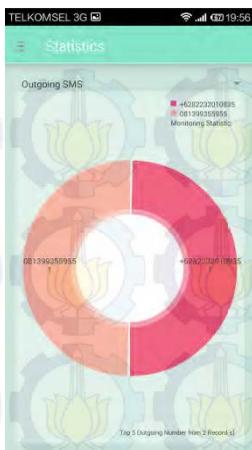
Skenario terakhir adalah pengujian untuk menampilkan statistik lima nomor telepon yang paling sering menjadi tujuan

pengiriman pesan singkat oleh perangkat. Berikut penjelasan dari skenario ini.

Tabel 5. 16 Rincian Pengujian Menampilkan Statistik SMS Keluar

Nomor	PA-15
Nama	Menampilkan statistik pesan singkat keluar
Tujuan	Memeriksa fungsi untuk menampilkan statistik pesan singkat yang dikirim oleh perangkat dapat berjalan dengan baik
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna memilih <i>statistics</i> pada navigasi aplikasi 4. Pengguna memilih <i>outgoing sms</i> pada <i>spinner</i> di halaman <i>statistics</i>
Masukan	Alamat <i>email</i> dan <i>password</i> untuk <i>login</i>
Keluaran yang diharapkan	Halaman <i>statistics</i> dengan diagram pai yang menampilkan data lima nomor telepon yang paling sering menerima pesan singkat dari perangkat
Hasil pengujian	Berhasil

Berikut tampilan antarmuka hasil pengujian dengan skenario di atas.



Gambar 5. 15 Hasil Pengujian PA-15

5.2.10. Pengujian Ubah Akun

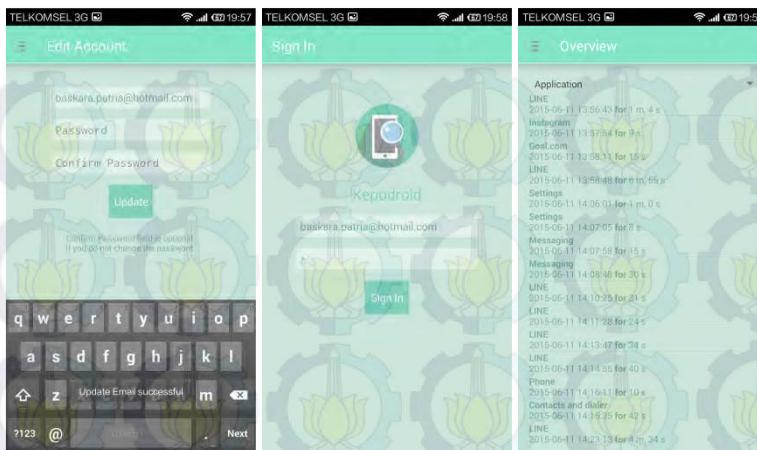
Pengujian ini dilakukan untuk memeriksa apakah fungsi yang menangani proses ubah akun dapat berjalan dengan baik atau tidak. Ada dua skenario pengujian yang dilakukan. Pertama, pengguna hanya mengubah alamat *email* saja tanpa mengubah *password*. Kedua, pengguna mengubah *password* saja tanpa mengubah alamat *email*. Ketiga, pengguna mengubah alamat *email* beserta *password*. Untuk memeriksa apakah aplikasi sudah menangani proses ini dengan baik, pengguna akan menggunakan alamat *email* atau *password* barunya untuk melakukan proses *login* ulang. Berikut penjelasan dari skenario pertama.

Tabel 5. 17 Rincian Pengujian Mengubah Alamat Email

Nomor	PA-16
Nama	Mengubah alamat <i>email</i>
Tujuan	Memeriksa fungsi untuk mengubah alamat <i>email</i> dapat berjalan dengan baik
Kondisi awal	Aplikasi pemantau berjalan di latar belakang

Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna memilih <i>edit account</i> pada navigasi aplikasi 4. Pengguna memasukkan alamat <i>email</i> baru 5. Pengguna menekan tombol <i>update</i> 6. Pengguna menutup aplikasi pemantau 7. Pengguna membuka kembali aplikasi pemantau 8. Pengguna melakukan proses <i>login</i> dengan alamat <i>email</i> yang baru
Masukan	<ol style="list-style-type: none"> 1. Alamat <i>email</i> dan <i>password</i> untuk <i>login</i> pertama 2. Alamat <i>email</i> pengganti 3. Alamat <i>email</i> dan <i>password</i> untuk <i>login</i> kedua
Keluaran yang diharapkan	Halaman <i>overview</i>
Hasil pengujian	Berhasil

Berikut *screenshot* pengujian dengan skenario di atas.



Gambar 5. 16 Hasil Pengujian PA-16

Berikut penjelasan skenario kedua dimana pengujian dilakukan dengan mengubah *password* akun.

Tabel 5. 18 Rincian Pengujian Mengubah Password

Nomor	PA-17
Nama	Mengubah <i>password</i>
Tujuan	Memeriksa fungsi untuk mengubah <i>password</i> dapat berjalan dengan baik
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna memilih <i>edit account</i> pada navigasi aplikasi 4. Pengguna memasukkan <i>password</i> baru 5. Pengguna menekan tombol <i>update</i> 6. Pengguna menutup aplikasi pemantau 7. Pengguna membuka kembali aplikasi pemantau

	8. Pengguna melakukan proses <i>login</i> dengan alamat <i>email</i> yang baru
Masukan	<ol style="list-style-type: none"> 1. Alamat <i>email</i> dan <i>password</i> untuk <i>login</i> pertama 2. <i>Password</i> pengganti 3. Alamat <i>email</i> dan <i>password</i> untuk <i>login</i> kedua
Keluaran yang diharapkan	Halaman <i>overview</i>
Hasil pengujian	Berhasil

Berikut *screenshot* hasil pengujian di atas.



Gambar 5. 17 Hasil Pengujian PA-17

Berikut penjelasan skenario terakhir dimana pengujian dilakukan dengan mengubah alamat *email* beserta *password* akun.

Tabel 5. 19 Rincian Pengujian Mengubah Alamat Email dan Password

Nomor	PA-18
Nama	Mengubah alamat <i>email</i> dan <i>password</i>

Tujuan	Memeriksa fungsi untuk mengubah alamat <i>email</i> beserta <i>password</i> dapat berjalan dengan baik
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna memilih <i>edit account</i> pada navigasi aplikasi 4. Pengguna memasukkan alamat <i>email</i> dan <i>password</i> baru 5. Pengguna menekan tombol <i>update</i> 6. Pengguna menutup aplikasi pemantau 7. Pengguna membuka kembali aplikasi pemantau 8. Pengguna melakukan proses <i>login</i> dengan alamat <i>email</i> yang baru
Masukan	<ol style="list-style-type: none"> 1. Alamat <i>email</i> dan <i>password</i> untuk <i>login</i> pertama 2. Alamat <i>email</i> dan <i>password</i> pengganti 3. Alamat <i>email</i> dan <i>password</i> untuk <i>login</i> kedua
Keluaran yang diharapkan	Halaman <i>overview</i>
Hasil pengujian	Berhasil

Berikut *screenshot* hasil pengujian di atas.



Gambar 5. 18 Hasil Pengujian PA-18

5.2.11. Pengujian Atur Jadwal dan Pengiriman *Record*

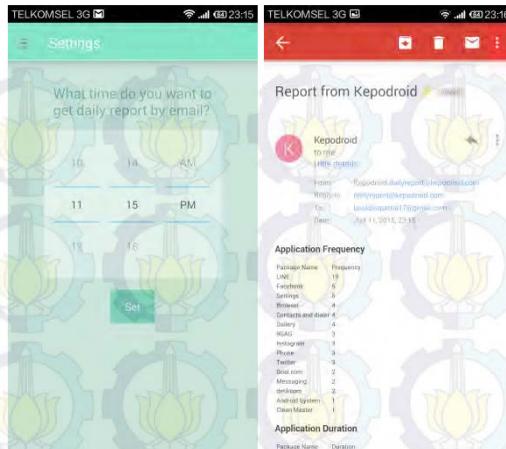
Pengujian ini dilakukan untuk memeriksa apakah fungsi pengaturan *alarm* pada aplikasi dapat berjalan dengan baik. Hal ini ditandai dengan terkirimnya *record* ke *webservice*. Oleh karena *webservice* bersifat hanya meneruskan data yang dikirim ke dirinya, maka untuk memeriksa apakah pengujian kali ini berhasil maka indikatornya adalah *record* terkirim ke alamat *email* tujuan. Skenario pengujian pengiriman *record* kali ini dibagi menjadi tiga jenis. Skenario pertama adalah pengujian pengaturan jadwal pengiriman *record* ketika pada jadwal yang ditentukan, perangkat sedang memiliki koneksi ke internet. Berikut penjelasan dari skenario pertama.

Tabel 5. 20 Rincian Pengujian Pengiriman Record Saat Terhubung Internet

Nomor	PA-19
Nama	Mengirimkan <i>record</i> ketika perangkat terhubung dengan internet
Tujuan	Memeriksa fungsi untuk mengatur jadwal

	pengiriman <i>record</i> dan fungsi untuk mengirim <i>record</i>
Kondisi awal	Aplikasi pemantau berjalan di latar belakang
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna memilih <i>setting</i> pada navigasi aplikasi 4. Pengguna memasukkan waktu yang ditentukan 5. Pengguna menekan tombol <i>set</i>
Masukan	<ol style="list-style-type: none"> 1. Alamat <i>email</i> dan <i>password</i> untuk <i>login</i> 2. Jam dan menit pengiriman <i>record</i>
Keluaran yang diharapkan	<i>Email</i> terkirim ke akun <i>email</i> tujuan sesuai jadwal
Hasil pengujian	Berhasil

Pada skenario di atas, *record* terkirim ke alamat *email* tujuan dengan benar. Berikut *screenshot* saat pengaturan waktu pengiriman dan saat *email* masuk.



Gambar 5. 19 Hasil Pengujian PA-19

Skenario kedua adalah pengujian pengaturan jadwal pengiriman *record* ketika pada saat jadwal yang ditentukan, koneksi perangkat ke internet sedang dimatikan. Hal ini dilakukan untuk memeriksa apakah fungsi mengaktifkan dan menonaktifkan koneksi internet menggunakan paket *mobile* data bekerja dengan baik. Berikut skenario pengujian kali ini.

Tabel 5. 21 Rincian Pengujian Pengiriman Record Saat Tidak Terhubung Internet

Nomor	PA-20
Nama	Mengirimkan <i>record</i> ketika perangkat tidak terhubung dengan internet
Tujuan	Memeriksa fungsi untuk mengatur jadwal pengiriman <i>record</i> dan fungsi untuk mengirim <i>record</i> ketika perangkat sedang tidak terhubung dengan internet
Kondisi awal	Aplikasi pemantau berjalan di latar belakang dan perangkat tidak sedang terhubung dengan internet

Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna memilih <i>setting</i> pada navigasi aplikasi 4. Pengguna memasukkan waktu yang ditentukan 5. Pengguna menekan tombol <i>set</i>
Masukan	<ol style="list-style-type: none"> 1. Alamat <i>email</i> dan <i>password</i> untuk <i>login</i> 2. Jam dan menit pengiriman <i>record</i>
Keluaran yang diharapkan	<i>Email</i> terkirim ke akun <i>email</i> tujuan sesuai jadwal, koneksi internet terputus
Hasil pengujian	Berhasil

Skenario terakhir adalah pengujian pengaturan jadwal pengiriman *report* ketika pada saat jadwal yang ditentukan, perangkat sedang dalam mode pesawat terbang. Hasil yang diharapkan adalah aplikasi tidak mengirimkan *report* dan menundanya. Berikut penjelasan skenario pengujian ini.

Tabel 5. 22 Rincian Pengujian Pengiriman Record Saat Mode Pesawat Terbang

Nomor	PA-21
Nama	Mengirimkan <i>record</i> ketika perangkat sedang dalam mode pesawat terbang
Tujuan	Memeriksa fungsi untuk mengatur jadwal pengiriman <i>record</i> dan fungsi untuk mengirim <i>record</i> ketika perangkat sedang dalam mode pesawat terbang
Kondisi awal	Aplikasi pemantau berjalan di latar belakang dan perangkat sedang dalam mode pesawat terbang
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna memilih <i>setting</i> pada navigasi aplikasi 4. Pengguna memasukkan waktu yang

	ditentukan 5. Pengguna menekan tombol <i>set</i>
Masukan	1. Alamat <i>email</i> dan <i>password</i> untuk <i>login</i> 2. Jam dan menit pengiriman <i>record</i>
Keluaran yang diharapkan	Perangkat tetap dalam mode pesawat terbang, <i>report</i> tidak dikirimkan
Hasil pengujian	Berhasil

Pada skenario terakhir ini, *report* tidak dikirimkan oleh aplikasi pemantau dan posisi perangkat tetap pada mode pesawat terbang. Selain itu, aplikasi pemantau juga tidak mengaktifkan koneksi internet pada perangkat. Namun setelah 30 menit dan perangkat telah terhubung dengan internet, *report* terkirim dengan normal.

5.2.12. Pengujian Penambahan Aplikasi Favorit

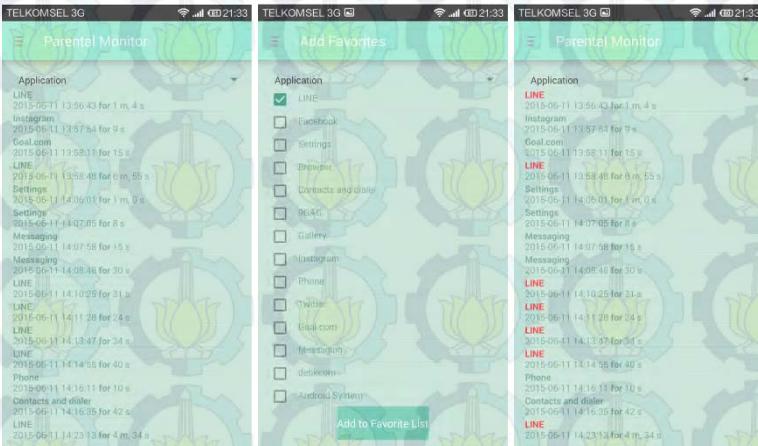
Pengujian ini dilakukan untuk memeriksa apakah fungsi penambahan aplikasi favorit bekerja dan aplikasi dapat membedakan aplikasi favorit dengan aplikasi lainnya. Hal ini ditandai ketika aplikasi yang masuk ke daftar favorit akan ditampilkan dengan warna merah pada halaman *overview*. Berikut penjelasan skenario pengujian kali ini.

Tabel 5. 23 Rincian Pengujian Menambahkan Aplikasi Favorit

Nomor	PA-22
Nama	Menambahkan aplikasi favorit
Tujuan	Memeriksa fungsi untuk menambahkan dan menampilkan aplikasi favorit
Kondisi awal	Aplikasi pemantau berjalan di latar belakang dan telah mencatat beberapa pemakaian aplikasi lain
Skenario	1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna memilih <i>add favorites</i> pada navigasi aplikasi 4. Pengguna memilih aplikasi yang

	ditentukan 5. Pengguna menekan tombol <i>add to favorites</i>
Masukan	1. Alamat <i>email</i> dan <i>password</i> untuk <i>login</i> 2. Aplikasi yang ingin dimasukkan ke daftar favorit
Keluaran yang diharapkan	Halaman <i>overview</i> menampilkan aplikasi favorit ditulis dengan warna merah
Hasil pengujian	Berhasil

Berikut *screenshot* proses pengujian skenario di atas.



Gambar 5. 20 Hasil Pengujian PA-22

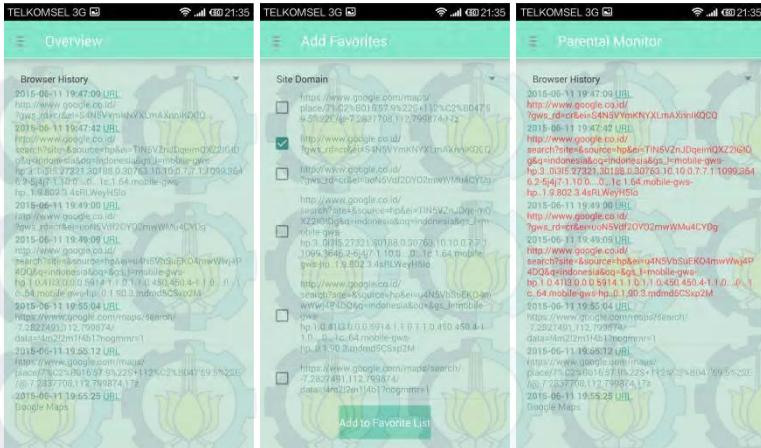
5.2.13. Pengujian Penambahan *Domain* Favorit

Pengujian ini dilakukan untuk memeriksa apakah fungsi penambahan *domain* favorit bekerja dan aplikasi dapat membedakan situs dengan *domain* favorit dan situs dengan *domain* lainnya. Hal ini ditandai ketika situs yang memiliki *domain* yang masuk ke daftar favorit akan ditampilkan dengan warna merah pada halaman *overview*. Berikut penjelasan skenario pengujian kali ini.

Tabel 5. 24 Rincian Pengujian Menambahkan Domain Favorit

Nomor	PA-23
Nama	Menambahkan <i>domain</i> favorit
Tujuan	Memeriksa fungsi untuk menambahkan dan menampilkan situs dengan <i>domain</i> favorit
Kondisi awal	Aplikasi pemantau berjalan di latar belakang dan telah mencatat beberapa kunjungan ke beberapa situs dengan <i>domain</i> berbeda-beda
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna memilih <i>add favorites</i> pada navigasi aplikasi 4. Pengguna memilih <i>site domain</i> pada <i>spinner</i> 5. Pengguna memilih situs yang ditentukan 6. Pengguna menekan tombol <i>add to favorites</i> 7. Pengguna memilih <i>browser history</i> pada <i>spinner</i> di halaman <i>overview</i>
Masukan	<ol style="list-style-type: none"> 1. Alamat <i>email</i> dan <i>password</i> untuk <i>login</i> 2. Situs dengan <i>domain</i> yang ingin dimasukkan ke daftar favorit
Keluaran yang diharapkan	Halaman <i>overview</i> menampilkan situs dengan <i>domain</i> favorit ditulis dengan warna merah
Hasil pengujian	Berhasil

Berikut *screenshot* proses pengujian dengan skenario di atas.



Gambar 5. 21 Hasil Pengujian PA-23

5.2.14. Pengujian Penghapusan Aplikasi Favorit

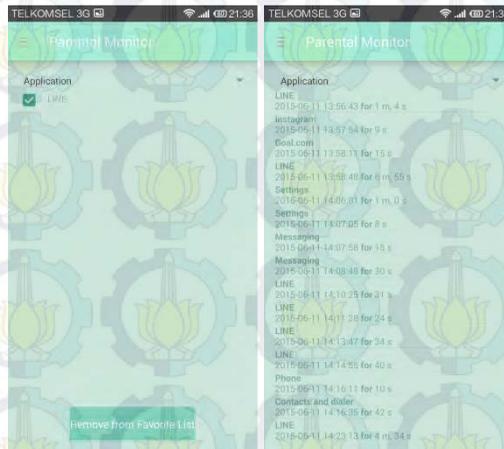
Pengujian ini dilakukan untuk memeriksa apakah fungsi penghapusan aplikasi dari daftar favorit bekerja dengan baik. Hal ini ditandai ketika aplikasi yang sebelumnya termasuk ke dalam daftar aplikasi favorit dan ditampilkan dengan warna merah pada halaman *overview*, kembali ditampilkan dengan warna *default*. Berikut penjelasan dari skenario pengujian kali ini.

Tabel 5. 25 Rincian Pengujian Menghapus Aplikasi Favorit

Nomor	PA-24
Nama	Menghapus aplikasi favorit
Tujuan	Memeriksa fungsi untuk menghapus aplikasi favorit
Kondisi awal	Aplikasi pemantau berjalan di latar belakang dan telah memiliki beberapa aplikasi yang termasuk ke dalam daftar favorit
Skenario	1. pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i>

	<ol style="list-style-type: none"> 3. Pengguna memilih <i>remove favorites</i> pada navigasi aplikasi 4. Pengguna memilih aplikasi yang ditentukan 5. Pengguna menekan tombol <i>remove from favorites</i>
Masukan	<ol style="list-style-type: none"> 1. Alamat <i>email</i> dan <i>password</i> untuk <i>login</i> 2. Aplikasi yang ingin dihilangkan dari daftar favorit
Keluaran yang diharapkan	Halaman <i>overview</i> menampilkan aplikasi yang baru saja dihilangkan dari daftar favorit ditulis dengan warna <i>default</i>
Hasil pengujian	Berhasil

Berikut *screenshot* proses pengujian dengan skenario di atas.



Gambar 5. 22 Hasil Pengujian PA-24

5.2.15. Pengujian Penghapusan *Domain* Favorit

Pengujian ini dilakukan untuk memeriksa apakah fungsi penghapusan *domain* dari daftar favorit bekerja dengan baik. Hal ini ditandai ketika situs-situs dengan *domain* yang sebelumnya termasuk ke dalam daftar *domain* favorit dan ditampilkan dengan

warna merah pada halaman *overview*, kembali ditampilkan dengan warna *default*. Berikut penjelasan dari skenario pengujian kali ini.

Tabel 5. 26 Rincian Pengujian Menghapus Domain Favorit

Nomor	PA-25
Nama	Menghapus <i>domain</i> favorit
Tujuan	Memeriksa fungsi untuk menghapus <i>domain</i> favorit
Kondisi awal	Aplikasi pemantau berjalan di latar belakang dan telah memiliki beberapa <i>domain</i> yang termasuk ke dalam daftar favorit
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna memilih <i>remove favorites</i> pada navigasi aplikasi 4. Pengguna memilih <i>site domain</i> pada <i>spinner</i> 5. Pengguna memilih <i>domain</i> yang akan dihilangkan 6. Pengguna menekan tombol <i>remove from favorites</i> 7. Pengguna memilih <i>browser history</i> pada <i>spinner</i> di halaman <i>overview</i>
Masukan	<ol style="list-style-type: none"> 1. Alamat <i>email</i> dan <i>password</i> untuk <i>login</i> 2. <i>Domain</i> yang ingin dihilangkan dari daftar favorit
Keluaran yang diharapkan	Halaman <i>overview</i> menampilkan situs dengan <i>domain</i> yang baru saja dihilangkan dari daftar favorit ditulis dengan warna <i>default</i>
Hasil pengujian	Berhasil

Berikut *screenshot* proses pengujian dengan skenario di atas.



Gambar 5. 23 Hasil Pengujian PA-25

5.3. Pengujian Non Fungsionalitas

Pada subbab ini dibahas pengujian non fungsionalitas yang dilakukan pada aplikasi pemantau pada perangkat bergerak. Pada tahap ini aplikasi akan dilihat perilakunya ketika pengguna melakukan pembersihan memori melalui *task manager*, pembersihan memori melalui aplikasi *third party*, menghentikan servis melalui *application manager*, dan memantau konsumsi memori oleh aplikasi.

5.3.1. Pembersihan Memori Melalui *Task Manager*

Pengujian kali ini dilakukan untuk mengetahui perilaku aplikasi ketika pengguna memutuskan untuk melakukan pembersihan memori melalui *task manager*. Normalnya, aplikasi akan tertutup dan servis-servis yang berjalan pada pada latar belakang perangkat akan berhenti berjalan. Android sebenarnya memberikan fasilitas *START_STICKY*¹¹ yang dapat ditambahkan pada kode sumber servis yang dibuat. Menurut dokumentasi dari situs *developer.android.com* fitur ini mengakibatkan sistem

¹¹ http://developer.android.com/reference/android/app/Service.html#START_STICKY

menyalakan kembali servis yang telah dimatikan oleh sistem ketika selesai melakukan pembersihan memori. Namun ketika dicoba untuk diimplementasikan, hasilnya negatif. Oleh sebab itu dilakukan pengujian terhadap metode *broadcast* yang diimplementasikan pada aplikasi pemantau kali ini.

Skenario yang dilakukan dimulai dengan menyalakan servis aplikasi pemantau supaya berjalan pada latar belakang. Setelah itu, pengguna melakukan pembersihan memori melalui *task manager*. Hasil yang diharapkan adalah servis pada aplikasi berjalan kembali. Berikut penjelasan skenario pengujian ini.

Tabel 5. 27 Rincian Pengujian Pembersihan Memori Melalui Task Manager

Nomor	PB-1
Nama	Membersihkan memori melalui <i>task manager</i>
Tujuan	Memeriksa fungsi untuk menyalakan servis kembali setelah dihentikan oleh sistem melalui <i>task manager</i>
Kondisi awal	Aplikasi pemantau berjalan di latar belakang dan <i>permission</i> untuk <i>auto start</i> telah diaktifkan
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna menutup aplikasi pemantau 4. Pengguna memeriksa adanya aplikasi pemantau pada daftar <i>task manager</i> 5. Pengguna menekan <i>clear memory</i> 6. Pengguna membuka menu <i>application manager</i> pada menu <i>setting</i> di perangkat 7. Pengguna memilih aplikasi pemantau pada menu <i>running application</i>
Masukan	1. Alamat <i>email</i> dan <i>password</i> untuk <i>login</i>
Keluaran yang	Servis yang terdapat pada aplikasi pemantau

diharapkan	berjalan
Hasil pengujian	Berhasil

Berikut tampilan layar ketika pengguna akan melakukan proses *clear memory* dari *task manager*.



Gambar 5. 24 Proses Pembersihan Memori Dari Task Manager

Setelah pengguna menekan tombol bergambar kuas, maka muncul notifikasi pada layar yang menunjukkan ada *foreground service* yang baru saja berjalan.



Gambar 5. 25 Notifikasi Foreground Service

Ketika notifikasi dibuka, muncul logo aplikasi pemantau yang menandakan bahwa servis milik aplikasi ini tengah berjalan di latar belakang.



Gambar 5. 26 Tampilan Notifikasi Foreground Service Berjalan

Meskipun pada kenyataannya servis milik aplikasi pemantau dinyalakan ulang, ternyata pada menu *application manager*, terlihat durasi servis milik aplikasi pemantau yang tidak di-*reset*.



Gambar 5. 27 Tampilan Menu Application Manager

5.3.2. Pembersihan Memori Melalui Aplikasi Luar

Pengujian ini dilakukan untuk memeriksa perilaku aplikasi pemantau ketika pengguna melakukan pembersihan memori menggunakan aplikasi luar. Pada kasus ini, dilakukan pengujian menggunakan aplikasi *Clean Master*. Berikut skenario pengujian kali ini.

Tabel 5. 28 Rincian Pengujian Pembersihan Memori Melalui Aplikasi Luar

Nomor	PB-2
Nama	Membersihkan memori melalui aplikasi luar
Tujuan	Memeriksa fungsi untuk menyalakan servis kembali setelah dihentikan oleh sistem melalui aplikasi luar
Kondisi awal	Aplikasi pemantau berjalan di latar belakang dan <i>permission</i> untuk <i>auto start</i> telah diaktifkan
Skenario	1. Pengguna membuka aplikasi pembersih memori

	<ol style="list-style-type: none"> 2. Pengguna memeriksa adanya aplikasi pemantau pada daftar yang akan dibersihkan 3. Pengguna menekan <i>boost</i> 4. Pengguna menutup aplikasi pembersih memori 5. Pengguna membuka menu <i>application manager</i> pada menu <i>setting</i> di perangkat 6. Pengguna memilih aplikasi pemantau pada menu <i>running application</i>
Masukan	-
Keluaran yang diharapkan	Servis yang terdapat pada aplikasi pemantau berjalan
Hasil pengujian	Berhasil

Berikut *screenshot* proses pengujian pembersihan memori melalui aplikasi luar.



Gambar 5. 28 Tampilan Clear Master

Pada tampilan di atas menunjukkan bahwa terdapat memori sebesar 25 MB yang bisa dibersihkan yang berasal dari pemakaian aplikasi pemantau. Berikut tampilan notifikasi

foreground service aplikasi pemantau setelah proses pembersihan memori yang dilakukan oleh *clean master* selesai.



Gambar 5. 29 Notifikasi *Foreground Service*

Berikut tampilan daftar servis yang dijalankan oleh aplikasi pemantau melalui menu *application manager*.



Gambar 5. 30 Tampilan Application Manager

Pada pengujian kali ini membuktikan bahwa servis ternyata tidak dihentikan oleh aplikasi *clear master* meskipun pada keterangannya terdapat sekitar 25 MB memori yang dibersihkan berasal dari pemakaian aplikasi pemantau.

5.3.3. Penghentian Servis Melalui *Application Manager*

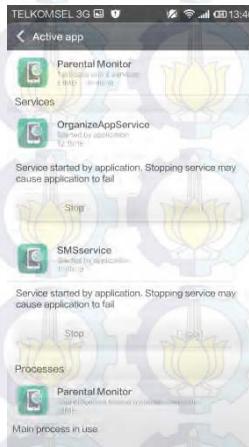
Pengujian ini dilakukan untuk memeriksa perilaku aplikasi pemantau ketika pengguna menghentikan servis yang berjalan pada aplikasi pemantau. Pada kasus ini, dilakukan penghentian melalui menu *application manager* yang ada pada menu *settings* milik perangkat. Berikut skenario pengujian kali ini.

Tabel 5. 29 Rincian Pengujian Penghentian Servis

Nomor	PB-3
Nama	Menghentikan servis melalui <i>application manager</i>
Tujuan	Memeriksa fungsi untuk menyalakan servis kembali setelah dihentikan oleh pengguna
Kondisi awal	Aplikasi pemantau berjalan di latar belakang dan <i>permission</i> untuk <i>auto start</i>

	telah diaktifkan
Skenario	<ol style="list-style-type: none"> 1. Pengguna membuka aplikasi pemantau 2. Pengguna melakukan proses <i>login</i> 3. Pengguna menutup aplikasi pemantau 4. Pengguna membuka menu <i>application manager</i> pada menu <i>setting</i> di perangkat 5. Pengguna memilih aplikasi pemantau pada menu <i>running application</i> 6. Pengguna menekan tombol <i>stop</i> pada servis milik aplikasi pemantau
Masukan	-
Keluaran yang diharapkan	Servis yang terdapat pada aplikasi pemantau berjalan kembali
Hasil pengujian	Berhasil

Berikut tampilan menu *application manager* ketika servis milik aplikasi pemantau belum dihentikan. Di situ terlihat bahwa durasi servis ini telah berjalan selama beberapa jam.



Gambar 5. 31 Tampilan Saat Akan Dihentikan

Berikut tampilan menu *application manager* ketika pengguna baru saja menghentikan kedua servis milik aplikasi

pemantau. Terlihat bahwa durasi servis berjalan menandakan bahwa servis ini baru saja di-*restart* ulang.



Gambar 5. 32 Tampilan Setelah Dihentikan

5.3.4. Konsumsi Memori

Pada subbab ini ditampilkan *screenshot* konsumsi memori aplikasi pemantau dari waktu ke waktu. Berikut *screenshot* konsumsi memori aplikasi pemantau pada saat aplikasi di latar belakang berjalan kurang dari 1 jam.



Gambar 5. 33 Konsumsi Memori < 1 Jam

Pada gambar di atas, angka menunjukkan bahwa aplikasi memakan kapasitas memori sebesar 37 MB. Selanjutnya, konsumsi memori oleh aplikasi terpantau sedikit menurun. Berikut *screenshot* yang menunjukkan angka konsumsi memori oleh aplikasi setelah berjalan selama sekitar 2 jam.



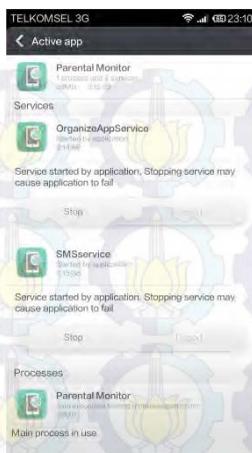
Gambar 5. 34 Konsumsi Memori 2 Jam

Pada gambar di atas, angka konsumsi memori menunjukkan adanya penurunan konsumsi memori menjadi 34 MB. Berikutnya, dilakukan pemeriksaan konsumsi memori setelah aplikasi pemantau berjalan sekitar 7 jam. Berikut *screenshot* hasil pemeriksaan tersebut.



Gambar 5. 35 Konsumsi Memori 7 Jam

Gambar di atas menunjukkan bahwa konsumsi memori oleh aplikasi pemantau sebesar 58 MB. Pada proses pemeriksaan berikutnya, aplikasi terlihat telah berjalan selama hampir 10 jam. Pada waktu ini, konsumsi memori yang dilakukan oleh aplikasi pemantau tercatat sebesar 26 MB. Berikut *screenshot* hasil pemantauan konsumsi memori aplikasi pemantau.



Gambar 5. 36 Konsumsi Memori 9 Jam

5.4. Evaluasi Pengujian

Berdasarkan hasil pengujian fungsionalitas dan pengujian non fungsionalitas, maka dapat dilakukan evaluasi sebagai berikut.

5.4.1. Evaluasi Pengujian Fungsionalitas

Pengujian fungsionalitas yang telah dilakukan memberikan hasil yang sesuai dengan skenario yang telah direncanakan. Evaluasi dari pengujian pada masing-masing fungsionalitas dijelaskan dengan sebagai berikut.

1. Pengujian PA-01 mengeluarkan hasil sesuai harapan dimana proses pendaftaran berjalan dengan sempurna.
2. Pengujian PA-02 mengeluarkan hasil sesuai harapan dimana proses *login* berjalan dengan sempurna.
3. Pengujian PA-03 mengeluarkan hasil sesuai harapan dimana proses pencatatan pemakaian aplikasi ditampilkan dengan benar.

4. Pengujian PA-04 mengeluarkan hasil sesuai harapan dimana pencatatan data pesan singkat yang masuk ditampilkan dengan benar.
5. Pengujian PA-05 mengeluarkan hasil sesuai harapan dimana pencatatan data pesan singkat yang dikirim oleh perangkat ditampilkan dengan benar.
6. Pengujian PA-06 mengeluarkan hasil sesuai harapan dimana pencatatan data panggilan telepon masuk yang diangkat oleh perangkat ditampilkan dengan benar.
7. Pengujian PA-07 mengeluarkan hasil sesuai harapan dimana pencatatan data *missed call* yang masuk ke perangkat ditampilkan dengan benar.
8. Pengujian PA-08 mengeluarkan hasil sesuai harapan dimana proses pencatatan data panggilan telepon yang dilakukan oleh perangkat ditampilkan dengan benar.
9. Pengujian PA-09 mengeluarkan hasil sesuai harapan, namun ada sedikit catatan. Proses pencatatan alamat situs dapat dilakukan dengan benar. Ketika situs dapat *ter-load* dengan sempurna dalam waktu singkat, maka judul situs yang tengah dibuka juga akan ditampilkan dengan tepat. Namun, ketika situs tidak dapat *ter-load* dalam waktu singkat, judul yang ditampilkan adalah alamat situs.
10. Pengujian PA-10 mengeluarkan hasil sesuai harapan namun ada sedikit catatan. Proses pencatatan koordinat lokasi berjalan dengan benar. Ketika perangkat tidak mampu menemukan lokasinya saat itu, maka yang disimpan adalah koordinat lokasi yang tercatat saat proses pencarian lokasi terakhir.
11. Pengujian PA-11 mengeluarkan hasil sesuai harapan dimana statistik aplikasi yang paling sering digunakan ditampilkan dengan benar.
12. Pengujian PA-12 mengeluarkan hasil sesuai harapan dimana statistik nomor telepon yang paling sering menghubungi perangkat ditampilkan dengan benar.

13. Pengujian PA-13 mengeluarkan hasil sesuai harapan dimana statistik nomor telepon yang paling sering dihubungi oleh perangkat ditampilkan dengan benar.
14. Pengujian PA-14 mengeluarkan hasil sesuai harapan dimana nomor telepon yang paling sering mengirimkan pesan singkat ke perangkat ditampilkan dengan benar.
15. Pengujian PA-15 mengeluarkan hasil sesuai harapan dimana nomor telepon yang paling sering menerima pesan singkat dari perangkat ditampilkan dengan benar.
16. Pengujian PA-16 mengeluarkan hasil sesuai harapan dimana alamat *email* berubah tetapi *password* akun tidak berubah.
17. Pengujian PA-17 mengeluarkan hasil sesuai harapan dimana alamat *email* tidak berubah tetapi *password* berubah.
18. Pengujian PA-18 mengeluarkan hasil sesuai harapan dimana alamat *email* dan *password* berubah.
19. Pengujian PA-19 mengeluarkan hasil sesuai harapan dimana *record* terkirim ke alamat *email* tujuan serta data pemakaian perangkat bergerak pada basis data dihapus oleh aplikasi.
20. Pengujian PA-20 mengeluarkan hasil sesuai harapan dimana *record* terkirim ke alamat tujuan dan koneksi internet perangkat dimatikan kembali secara otomatis oleh aplikasi serta data pemakaian perangkat bergerak pada basis data dihapus oleh aplikasi.
21. Pengujian PA-21 mengeluarkan hasil sesuai harapan dimana *record* tidak dikirimkan oleh aplikasi dan data pemakaian perangkat bergerak tidak dihapus oleh aplikasi. Proses pengiriman dilakukan 30 menit setelah jadwal yang seharusnya.
22. Pengujian PA-22 mengeluarkan hasil sesuai harapan dimana aplikasi yang baru saja ditambahkan ke daftar favorit ditampilkand dengan warna merah.

23. Pengujian PA-23 mengeluarkan hasil sesuai harapan dimana situs dengan *domain* yang baru saja dimasukkan ke daftar favorit ditampilkan dengan warna merah.
24. Pengujian PA-24 mengeluarkan hasil sesuai harapan dimana aplikasi yang baru saja dihilangkan dari daftar favorit ditampilkan dengan warna *default*.
25. Pengujian PA-25 mengeluarkan hasil sesuai harapan dimana situs dengan *domain* yang baru saja dihilangkan dari daftar favorit ditampilkan dengan warna *default*.

5.4.2. Evaluasi Pengujian Non Fungsionalitas

Pengujian non fungsionalitas telah dilakukan dengan beberapa skenario yang ada dan memberikan hasil yang sesuai harapan. Evaluasi dari proses-proses pengujian dijelaskan sebagai berikut.

1. Pengujian PB-1 mengeluarkan hasil sesuai harapan dimana cara pengaktifan kembali servis menggunakan *broadcast* mampu menyasati terhentinya proses pemantauan ketika pengguna melakukan pembersihan memori melalui *task manager*.
2. Pengujian PB-2 mengeluarkan hasil sesuai harapan dimana hasil menunjukkan bahwa ternyata aplikasi luar tidak mengganggu kinerja aplikasi pemantau saat melakukan proses pembersihan memori.
3. Pengujian PB-3 mengeluarkan hasil sesuai harapan dimana cara pengaktifan kembali servis menggunakan *broadcast* mampu menangani masalah berhentinya servis saat pengguna menghentikan servis secara paksa melalui menu *application manager*.
4. Pengujian PB-4 mengeluarkan hasil yang relatif sesuai harapan. Jumlah konsumsi memori yang fluktuatif menunjukkan bahwa aplikasi pemantau dapat bertahan dengan baik ketika servis-servis miliknya berjalan dalam waktu yang lama. Hal ini mengindikasikan bahwa proses-proses utama milik aplikasi pemantau tetap bisa

bekerja dengan baik ketika sistem operasi melakukan pembersihan memori secara berkala. Konsumsi maksimal memori yang dilakukan oleh aplikasi pemantau relatif sekitar 60 MB.

Hasil pengujian non fungsionalitas pada aplikasi pemantau sebenarnya bergantung pada spesifikasi dan kondisi ROM dan *firmware* perangkat. Khusus pada perangkat yang digunakan pada pengujian kali ini, pengguna harus mengaktifkan opsi *auto-start* pada bagian *permission list* yang ada pada menu *setting* di perangkat. Jika tidak, secara *default* ROM perangkat-perangkat yang tidak dalam status *root* tidak akan mengizinkan servis-servis pada suatu aplikasi untuk dapat menyala secara otomatis.

DAFTAR PUSTAKA

Anna, L. K. (2011, Oktober 26). *Bahaya Balita Gemar Memainkan iPad*. Dipetik Mei 13, 2015, dari Kompas Health:

<http://health.kompas.com/read/2011/10/26/08073739/Bahaya.Balita.Gemar.Memainkan.iPad>

Hall, B. S. (2015, April 17). *Best Parental-Control Apps 2015*. Dipetik Mei 13, 2015, dari Tom's Guide:

<http://www.tomsguide.com/us/best-parental-control-apps,review-2258.html>

Singhal, M., & Shukla, A. (2012, Mei 20). Implementation of Location based Services in Android using GPS and Web Services. *IJCSI International Journal of Computer Science Issues*, 237-242.

Stenneth, L., Wolfson, O., Yu, P., & Xu, B. (2011). Transportation mode detection using mobile phones and GIS information. *19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 54-63.

BAB VI PENUTUP

Pada bab ini dijelaskan kesimpulan yang dapat diambil dalam pengerjaan tugas akhir dan saran untuk pengembangan lebih lanjut dari sistem pemantau pemakaian perangkat bergerak anak ini.

5.1. Kesimpulan

Dalam proses pengerjaan tugas akhir dari tahap pendahuluan, kajian pustaka, analisis, perancangan, implementasi dan pengujian perangkat diperoleh kesimpulan sebagai berikut.

1. Cara kerja *service* adalah dengan berjalan di belakang layar dan menjalankan proses yang diinginkan sampai ada perintah untuk berhenti. Sedangkan cara kerja *asynchronous task* adalah dengan berjalan di belakang layar dan menjalankan proses lalu berhenti dengan sendirinya ketika proses tersebut selesai. Sementara cara kerja *broadcast* adalah dengan mengirimkan sebuah pesan atau perintah yang dibaca oleh kelas penerimanya untuk melakukan *trigger* tertentu.
2. *Service* pada aplikasi pemantau dapat dibuat menyala kembali dengan seketika saat baru saja dimatikan dengan cara mengirimkan sebuah *broadcast* kepada suatu kelas yang di dalamnya terdapat fungsi untuk menyalakan kembali *service* ketika *service* tersebut dimatikan.
3. Aplikasi pemantau pada perangkat yang dibangun berbasis android dapat memantau aktivitas dan lokasi perangkat bergerak.
4. Aplikasi pemantau pada perangkat yang dibangun berbasis android dapat memproduksi statistik pemakaian perangkat bergerak.

5. Sistem pemantau pemakaian perangkat bergerak mampu mengirimkan data beserta statistik yang diproduksi oleh aplikasi ke alamat *email* tujuan secara terjadwal.

5.2. Saran

Berikut ini merupakan beberapa saran mengenai pengembangan lebih lanjut sistem pemantau pemakaian perangkat bergerak anak berdasarkan hasil rancangan, implementasi dan uji coba yang telah dilakukan.

1. Perbaiki tampilan antarmuka pada aplikasi perangkat bergerak. Tampilan yang elegan dan mengikuti panduan *material design* yang terbaru akan menarik masyarakat untuk menggunakan aplikasi ini.
2. Perbaiki tampilan *report* yang dikirimkan oleh sistem ke alamat *email* tujuan. Tampilan statistik dalam bentuk grafik yang lebih informatif akan memudahkan pengguna membaca informasi di dalamnya.
3. Menambahkan fitur pencatat *traffic* media sosial maupun aplikasi *chatting*. Hal ini untuk menyikapi semakin seringnya penggunaan layanan seperti ini untuk sarana komunikasi antar individu.
4. Aplikasi ini masih memiliki kelemahan pada pencatatan judul situs. Pada saat halaman situs belum terbuka pada kurun waktu tertentu karena koneksi jaringan yang lambat, maka yang dicatat sebagai judul adalah alamat situs tersebut. Untuk itu perlu adanya perbaikan pada pengembangan selanjutnya.
5. Ikon aplikasi ini masih terlihat ketika terpasang pada perangkat. Hal ini memungkinkan anak menghapus aplikasi dari perangkatnya. Diperlukan suatu metode implementasi yang berbeda agar ikon aplikasi tersembunyi sehingga anak tidak tahu bahwa dia sedang dipantau oleh orang tua.