

PEMBUATAN APLIKASI SMS SCHEDULER
SMS SCHEDULER APPLICATION DEVELOPMENT

Laporan ini disusun untuk memenuhi salah satu syarat menyelesaikan
Pendidikan Diploma III Program Studi Teknik Informatika
Jurusan Teknik Komputer dan Informatika

Penguji :
Ade Chandra Nugraha, S.Si., M.T.
Joe Lian Min, M.Eng

Disusun oleh :
Keukeu Anggarani Putri **101511015**
Laras Ervintyana Deni K.S **101511016**
Rizal Zulfikar Rinanda **101511028**
KOTA 109



JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA
POLITEKNIK NEGERI BANDUNG
2013

PEMBUATAN APLIKASI SMS SCHEDULER

Disusun oleh :

Keukeu Anggarani Putri	101511015
Laras Ervintyana Deni K.S.	101511016
Rizal Zulfikar Rinanda	101511028

Tugas Akhir ini telah disidangkan pada tanggal **8 Juli 2013** dan

disahkan sesuai ketentuan

Pembimbing II,

Pembimbing I

Titis Sutisna, S.ST
NIP. 19710801 200701 1 002

Santi Sundari, S.Si., M.T
NIP. 19710903 199903 2 001

Penguji II,

Penguji I,

Joe Lian Min, M.Eng
NIP. 19661018 199512 1 001

Ade Chandra Nugraha, S.Si, M.T
NIP. 19731227 199903 1 003

Ketua Program Studi Teknik Informatika D3

Irwan Setiawan, S.Si, M.T
NIP. 19800419 200501 1 002



Nama : Keukeu Anggarani Putri
NIM : 101511015

Tempat, Tanggal Lahir : Bandung, 23 April 1992
SD Lulus Tahun : 2004 dari SD Negeri Pancasila Lembang
SLTP Lulus Tahun : 2007 dari SMP Negeri 2 Bandung
SLTA Lulus Tahun : 2010 dari SMA Negeri 1 Bandung
Prestasi yang pernah dicapai : -



Nama : Laras Ervintyana Deni Kartika Sari
NIM : 101511016

Tempat, Tanggal Lahir : Madiun, 1 November 1992
SD Lulus Tahun : 2004 dari SDS Purnama Cimahi
SLTP Lulus Tahun : 2007 dari SMPN 2 Cimahi
SLTA Lulus Tahun : 2010 dari SMAN 2 Cimahi
Prestasi yang pernah dicapai : -



Nama : Rizal Zulfikar Rinanda
NIM : 101511028

Tempat, Tanggal Lahir : Cirebon, 20 Juli 1992
SD Lulus Tahun : 2004 dari SD Negeri Sadagori 1 Cirebon
SLTP Lulus Tahun : 2007 dari SMP Negeri 7 Cirebon
SLTA Lulus Tahun : 2010 dari SMA Negeri 2 Cirebon
Prestasi yang pernah dicapai :-

ABSTRAK

Tugas akhir ini berkenaan dengan pembuatan aplikasi pengiriman pesan. Aplikasi ini mengirimkan pesan-pesan SMS yang sebelumnya telah dijadwalkan. Aplikasi ini disebut SMS *Scheduler*.

Terdapat permasalahan yang belum ditangani pada aplikasi-aplikasi SMS *Scheduler* yang ada saat ini, baik pada aplikasi ber-*platform* Android, Blackberry, iOS, maupun Java ME. Permasalahan-permasalahan tersebut adalah tidak adanya informasi status pengiriman pesan, tidak adanya pemberitahuan ketika pesan akan dikirimkan, dan tidak adanya fitur yang menangani pengiriman pesan khusus untuk ucapan selamat ulang tahun atau anniversary, dimana pada pesan tersebut terdapat tanggal, bulan, tahun atau usia yang dapat diproses setiap kali pesan dikirimkan. Masalah lainnya terdapat pada aplikasi yang ber*platform* Java ME, yaitu mekanisme penggunaan nomor *recipient* harus ditulis secara manual (tidak dapat memilih dari kontak *device*). Ada pula masalah pada aplikasi yang ber*platform* Blackberry, yaitu mekanisme pengambilan nomor dari kontak harus diulang sebanyak penerima pesan dan ketika melakukan pemilihan kontak, tidak terlihat kontak mana yang sebelumnya sudah dipilih. Ketidaklengkapan fitur-fitur tersebut akan dipenuhi dalam aplikasi SMS *Scheduler* yang dibangun.

Aplikasi telah berhasil dibangun dan ditest. Hasil *test* menunjukkan bahwa aplikasi dapat mengirimkan pesan sesuai *schedule* yang sudah dibuat, dapat menampilkan notifikasi pengiriman pesan, dapat menyediakan fungsi untuk mengakses kontak *device*, dan dapat menyediakan *template* pesan, baik *template* biasa maupun *template* khusus untuk memberikan ucapan selamat ulang tahun atau *anniversary*.

Manfaat dari aplikasi ini diantaranya adalah *user* dapat menggunakan *multiple recipient* tanpa harus menuliskannya secara manual, *user* dapat mengetahui status dari pesan yang dikirimkannya, dan *user* dapat menggunakan *template* khusus jika ingin mengirimkan pesan untuk mengucapkan selamat ulang tahun atau *anniversary*. Namun, aplikasi ini baru bisa dijalankan pada *device* dengan *platform* Android. Oleh karena itu, rancangan aplikasi yang sudah dibuat dapat dijadikan *prototype* sebagai dasar pembuatan aplikasi untuk *platform* lain.

Kata Kunci: SMS *Scheduler*, mobile application, *platform*.

ABSTRACT

This final project deals with the development of messaging application. This application delivers SMS messages which have been previously scheduled. This application is called SMS Scheduler.

There exist problems that have not been handled in current SMS Scheduler application, either on Android, Blackberry, iOS, or Java ME based application. These problems include no information of message status, no information when message will be sent, and there is no feature that can handle particular message which contain date, month, year or age which can be processed every time the message is sent. Another problem exists in the application which runs on the Java ME platform, that is the mechanism to use recipient number must be written manually (it cannot choose the contact from device). In addition, there is a problem which exists in the Blackberry platform based application as well, that is the mechanism to take a number from the device contact should be repeated as many as recipient numbers and when the contact is chosen, we cannot see which the number that previously have been selected. These uncompleteness of features will be fulfilled in this SMS Scheduler application.

The application has been built successfully and tested. The test result showed that the application can send a message according to its time, can show sending notification, can provide function to access device contact, and can provide message template, both template for normal use and template for saying happy birthday or anniversary.

The advantages of this application are user can use multiple recipient without write it manually, user can know a status of a message that he/she sent, and user can use a template if he/she wants to send a message to say happy birthday or anniversary. However, this application can only be executed on Android platform device. Therefore, the application design can be used as a prototype for the basis of the application development for other platforms.

Key words: SMS Scheduler, mobile application, platform

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Allah SWT, karena atas rahmat dan ridho-Nya penulis dapat menyelesaikan laporan Tugas Akhir dengan judul “Pembuatan Aplikasi SMS *Scheduler*” ini sesuai dengan apa yang diharapkan dan tepat pada waktunya. Laporan ini disusun untuk memenuhi satu syarat menyelesaikan Pendidikan Diploma III Program Studi Teknik Informatika Jurusan Teknik Komputer dan Informatika.

Dalam pelaksanaannya, penulis menyadari bahwa penulis telah mendapatkan banyak hal dari berbagai pihak berupa bantuan, arahan, bimbingan, serta dukungan baik secara moril maupun materil. Oleh karena itu, kami ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

- Orang tua penulis yang telah memberikan kami dukungan secara moril, materil maupun spiritual selama kami berkuliahan di Jenjang Diploma III Program Studi Teknik Informatika Jurusan Teknik Komputer dan Informatika Politeknik Negeri Bandung terlebih ketika kami sedang menyelesaikan Tugas Akhir kami.
- Ibu Santi Sundari, S.Si., M.T. selaku Pembimbing I yang telah membimbing penulis, memberikan ide, gagasan, pengarahan dan motivasi serta bimbingannya yang cukup intensif pada penulis dalam menyusun dan menyelesaikan Tugas Akhir ini, serta selaku Pengaji I yang telah memberikan kritik dan masukan pada Seminar I.
- Bapak Titis Sutisna, S.ST. selaku Pembimbing II dan Pengaji II pada Seminar I, yang telah memberikan bimbingan, dukungan, pengarahan, saran, dan motivasi kepada penulis dalam menyusun dan menyelesaikan Tugas Akhir ini, serta selaku Pengaji II yang telah memberikan kritik dan masukan pada Seminar I.
- Bapak Ade Chandra Nugraha, S.Si., MT. selaku Pengaji I yang telah memberikan kritik dan masukan pada Seminar II, Seminar III dan Sidang.
- Bapak Joe Lian Min, M.Eng., selaku Pengaji II yang telah memberikan kritik dan masukan pada Seminar III dan Sidang.
- Ibu Dr. Transmissia Semiawan, selaku koordinator Tugas Akhir yang telah memberikan arahan, memberikan masukan dan selalu mengingatkan kami untuk selalu serius dalam mengerjakan Tugas Akhir.

- Bapak Suprihanto, BSEE, selaku koordinator Tugas Akhir yang telah memberikan arahan dan masukan dalam mengerjakan Tugas Akhir.
- Seluruh sahabat serta rekan-rekan kami di Jurusan Teknik Komputer dan Informatika angkatan 2010 atas semua dukungan, bantuan, serta canda tawanya selama kami menutut ilmu di Politeknik Negeri Bandung.
- Pihak-pihak lain yang tidak dapat kami sebutkan satu persatu atas segala hal yang telah diberikan selama kami menuntut ilmu di Politeknik Negeri Bandung.

Penulis menyadari masih banyak kekurangan dalam penyusunan laporan ini yang semata-mata dikarenakan keterbatasan dari kemampuan penulis. Untuk itu penulis mengharapkan kritik dan saran yang membangun dari semua pihak.

Akhir kata, semoga laporan ini dapat bermanfaat bagi berbagai pihak. Semoga Allah SWT, melimpahkan rahmat, taufik dan hidayah-Nya serta membalaas amal yang telah diberikan kepada kami. Amin.

Bandung, Juli 2013

Kelompok TA 109

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	iii
DAFTAR GAMBAR	v
DAFTAR TABEL.....	vii
DAFTAR SINGKATAN	xi
DAFTAR SIMBOL	xii
BAB I Pendahuluan.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	3
1.3 Tujuan dan Manfaat Sistem.....	4
1.4 Metode Pengembangan Aplikasi	4
1.5 Batasan Aplikasi.....	11
1.6 Lingkup Sistem.....	12
BAB II Tinjauan Pustaka.....	13
2.1 Teori Pendukung	13
2.1.1 <i>Short Message Services (SMS)</i>	13
2.1.2 <i>SMS Scheduler</i>	14
2.1.3 <i>Software Development Life Cycle (SDLC)</i>	15
2.1.4 Android	18
2.1.5 Blackberry.....	21
2.1.6 iOS	21
2.1.7 Java ME/ J2ME.....	22
2.1.8 Bahasa Pemrograman Java	23
2.1.9 <i>Predictive Search</i>	24
2.1.10 <i>Native Development</i>	25
2.1.11 <i>Mapping Tools (Pemetaan Tools)</i>	25
2.1.12 Teknologi Pendukung	26
2.1.13 Eclipse IDE	26
2.1.14 Netbeans IDE	27
2.1.15 Xcode IDE	27
2.1.16 ADT Plugin.....	28
2.1.17 Blackberry Plugin	28
2.1.18 Android SDK	29
2.1.19 Blackberry SDK.....	29
2.1.20 iOS SDK	29
BAB III Hasil Analisis dan Pembahasan.....	30
3.1 Analisis dan Evaluasi Sistem Saat Ini	30
3.1.1 Analisis Sistem Saat Ini	31
3.1.1.1 SMS Scheduler (<i>Author : Lylynx.com</i>)	31
3.1.1.2 Auto SMS (<i>Author : Thein Min Naing</i>).....	33
3.1.1.3 SMS Scheduler (<i>Author : DiorDNA Apps</i>)	34
3.1.1.4 SMS Scheduler (<i>Author : Vinsol</i>)	35
3.1.1.5 Holo SMS Scheduler (<i>Author : Casey Vockrodt, Destruction Labs</i>)	36

3.1.2	Evaluasi Sistem Saat Ini	37
3.2	Analisis Kebutuhan	51
3.2.1	Domain Kajian.....	51
3.2.2	Teknologi	53
3.2.2.1	Teknik Pembuatan Aplikasi.....	53
3.2.2.2	<i>Tools</i>	54
3.3	Pengumpulan Data (Survei).....	56
3.3.1	Kesimpulan Hasil Survei	57
3.4	Kesimpulan Analisis dan Pembahasan	58
BAB IV	Perancangan.....	60
4.1	Perancangan Data	60
4.2	Arsitektur Sistem	61
4.2.1	Perancangan Arsitektur Aplikasi SMS <i>Scheduler</i>	61
4.2.2	Perancangan Proses Aplikasi SMS <i>Scheduler</i>	76
4.2.2.1	<i>Sequence Diagram</i> Membuat <i>Schedule</i>	77
4.3	Perancangan <i>User Interface</i>	132
BAB V	Implementasi.....	171
5.1	Batasan Implementasi.....	171
5.2	Implementasi Data	171
5.3	Implementasi Aplikasi.....	172
BAB VI	Pengujian.....	176
6.1	Perencanaan Pengujian	176
BAB VII	Penutup.....	179
7.1	Kesimpulan.....	179
7.2	Saran	180
	DAFTAR PUSTAKA	181
	Lampiran A <i>Software Requirement Specification</i>	A-1
	Lampiran B <i>Mapping Tools</i>	B-1
	Lampiran C Status Implementasi <i>Requirement</i>	C-1
	Lampiran D <i>Form</i> Kuisioner.....	D-1
	Lampiran E <i>Screenshot</i> Implementasi Aplikasi SMS <i>Scheduler</i>	E-1

DAFTAR GAMBAR

Gambar 1 . Model SDLC <i>Incremental</i>	16
Gambar 2 . Arsitektur Android.....	18
Gambar 3 . Java Platform, Micro Edition (Java ME).....	23
Gambar 4 . Cara kerja Java.....	24
Gambar 5 . <i>Screenshot</i> aplikasi SMS Scheduler (<i>Author</i> : Lylynx.com)	32
Gambar 6 . <i>Screenshot</i> aplikasi Auto SMS (<i>Author</i> : Thein Min Naing).....	34
Gambar 7 . <i>Screenshot</i> aplikasi SMS Scheduler (<i>Author</i> : DiorDNA Apps).....	35
Gambar 8 . <i>Screenshot</i> aplikasi SMS Scheduler (<i>Author</i> : Vinsol).....	36
Gambar 9 . <i>Screenshot</i> aplikasi Holo SMS Scheduler (<i>Author</i> : Casey Vockrodt, Destruction Labs)	37
Gambar 10 . <i>Use case diagram current system SMS Scheduler</i>	45
Gambar 11 . <i>E-R Diagram SMS Scheduler</i>	60
Gambar 12 . <i>Physical Data Model (PDM) Database SMS Scheduler</i>	61
Gambar 13 . <i>Package diagram</i> aplikasi SMS Scheduler	62
Gambar 14 . <i>Class diagram</i> SMS Scheduler.....	63
Gambar 15 . <i>Sequence diagram New Schedule</i>	77
Gambar 16 . <i>Sequence diagram interrupt</i>	83
Gambar 17 . <i>Sequence diagram queue dan send message</i>	84
Gambar 18 . <i>Sequence diagram notification</i>	88
Gambar 19 . <i>Sequence diagram menghapus schedule</i>	89
Gambar 20 . <i>Sequence diagram Menampilkan Kategori Pesan</i>	92
Gambar 21 . <i>Sequence diagram resend schedule</i>	94
Gambar 22 . <i>Sequence diagram edit all</i>	97
Gambar 23 . <i>Sequence diagram edit recipient</i>	105
Gambar 24 . <i>Sequence diagram edit date and time schedule</i>	109
Gambar 25 . <i>Sequence diagram edit content schedule</i>	113
Gambar 26 . <i>Sequence diagram pause schedule</i>	116
Gambar 27 . <i>Sequence diagram resume schedule</i>	117
Gambar 28 . <i>Sequence diagram menampilkan alert</i>	119
Gambar 29 . <i>Sequence diagram membuat template</i>	121
Gambar 30 . <i>Sequence diagram menghapus template</i>	124
Gambar 31 . <i>Sequence diagram setting</i>	126
Gambar 32 . <i>Statechart</i> : UI halaman <i>New Schedule (Form New Schedule)</i>	135
Gambar 33 . <i>Statechart</i> : UI halaman <i>contact picker</i>	137
Gambar 34 . <i>Statechart</i> : UI halaman <i>date and time picker</i>	138
Gambar 35 . <i>Statechart</i> : UI halaman <i>warning invalid schedule</i>	139
Gambar 36 . <i>Statechart</i> : UI halaman <i>warning schedule time expired</i>	140
Gambar 37 . <i>Statechart</i> : UI halaman <i>Edit Schedule (Edit All)</i>	143
Gambar 38 . <i>Statechart</i> : UI halaman <i>Edit Schedule (Edit Content)</i>	144
Gambar 39 . <i>Statechart</i> : UI halaman <i>Edit Schedule (Edit Recipient)</i>	145
Gambar 40 . <i>Statechart</i> : UI halaman <i>Edit Schedule (Edit Date and Time)</i>	146
Gambar 41 . <i>Statechart</i> : UI halaman menghapus <i>schedule</i>	147
Gambar 42 . <i>Statechart</i> : UI halaman Kategori Pesan - <i>List Scheduled</i>	149
Gambar 43 . <i>Statechart</i> : UI halaman Kategori Pesan - <i>List Sent</i>	150

Gambar 44 .	<i>Statechart</i> : UI halaman Kategori Pesan - <i>List Failed</i>	152
Gambar 45 .	<i>Statechart</i> : UI halaman Kategori Pesan - <i>List Paused</i>	153
Gambar 46 .	UI halaman <i>history</i>	154
Gambar 47 .	<i>Statechart</i> : UI halaman <i>detail history</i>	155
Gambar 48 .	<i>Statechart</i> : UI halaman <i>alert</i>	156
Gambar 49 .	<i>Statechart</i> : UI halaman <i>detail alert</i>	157
Gambar 50 .	<i>Statechart</i> : UI halaman <i>notification sending</i>	158
Gambar 51 .	<i>Statechart</i> : UI halaman <i>notification</i> (laporan pengiriman).....	160
Gambar 52 .	<i>Statechart</i> : UI halaman <i>template</i>	161
Gambar 53 .	<i>Statechart</i> : UI halaman <i>New Template</i>	163
Gambar 54 .	<i>Statechart</i> : UI halaman <i>invalid date</i>	164
Gambar 55 .	<i>Statechart</i> : UI halaman <i>setting</i>	165
Gambar 56 .	<i>Statechart</i> : UI halaman <i>Setting Alert</i>	166
Gambar 57 .	<i>Statechart</i> : UI halaman <i>setting alert time</i>	167
Gambar 58 .	<i>Statechart</i> : UI halaman <i>setting notification</i>	168
Gambar 59 .	<i>Statechart</i> : UI halaman <i>sound picker</i>	169
Gambar 60 .	<i>Statechart</i> : UI halaman <i>about us</i>	170
Gambar 61 .	<i>Component diagram</i> aplikasi SMS <i>Scheduler</i>	172

DAFTAR TABEL

Tabel 1.	Jadwal <i>increment</i>	5
Tabel 2.	Hasil analisis Aplikasi SMS <i>Scheduler</i> yang ada saat ini	40
Tabel 3.	Hasil analisis Aplikasi SMS <i>Scheduler</i> yang ada saat ini (Lanjutan).....	41
Tabel 4.	Hasil analisis Aplikasi SMS <i>Scheduler</i> yang ada saat ini (Lanjutan).....	42
Tabel 5.	Hasil analisis Aplikasi SMS <i>Scheduler</i> yang ada saat ini (Lanjutan).....	43
Tabel 6.	Hasil analisis Aplikasi SMS <i>Scheduler</i> yang ada saat ini (Lanjutan).....	44
Tabel 7.	Deskripsi aktor.....	45
Tabel 8.	Deskripsi <i>case</i>	45
Tabel 9.	Skenario <i>use case add schedule</i>	46
Tabel 10.	Skenario <i>case add schedule</i> (lanjutan)	47
Tabel 11.	Skenario <i>case update schedule</i>	47
Tabel 12.	Skenario <i>case update schedule</i> (lanjutan).....	48
Tabel 13.	Skenario <i>case delete schedule</i>	49
Tabel 14.	Skenario <i>case send message from schedule</i>	49
Tabel 15.	Skenario <i>case send message from schedule</i> (lanjutan).....	50
Tabel 16.	Skenario <i>case show notification</i>	51
Tabel 17.	Skenario <i>case show history</i>	51
Tabel 18.	Hasil Survei	57
Tabel 19.	Keterangan <i>Package Diagram</i>	62
Tabel 20.	Keterangan <i>Class ScheduleDatabaseHelper</i>	64
Tabel 21.	Keterangan <i>Class ScheduleDatabaseOpenHelper</i>	66
Tabel 22.	Keterangan <i>Class Schedule</i>	67
Tabel 23.	Keterangan <i>Class SMSActivity</i>	69
Tabel 24.	Keterangan <i>Class MessageInterval</i>	69
Tabel 25.	Keterangan <i>Class NewScheduleView</i>	70
Tabel 26.	Keterangan <i>Class ListView</i>	71
Tabel 27.	Keterangan <i>Class Edit</i>	72
Tabel 28.	Keterangan <i>Class Alert</i>	72
Tabel 29.	Keterangan <i>Class NewTemplateView</i>	73
Tabel 30.	Keterangan <i>Class Template</i>	73
Tabel 31.	Keterangan <i>Class SettingView</i>	74
Tabel 32.	Keterangan <i>Class DefaultSetting</i>	75
Tabel 33.	Keterangan <i>sequence diagram New Schedule</i>	78
Tabel 34.	Spesifikasi proses <i>saveToDatabase</i>	78
Tabel 35.	Spesifikasi proses <i>get messageIdFromMessage</i>	79
Tabel 36.	Spesifikasi proses <i>setMessageType</i>	80
Tabel 37.	Spesifikasi proses <i>getMessageType</i>	80
Tabel 38.	Spesifikasi proses <i>saveScheduleToMessage</i>	80
Tabel 39.	Spesifikasi proses <i>saveScheduleToType</i>	81
Tabel 40.	Spesifikasi proses <i>saveScheduleToTime</i>	81
Tabel 41.	Spesifikasi proses <i>saveScheduleToContact</i>	82
Tabel 42.	Spesifikasi proses <i>saveScheduleToRecipient</i>	82
Tabel 43.	Spesifikasi proses <i>repetition</i>	82

Tabel 44.	Keterangan <i>sequence diagram interrupt</i>	84
Tabel 45.	Keterangan <i>sequence diagram queue</i> dan <i>send message</i>	85
Tabel 46.	Spesifikasi proses <i>get messageIdFromTime</i>	85
Tabel 47.	Spesifikasi proses <i>getMessageFromMessage</i>	86
Tabel 48.	Spesifikasi proses <i>getRecipientFromRecipient</i>	86
Tabel 49.	Spesifikasi proses <i>sendSMS</i>	87
Tabel 50.	Keterangan <i>sequence diagram notification</i>	88
Tabel 51.	Keterangan <i>sequence diagram menghapus schedule</i>	89
Tabel 52.	Spesifikasi proses <i>deleteRecipient</i>	89
Tabel 53.	Spesifikasi proses <i>getMessageType</i>	90
Tabel 54.	Spesifikasi proses <i>deleteRecipient</i>	90
Tabel 55.	Spesifikasi proses <i>deleteMessageType</i>	91
Tabel 56.	Spesifikasi proses <i>getTime</i>	91
Tabel 57.	Spesifikasi proses <i>deleteTime</i>	91
Tabel 58.	Spesifikasi proses <i>deleteMessage</i>	92
Tabel 59.	Keterangan <i>sequence diagram Menampilkan Kategori Pesan</i>	93
Tabel 60.	Keterangan <i>sequence diagram resend schedule</i>	94
Tabel 61.	Spesifikasi proses <i>getRecipient</i>	95
Tabel 62.	Spesifikasi proses <i>getMessage</i>	95
Tabel 63.	Spesifikasi proses <i>sendSMS</i>	95
Tabel 64.	Keterangan <i>sequence diagram edit all schedule</i>	98
Tabel 65.	Spesifikasi proses <i>getRecipient</i>	99
Tabel 66.	Spesifikasi proses <i>getMessageDatetime</i>	100
Tabel 67.	Spesifikasi proses <i>getMessageFrequency</i>	100
Tabel 68.	Spesifikasi proses <i>getTime</i>	101
Tabel 69.	Spesifikasi proses <i>count</i>	101
Tabel 70.	Spesifikasi proses <i>getMessage</i>	101
Tabel 71.	Spesifikasi proses <i>deleteRecipient</i>	102
Tabel 72.	Spesifikasi proses <i>saveScheduleToRecipient</i>	102
Tabel 73.	Spesifikasi proses <i>getMessageType</i>	102
Tabel 74.	Spesifikasi proses <i>updateType</i>	103
Tabel 75.	Spesifikasi proses <i>updateContent</i>	103
Tabel 76.	Spesifikasi proses <i>deleteTime</i>	103
Tabel 77.	Spesifikasi proses <i>saveScheduleToTime</i>	104
Tabel 78.	Spesifikasi proses <i>repetition</i>	104
Tabel 79.	Keterangan <i>sequence diagram edit recipient</i>	105
Tabel 80.	Spesifikasi proses <i>getRecipient</i>	106
Tabel 81.	Spesifikasi proses <i>updateRecipient</i>	106
Tabel 82.	Spesifikasi proses <i>getRecipientFromRecipient</i>	107
Tabel 83.	Spesifikasi proses <i>getStringRecipients</i>	107
Tabel 84.	Spesifikasi proses <i>deleteRecipient</i>	108
Tabel 85.	Spesifikasi proses <i>saveScheduleToRecipient</i>	108
Tabel 86.	Keterangan <i>sequence diagram edit date and time schedule</i>	109
Tabel 87.	Spesifikasi proses <i>updateTime</i>	110
Tabel 88.	Spesifikasi proses <i>getTime</i>	111
Tabel 89.	Spesifikasi proses <i>count</i>	111

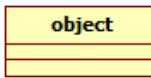
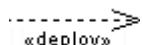
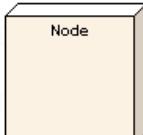
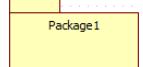
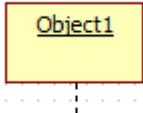
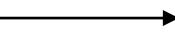
Tabel 90. Spesifikasi proses deleteTime.....	111
Tabel 91. Spesifikasi proses getMessageFrequency.....	112
Tabel 92. Spesifikasi proses saveScheduleToTime.....	112
Tabel 93. Spesifikasi proses repetition	112
Tabel 94. Keterangan <i>sequence diagram edit content schedule</i>	114
Tabel 95. Spesifikasi proses getMessage.....	114
Tabel 96. Spesifikasi proses updateType.....	115
Tabel 97. Spesifikasi proses updateContent	115
Tabel 98. Keterangan <i>sequence diagram pause schedule</i>	116
Tabel 99. Spesifikasi proses pauseSchedule.....	116
Tabel 100. Spesifikasi proses getTime	117
Tabel 101. Keterangan <i>sequence diagram resume schedule</i>	118
Tabel 102. Spesifikasi proses resumeSchedule	118
Tabel 103. Spesifikasi proses getTime	119
Tabel 104. Keterangan <i>sequence diagram menampilkan alert</i>	120
Tabel 105. Keterangan <i>sequence diagram membuat template</i>	122
Tabel 106. Spesifikasi proses saveTemplateToCategory.....	122
Tabel 107. Spesifikasi proses getCategoryId.....	123
Tabel 108. Spesifikasi proses saveTemplateToTemplate	123
Tabel 109. Keterangan <i>sequence diagram menghapus template</i>	124
Tabel 110. Spesifikasi proses deleteTemplate	124
Tabel 111. Keterangan <i>sequence diagram setting</i>	127
Tabel 112. Spesifikasi proses getDefaultAlertActivation	128
Tabel 113. Spesifikasi proses getDefaultAlertTime	128
Tabel 114. Spesifikasi proses getDefaultAlertTone	128
Tabel 115. Spesifikasi proses getDefaultNotifTone	129
Tabel 116. Spesifikasi proses saveSettingToSetting	129
Tabel 117. Spesifikasi proses getAlertActivation.....	129
Tabel 118. Spesifikasi proses updateAlertActivation	130
Tabel 119. Spesifikasi proses getAlertTime	130
Tabel 120. Spesifikasi proses updateAlertTime	130
Tabel 121. Spesifikasi proses getAlertTone	131
Tabel 122. Spesifikasi proses updateAlertTone	131
Tabel 123. Spesifikasi proses getNotifTone	131
Tabel 124. Spesifikasi proses updateNotifTone	132
Tabel 125. Rancangan UI halaman <i>New Schedule (Form New Schedule)</i>	133
Tabel 126. Rancangan UI halaman <i>New Schedule (Form New Schedule)</i> (Lanjutan)	135
Tabel 127. Rancangan UI halaman <i>New Schedule (Contact Picker)</i>	136
Tabel 128. Rancangan UI halaman <i>New Schedule (Contact Picker)</i> (Lanjutan).....	137
Tabel 129. Rancangan UI halaman <i>New Schedule</i> (Menampilkan Pengaturan Tanggal dan Waktu)	137
Tabel 130. Rancangan UI halaman <i>New Schedule</i> (Menampilkan Pengaturan Tanggal dan Waktu) (Lanjutan)	138
Tabel 131. Rancangan UI halaman <i>New Schedule</i> (Menampilkan warning invalid schedule)	139

Tabel 132. Rancangan UI halaman <i>New Schedule</i> (Menampilkan <i>warning schedule time expired</i>)	140
Tabel 133. Rancangan UI halaman <i>Edit Schedule (Edit All)</i>	141
Tabel 134. Rancangan UI halaman <i>Edit Schedule (Edit All)</i> (lanjutan)	142
Tabel 135. Rancangan UI halaman <i>Edit Schedule (Edit All)</i> (Lanjutan)	143
Tabel 136. Rancangan UI halaman <i>Edit Schedule (Edit Content)</i>	144
Tabel 137. Rancangan UI halaman <i>Edit Schedule (Edit Recipient)</i>	145
Tabel 138. Rancangan UI halaman <i>Edit Schedule (Edit Date and Time)</i>	146
Tabel 139. Rancangan UI halaman menghapus <i>schedule</i>	147
Tabel 140. Rancangan UI halaman kategori pesan (<i>List Schedule</i>)	148
Tabel 141. Rancangan UI halaman kategori pesan (<i>List Sent</i>)	149
Tabel 142. Rancangan UI halaman kategori pesan (<i>List Failed</i>)	151
Tabel 143. Rancangan UI halaman kategori pesan (<i>List Paused</i>)	152
Tabel 144. Rancangan UI halaman <i>history</i>	154
Tabel 145. Rancangan UI halaman <i>detail history</i>	155
Tabel 146. Rancangan UI halaman <i>Alert</i>	156
Tabel 147. Rancangan UI halaman <i>Alert (detail alert)</i>	157
Tabel 148. Rancangan UI halaman <i>Notification (sending)</i>	158
Tabel 149. Rancangan UI halaman <i>Notification (laporan pengiriman)</i>	159
Tabel 150. Rancangan UI halaman <i>Notification (laporan pengiriman)</i> (Lanjutan)	160
Tabel 151. Rancangan UI halaman <i>template</i>	160
Tabel 152. Rancangan UI halaman <i>Template (New template)</i>	162
Tabel 153. Rancangan UI halaman <i>Template (New template)</i> (Lanjutan)	163
Tabel 154. Rancangan UI halaman <i>Template (Invalid Date)</i>	164
Tabel 155. Rancangan UI halaman <i>setting</i>	165
Tabel 156. Rancangan UI halaman <i>setting(alert)</i>	166
Tabel 157. Rancangan UI halaman <i>setting(Alert Time)</i>	167
Tabel 158. Rancangan UI halaman <i>setting(notification)</i>	168
Tabel 159. Rancangan UI halaman <i>setting(sound picker)</i>	169
Tabel 160. Rancangan UI halaman <i>about us</i>	170
Tabel 161. Status implementasi data	171
Tabel 162. Penjelasan deployment diagram	173
Tabel 163. <i>Class</i> dan <i>method</i> Android yang digunakan dalam pembuatan aplikasi	173
Tabel 164. <i>Test Case</i>	176

DAFTAR SINGKATAN

<i>Singkatan</i>	<i>Deskripsi</i>
IDE	<i>Integrated Development Environment</i>
J2ME/ Java ME	<i>Java Platform Micro Edition</i>
OS	<i>Operating System</i>
SDK	<i>Software Development Kit</i>
SMS	<i>Short Message Service</i>
SRS	<i>Software Requirement Specification</i>

DAFTAR SIMBOL

Simbol	Nama	Penggunaan	Keterangan
	<i>Agregation</i>	<i>Class diagram</i>	Menggambarkan relasi dengan perlakuan khusus yang disebut dengan “part-of” (bagian dari) yang menangani antar objek dimana salah satunya adalah bagian dari objek lain
	<i>Assosiation</i>	<i>Class diagram</i>	Menggambarkan relasi antar <i>class</i>
	<i>Class</i>	<i>Class diagram</i>	Merepresentasikan suatu objek, dengan kelengkapan methods dan attributes.
	<i>Dependency</i>	<i>Class diagram</i>	Menggambarkan kebergantungan antar <i>class</i> , dimana jika terjadi perubahan pada <i>element</i> yang <i>independent</i> , maka akan mempengaruhi <i>element</i> yang tidak <i>independent</i>
	<i>Deployment</i>	<i>Deployment diagram</i>	Menggambarkan kebergantungan antar komponen
	<i>Node</i>	<i>Deployment diagram</i>	Menggambarkan sumber daya yang digunakan pada saat aplikasi dijalankan
	<i>Atribut</i>	<i>E-R diagram</i>	Menggambarkan atribut dari setiap <i>entity</i> .
	<i>Entity</i>	<i>E-R diagram</i>	Menggambarkan <i>entity</i> .
	<i>Relationship</i>	<i>E-R diagram</i>	Menggambarkan relasi antar <i>entity</i> .
	<i>Package</i>	<i>Package diagram</i>	Merepresentasikan package dari <i>class</i> .
	<i>Message to self</i>	<i>Sequence diagram</i>	Menggambarkan pesan yang menuju dirinya sendiri.
	<i>Object</i>	<i>Sequence diagram</i>	Menggambarkan objek pada diagram.
	<i>Object message</i>	<i>Sequence diagram</i>	Menggambarkan pesan antar dua objek

DAFTAR SIMBOL (LANJUTAN)

Simbol	Nama	Penggunaan	Keterangan
	<i>Return message</i>	<i>Sequence diagram</i>	Menggambarkan pengembalian dari pemanggilan prosedur.
	<i>State</i>	<i>Statechart diagram</i>	Menggambarkan suatu <i>state</i> pada diagram.
	<i>Transition</i>	<i>Statechart diagram</i>	Menggambarkan transisi dari suatu aktivitas ke aktivitas lainnya.
	<i>Aktor</i>	<i>Use case diagram</i>	Menggambarkan aktor pada diagram kelas.
	<i>Association</i>	<i>Use case diagram</i>	Menggambarkan hubungan asosiasi antar <i>use case</i> .
	<i>Extend</i>	<i>Use case diagram</i>	Menggambarkan relasi <i>use case</i> tambahan, dimana <i>use case</i> tersebut dapat berdiri sendiri tanpa <i>use case</i> tambahan tersebut.
	<i>Include</i>	<i>Use case diagram</i>	Menggambarkan relasi <i>use case</i> dimana proses yang bersangkutan akan dilanjutkan ke proses yang dituju.
	<i>Use case</i>	<i>Use case diagram</i>	Menggambarkan proses pada diagram <i>use case</i> .

BAB I

Pendahuluan

1.1 Latar Belakang

Ada banyak cara untuk menyampaikan informasi, salah satunya dengan melalui SMS (*Short Message Service*) atau pesan singkat. Dengan menggunakan SMS, suatu informasi bisa lebih cepat disampaikan dan langsung disampaikan kepada orang yang bersangkutan, baik SMS untuk individu maupun *massal/ broadcast*. Namun terkadang penyampaian informasi melalui SMS ini justru terhambat baik dari faktor pengirim pesan maupun dari operator. Masalah dari sisi pengirim pesan contohnya tertundanya informasi yang harus disampaikan untuk beberapa waktu kedepan sehingga informasi menjadi tidak tersampaikan.

Untuk mengatasi masalah diatas, saat ini telah tersedia aplikasi-aplikasi untuk melakukan pengiriman pesan yang sudah dijadwal terlebih dahulu (baik untuk individu maupun *massal/ broadcast*), dimana pesan yang dibuat *user* dapat disimpan terlebih dahulu menjadi sebuah *schedule* dan nantinya akan dikirimkan secara otomatis oleh aplikasi sesuai waktu yang telah ditentukan *user*. Artinya, pesan yang dibuat pada aplikasi tersebut tidak harus langsung dikirimkan pada saat itu juga.

Namun, pada aplikasi yang ada saat ini masih terdapat beberapa masalah yang belum ditangani, salah satunya adalah kurangnya informasi pengiriman pesan bagi *user* yang mengirimkan pesan (notifikasi), yaitu informasi pesan yang gagal terkirim dan informasi pesan telah diterima oleh penerima pesan (dalam hal ini pesan sudah masuk ke *device* penerima pesan). Tidak adanya notifikasi pesan yang gagal dan pesan telah diterima bagi pengirim pesan menyebabkan ia tidak mengetahui ketika pesan yang dikirimkannya ternyata gagal terkirim dan ketika pesan telah diterima oleh penerima pesan.

Ada pula masalah mengenai *template* pesan, *alert* dan *setting* aplikasi. *Template* pesan yang ada pada aplikasi SMS *Scheduler* saat ini hanya menyediakan *template* pesan biasa, seperti “*I’m busy*”, “*I’m in a meeting. I’ll contact you later*”, “*Happy Birthday*”, dan lainnya. Namun, *template* pesan khusus untuk memberi ucapan selamat ulang tahun atau hari jadi belum tersedia. *Template* pesan khusus yang dimaksud disini adalah *template* pesan didalamnya terdapat tahun atau usia yang dapat berubah secara otomatis sesuai

dengan tahun peringatannya. Oleh karena itu, jika *user* ingin mengirimkan pesan untuk memperingati hari jadi atau hari ulang tahun seseorang, *user* harus merubah sendiri tahun atau usianya setiap kali akan mengirimkan pesan. Selain itu, pada aplikasi yang ada saat ini belum tersedia pengingat (*alert*) sebelum jadwal pengiriman dilaksanakan. Hal tersebut terkadang menjadi masalah karena ada kemungkinan *user* tiba-tiba ingin merubah isi *schedule* atau membatalkan pengiriman *schedule* saat mendekati waktu pelaksanaan *schedule*. Untuk mengatur *alert* dan notifikasi agar dapat disesuaikan dengan keinginan *user*, maka aplikasi pun perlu menyediakan fitur *setting*.

Karena aplikasi yang ada saat ini digunakan pula untuk mengirimkan *broadcast message*, maka muncul masalah lainnya yang terdapat pada aplikasi pada *platform Java ME/ J2ME* dan Blackberry. Pada aplikasi pada *platform Java ME/ J2ME*, nomor penerima pesan yang digunakan harus ditulis secara manual (tidak dapat memilih dari kontak *device*). Hal ini memungkinkan adanya kesalahan penulisan jika nomor penerima yang akan digunakan lebih dari satu nomor, dan menyebabkan informasi menjadi tidak tersampaikan pada orang yang dituju.

Sementara itu, pada aplikasi yang dapat dijalankan di *platform Blackberry*, nomor penerima pesan yang dapat digunakan dapat diambil dari kontak *device*, hanya saja mekanisme pengambilan nomor dari kontak harus diulang sebanyak penerima pesan yang akan digunakan dan ketika melalukan pemilihan kontak tidak terlihat kontak mana yang sebelumnya sudah dipilih. Hal ini memungkinkan nomor yang sebelumnya sudah dipilih menjadi terpilih kembali, dan memungkinkan nomor lainnya yang ingin digunakan justru tidak terpilih.

Untuk mengatasi masalah-masalah tersebut, maka dibuatlah aplikasi untuk melakukan pengiriman SMS terjadwal yang diberi nama SMS *Scheduler* yang dapat melaksanakan fungsi utamanya, yaitu membuat jadwal pengiriman pesan dan mengirimkan pesan sesuai jadwal tersebut, serta didukung dengan beberapa fitur tambahan lainnya untuk lebih membantu penggunaan aplikasi, seperti fitur notifikasi, *alert*, *template* pesan dan *setting*. Selain itu, dikarenakan masalah-masalah diatas belum dapat ditangani pada aplikasi yang ada saat ini, baik aplikasi pada *platform Android*, Blackberry, iOS, dan Java ME/ J2ME, ditambah dengan adanya masalah mekanisme *input* nomor *recipient* pada aplikasi dengan *platform Blackberry* dan Java ME/ J2ME, maka aplikasi SMS *Scheduler* ini pun dibuat untuk beberapa *platform*, yaitu Android, Blackberry, iOS, dan Java ME/ J2ME.

1.2 Perumusan Masalah

Dari latar belakang diatas, dapat dirumuskan beberapa permasalahan dari aplikasi SMS *Scheduler* untuk *platform* Android, Blackberry, iOS, dan Java ME/ J2ME yang ada saat ini sebagai berikut :

- a. Tidak adanya informasi pengiriman pesan berhasil dikirim atau gagal dikirim, sehingga pengirim pesan tidak mengetahui apakah pesan yang dikirimnya berhasil atau gagal.
- b. Tidak adanya informasi jika pesan yang dikirim telah diterima oleh *recipient* (masuk di *device* penerima pesan), sehingga memungkinkan pengirim pesan melakukan pengiriman lagi untuk meyakinkan dirinya sendiri bahwa pesan sudah diterima *recipient*.
- c. Pada aplikasi di *platform* Java ME/ J2ME, nomor penerima pesan yang dapat digunakan harus ditulis secara manual, sehingga memungkinkan adanya kesalahan pengetikan nomor penerima dan menyebabkan informasi menjadi tidak tersampaikan pada orang yang dituju.
- d. Sementara, pada aplikasi di *platform* Blackberry, nomor penerima pesan yang dapat digunakan dapat diambil dari kontak *device*, hanya saja mekanisme pengambilan nomor dari kontak harus diulang sebanyak penerima pesan yang akan digunakan dan ketika melalukan pemilihan kontak tidak terlihat kontak mana yang sebelumnya sudah dipilih. Hal ini memungkinkan nomor yang sebelumnya sudah dipilih menjadi terpilih kembali, dan memungkinkan nomor lainnya yang ingin digunakan justru tidak terpilih
- e. Belum tersedianya fitur *template* pesan khusus untuk memberi ucapan selamat ulang tahun atau hari jadi. Oleh karena itu, jika *user* ingin mengirimkan pesan untuk memperingati hari jadi atau hari ulang tahun seseorang, *user* harus merubah sendiri tahun atau usianya setiap kali akan mengirimkan pesan.
- f. Belum tersedia pengingat (*alert*) sebelum jadwal pengiriman dilaksanakan, karena bisa saja *user* tiba-tiba ingin merubah isi *schedule* atau membatalkan pengiriman *schedule* saat mendekati waktu pelaksanaan *schedule*.

1.3 Tujuan dan Manfaat Sistem

Tujuan dari pembuatan aplikasi ini antara lain :

- a. Memberikan informasi pada *user* bahwa pesan sudah diterima oleh *recipient* (masuk di *device* penerima pesan).
- b. Memberikan informasi untuk pesan yang gagal dikirim.
- c. Menyediakan fitur pemilihan penerima pesan yang dapat mengakses kontak *device*, sehingga pengguna aplikasi tidak perlu menulis nomor penerima pesan secara manual, terutama jika penerima pesan yang dituju lebih dari satu nomor.
- d. Menyediakan fitur untuk memudahkan pengambilan nomor penerima dari kontak *device* sehingga pengguna aplikasi tidak perlu mengulangi proses pengambilan kontak sebanyak nomor penerima yang dituju.
- e. Memperlihatkan kontak mana saja yang sudah dipilih pengguna, sehingga meminimalisir pemilihan kontak yang sudah dipilih.
- f. Menyediakan fitur *template* pesan khusus untuk memberi ucapan selamat ulang tahun atau hari jadi, yaitu *template* pesan yang didalamnya dapat disisipkan tanggal ulang tahun atau hari jadi, dimana data tahun atau usia dari tanggal ulang tahun atau hari jadi tersebut nantinya dapat berubah secara otomatis sesuai dengan waktu peringatannya.
- g. Mengingatkan *user* bahwa ada pesan yang akan dikirimkan.
- h. Untuk dapat digunakan pada *platform* Android, Blackberry, iOS dan Java ME/ J2ME dengan fitur-fitur tambahan yang dijelaskan pada *point* a-e.

1.4 Metode Pengembangan Aplikasi

Metode pengembangan aplikasi yang digunakan adalah metode *incremental*, dimana setiap *increment*-nya dimaksudkan untuk menyelesaikan setiap fitur yang ada pada aplikasi SMS *Scheduler*. Metode pengembangan aplikasi ini dipilih agar dalam waktu dekat sudah ada produk yang dapat dihasilkan, minimal aplikasi dengan fitur utama.

Berikut ini tahapan-tahapan yang dilakukan dalam model *incremental* :

1. Analisis, merupakan proses penentuan kebutuhan atau analisis kebutuhan. Dilanjutkan dengan proses spesifikasi dimana menggunakan analisis kebutuhan sebagai acuannya. Pada tahap ini, digunakan *tools use case diagram* yang mengacu pada buku *Object-Oriented Analysis and Design with UML*.

2. Desain, merupakan proses perancangan desain aplikasi. Pada tahap ini digunakan *tools class diagram, sequence diagram, dan statechart diagram* yang mengacu pada buku *Object-Oriented Analysis and Design with UML*, dan *The Unified Modeling Language Reference Manual*.
3. *Code*, merupakan proses implementasi dari perancangan yang sudah dibuat.
4. *Test*, merupakan proses pengujian aplikasi. Pada tahap ini, referensi yang digunakan adalah buku *Effective Methods for Software Testing - Third Edition*.

Berikut ini merupakan penjelasan dari pekerjaan yang dikerjakan pada setiap *increment*.

Tabel 1. Jadwal *increment*

	Analisis	Design	Implementasi	Testing
Increment 1	<ul style="list-style-type: none"> • Fitur dan proses pada <i>current system</i> • Fitur dan proses pada sistem yang akan dibuat • Data yang diperlukan secara umum pada sistem yang akan dibuat 	<ul style="list-style-type: none"> • Membuat rancangan <i>database</i> • Membuat alur untuk UI • Membuat rancangan perilaku sistem yang akan dibuat • <i>Mock up UI</i> 	<ul style="list-style-type: none"> • Membuat dokumen hasil analisis aplikasi yang ada saat ini • Membuat dokumen analisis untuk aplikasi yang akan dibuat (SRS) 	Melakukan pengujian terhadap masing-masing fitur yang ada pada aplikasi saat ini
Artifak dari <i>increment 1</i> adalah hasil analisis aplikasi yang sudah ada saat ini, hasil analisis aplikasi yang akan dibuat, rancangan <i>database</i> aplikasi yang akan dibuat (secara umum), dan <i>mock up user interface</i> aplikasi yang akan dibuat (secara umum).				
Increment 2 [Fitur membuat dan mengirim <i>schedule</i>]	<ul style="list-style-type: none"> • Menentukan cara untuk : <ul style="list-style-type: none"> - mengakses kontak dari <i>device</i>. - mengakses grup dari <i>device</i> - mengambil data <i>recipient</i> dari input manual. - mengambil 1 atau lebih data <i>recipient (multiple recipient)</i> dari kontak atau grup yang ada pada <i>device</i>. - menangani proses penyimpanan data ke <i>database</i>. - menangani pengiriman pesan dan <i>repeating</i>. 	<ul style="list-style-type: none"> • Merancang algoritma untuk : <ul style="list-style-type: none"> - pembuatan <i>schedule</i> - validasi <i>form</i> dan waktu pengiriman - penyimpanan <i>schedule</i> - pengiriman <i>schedule</i> dan <i>repeating</i> - menangani proses pada <i>typical message</i>. • <i>User interface</i> untuk : <ul style="list-style-type: none"> - <i>Form</i> untuk <i>schedule</i> baru - <i>Date and time picker</i> - <i>Contact picker</i> 	<ul style="list-style-type: none"> • Mengakses kontak dari <i>device</i>. • Mengambil data <i>recipient</i> dari input manual. • Mengambil 1 atau lebih data <i>recipient (multiple)</i> dari kontak <i>device</i>. • Menyimpan data dari <i>form</i> yang diinput <i>user</i> ke <i>database</i>. • Mengirimkan pesan sesuai dengan jadwal yang dibuat dan sesuai dengan banyaknya pengulangan pengiriman yang dipilih <i>user</i>. 	<ul style="list-style-type: none"> • Melakukan <i>test</i> untuk : <ul style="list-style-type: none"> - Mengakses kontak dari <i>device</i> - mengambil nomor <i>recipient</i> yang diinput secara manual - mengambil 1 atau lebih data <i>recipient</i> dari kontak <i>device</i> - membuat beberapa <i>schedule</i> dengan waktu pengiriman yang sama. - validasi <i>form</i> dan waktu pengiriman

	Analisis	Design	Implementasi	Testing
Increment 2 [Fitur membuat dan mengirim <i>schedule</i>]	<ul style="list-style-type: none"> - Menangani <i>typical message</i> (pesan dengan data yang dapat diproses setiap kali dilakukan pengiriman) 	<ul style="list-style-type: none"> - Menampilkan pesan <i>error</i> ketika hasil validasi tidak valid 	<ul style="list-style-type: none"> • Validasi <i>form</i> dan waktu pengiriman sebelum diinput ke <i>database</i> • Menangani proses pada <i>typical message</i> • <i>User interface</i> 	<ul style="list-style-type: none"> - melakukan pengecekan <i>content message</i> untuk membedakan <i>typical message</i> dan <i>normal message</i> - melakukan pengiriman pesan sesuai dengan <i>schedule</i> yang dibuat - menjalankan komponen <i>user interface</i>

Artifak dari *increment 2* adalah hasil analisis dan *design* untuk proses-proses pada fitur membuat dan mengirimkan *schedule*, aplikasi yang sudah tersedia fitur untuk membuat *schedule* baru dan aplikasi dapat mengirimkan pesan sesuai dengan *schedule* yang telah dibuat, dan hasil *testing* untuk fitur tersebut.

	Analisis	Design	Implementasi	Testing
Increment 3 [Fitur mengubah <i>schedule</i>]	<ul style="list-style-type: none"> • Menentukan cara untuk : <ul style="list-style-type: none"> - mengambil data yang akan diubah dari <i>database</i> (sesuai dengan keinginan <i>user</i>, yaitu mengambil dan menampilkan semua data, data penerima pesan saja, data konten pesan saja atau data waktu pengiriman saja). - melakukan <i>update</i> data yang diinput <i>user</i> ke <i>database</i>. - menangani perubahan waktu pengiriman <i>schedule</i> yang sudah tersusun sebelumnya (jika <i>user</i> mengubah waktu pengiriman). 	<ul style="list-style-type: none"> • Merancang algoritma untuk : <ul style="list-style-type: none"> - mengambil data <i>schedule</i> dari <i>database</i>, baik mengambil semua data, data penerima pesan saja, data konten pesan saja atau data waktu pengiriman saja. - mengupdate <i>schedule</i> dengan data yang diinput <i>user</i>. - menangani perubahan pengiriman <i>schedule</i> yang diakibatkan dari adanya perubahan waktu pengiriman. • <i>User interface</i> untuk : <ul style="list-style-type: none"> - <i>Form</i> untuk mengubah semua data <i>schedule</i> - <i>Form</i> untuk mengubah data penerima pesan saja. 	<ul style="list-style-type: none"> • Mengambil dan menampilkan data dari <i>schedule</i> yang sebelumnya sudah tersimpan di <i>database</i> • Mengambil dan menampilkan data yang akan diubah sesuai dengan pilhan yang diinginkan <i>user</i> (semua data, data penerima pesan saja, data konten pesan saja atau data waktu pengiriman saja). • Meng-update data <i>schedule</i> di <i>database</i> dengan data yang diinput <i>user</i>. • Mengatur perubahan pengiriman <i>schedule</i> yang diakibatkan dari adanya perubahan waktu pengiriman • Validasi <i>form</i> dan waktu pengiriman sebelum diinput ke <i>database</i> 	<ul style="list-style-type: none"> • Melakukan <i>test</i> untuk : <ul style="list-style-type: none"> - mengambil dan menampilkan data dari <i>schedule</i> yang sebelumnya sudah tersimpan di <i>database</i> - mengambil dan menampilkan data yang akan diubah sesuai dengan pilhan yang diinginkan <i>user</i> (semua data, data penerima pesan saja, data konten pesan saja atau data waktu pengiriman saja) - meng-update data <i>schedule</i> di <i>database</i> dengan data yang diinput <i>user</i> - proses perubahan pengiriman <i>schedule</i> yang diakibatkan dari adanya perubahan waktu pengiriman - validasi <i>form</i> dan waktu pengiriman - mengakses kontak dari <i>device</i>.

	Analisis	Design	Implementasi	Testing
Increment 3 [Fitur mengubah <i>schedule</i>]		<ul style="list-style-type: none"> - <i>Form</i> untuk mengubah waktu pengiriman saja. - <i>Form</i> untuk mengubah isi pesan saja. - Menampilkan pesan <i>error</i> ketika hasil validasi tidak <i>valid</i> 	<ul style="list-style-type: none"> • Mengakses kontak dari <i>device</i>. • Mengambil data <i>recipient</i> dari input manual. • Mengambil 1 atau lebih data <i>recipient</i> (<i>multiple</i>) dari kontak <i>device</i> • <i>User interface</i> 	<ul style="list-style-type: none"> - mengambil data <i>recipient</i> dari input manual - mengambil 1 atau lebih data <i>recipient</i> (<i>multiple</i>) dari kontak <i>device</i> - melakukan pengiriman pesan sesuai dengan <i>schedule</i> yang diubah - menjalankan komponen <i>user interface</i>

Artifak dari *increment 3* adalah hasil analisis dan *design* untuk proses-proses pada fitur mengubah *schedule*, aplikasi dengan fitur yang sebelumnya dibuat, ditambah dengan fitur untuk mengubah *schedule* dan aplikasi dapat mengirimkan pesan sesuai dengan *schedule* yang telah dibuat maupun *schedule* yang diubah, dan hasil *testing* untuk fitur tersebut.

	Analisis	Design	Implementasi	Testing
Increment 4 [Fitur menghapus <i>schedule</i>]	Menentukan cara untuk menghapus data <i>schedule</i> dari <i>database</i> .	<ul style="list-style-type: none"> • Merancang algoritma untuk menghapus <i>schedule</i> yang dipilih dari <i>database</i> • <i>User interface</i> untuk : <ul style="list-style-type: none"> - Menampilkan konfirmasi penghapusan - Menampilkan pesan bahwa penghapusan berhasil 	<ul style="list-style-type: none"> • Menghapus <i>schedule</i> yang dipilih dari <i>database</i> • <i>User interface</i> 	<ul style="list-style-type: none"> • Melakukan test untuk : <ul style="list-style-type: none"> - menghapus <i>schedule</i> yang dipilih dari <i>database</i>. - menjalankan komponen <i>user interface</i>

Artifak dari *increment 4* adalah hasil analisis dan *design* untuk proses-proses pada fitur menghapus *schedule*, aplikasi dengan fitur yang sebelumnya dibuat, ditambah dengan fitur untuk menghapus *schedule*, dan hasil *testing* untuk fitur tersebut.

	Analisis	Design	Implementasi	Testing
Increment 5 [Fitur <i>alert</i>]	<ul style="list-style-type: none"> • Menentukan cara untuk : <ul style="list-style-type: none"> - menampilkan <i>alert</i> beberapa waktu sebelum pelaksanaan pengiriman. - menangani <i>alert</i> yang tidak mendapatkan respon dari <i>user</i> hingga waktu pengiriman <i>schedule</i> tiba 	<ul style="list-style-type: none"> • Merancang algoritma untuk : <ul style="list-style-type: none"> - menampilkan <i>alert</i> sebelum pengiriman (sesuai dengan waktu yang sudah di atur <i>user</i>) - melanjutkan pengiriman <i>schedule</i> - menghentikan pengiriman <i>schedule</i> 	<ul style="list-style-type: none"> • Menampilkan <i>alert</i> sebelum pengiriman (sesuai dengan waktu yang sudah di atur <i>user</i>) • Melanjutkan pengiriman <i>schedule</i>, baik pada saat <i>user</i> memilih untuk melanjutkan pengiriman atau pada saat <i>user</i> tidak merespon <i>alert</i> hingga waktu pengiriman tiba. 	<ul style="list-style-type: none"> • Melakukan test untuk : <ul style="list-style-type: none"> - menampilkan <i>alert</i> sebelum pengiriman pesan sesuai dengan <i>schedule</i> yang dibuat. - melanjutkan pengiriman pesan ketika <i>user</i> memilih 'yes'

	Analisis	Design	Implementasi	Testing
Increment 4 [Fitur menghapus schedule]	<ul style="list-style-type: none"> - menangani pilihan yang disediakan pada <i>alert box</i> - melanjutkan pengiriman <i>schedule</i> - menghentikan pengiriman <i>schedule</i> - menampilkan detail dari <i>schedule</i> yang akan dikirim - mengakses nada dering yang ada pada <i>device</i> - mengambil nada dering yang akan digunakan pada saat menampilkan <i>alert</i> <p>• <i>User interface</i> untuk :</p> <ul style="list-style-type: none"> - Informasi dan pilihan yang ditampilkan pada <i>alert box</i> - Menampilkan detail dari pesan yang akan dikirimkan 	<ul style="list-style-type: none"> - menangani <i>alert</i> yang tidak direspon oleh <i>user</i> hingga waktu pengirimannya tiba - mengambil detail <i>schedule</i> yang akan dikirimkan - Mengakses nada dering yang ada pada <i>device</i> - Menampilkan detail dari pesan yang akan dikirimkan <p>• <i>User interface</i></p>	<ul style="list-style-type: none"> - Menghentikan pengiriman <i>schedule</i> - Mengambil dan menampilkan detail <i>schedule</i> yang akan dikirimkan - Mengakses nada dering yang ada pada <i>device</i> - Menampilkan <i>alert</i> secara berkala (setiap 5 menit sekali) ketika <i>alert</i> tidak direspon oleh <i>user</i> <p>• <i>User interface</i></p>	<ul style="list-style-type: none"> - menghentikan/membatalkan pengiriman pesan ketika <i>user</i> memilih ‘no’ - menampilkan detail informasi dari <i>schedule</i> yang akan dibuat, ketika <i>user</i> memilih ‘see detail’ - menampilkan <i>alert</i> secara berkala jika <i>user</i> tidak memberikan konfirmasi apapun saat <i>alert</i> ditampilkan - melanjutkan pengiriman pesan ketika <i>user</i> masih tidak memberikan konfirmasi apapun, sementara waktu pengiriman <i>schedule</i> telah tiba. - menjalankan komponen <i>user interface</i>

Artifak dari *increment 5* adalah hasil analisis dan *design* untuk proses-proses pada fitur *alert*, aplikasi dengan fitur yang sebelumnya dibuat, ditambah dengan fitur *alert* yang ditampilkan sebelum pelaksanaan *schedule*, dan hasil *testing* untuk fitur tersebut.

	Analisis	Design	Implementasi	Testing
Increment 6 [Fitur notification]	<ul style="list-style-type: none"> • Menentukan cara untuk menampilkan notifikasi ketika <ul style="list-style-type: none"> - pengiriman berlangsung - pesan berhasil dikirim dan pesan telah diterima <i>recipient</i> - pesan gagal dikirim • Menentukan cara untuk menangani pengiriman ulang (<i>resend</i>) pesan yang gagal dikirim. 	<ul style="list-style-type: none"> • Membuat rancangan algoritma untuk menentukan : <ul style="list-style-type: none"> - kondisi pesan sedang dikirim - kondisi pesan berhasil dikirim dan telah diterima <i>recipient</i> - kondisi pesan gagal dikirim • Membuat rancangan algoritma untuk menangani pengiriman ulang (<i>resend</i>) 	<ul style="list-style-type: none"> • Menampilkan notifikasi ketika : <ul style="list-style-type: none"> - pengiriman berlangsung - pesan berhasil dikirim dan pesan telah diterima <i>recipient</i> - pesan gagal dikirim • Menjalankan proses <i>resend</i> untuk pesan yang gagal dikirim <p>• <i>User interface</i></p>	<ul style="list-style-type: none"> • Melakukan tes untuk : <ul style="list-style-type: none"> - menampilkan notifikasi saat pesan sedang dikirimkan - menampilkan notifikasi saat pesan berhasil dikirim dan berhasil diterima <i>recipient</i> - menampilkan notifikasi saat pesan gagal terkirim - melakukan pengiriman ulang untuk pesan yang gagal terkirim

	Analisis	Design	Implementasi	Testing
Increment 6 [Fitur notification]		<ul style="list-style-type: none"> • <i>User interface</i> untuk : <ul style="list-style-type: none"> - Menampilkan notifikasi ketika pengiriman berlangsung - Menampilkan notifikasi ketika pesan berhasil dikirim dan pesan telah diterima <i>recipient</i> - Menampilkan notifikasi ketika pesan gagal dikirim 		<ul style="list-style-type: none"> - menjalankan komponen <i>user interface</i>

Artifak dari *increment 6* adalah hasil analisis dan *design* untuk proses-proses pada fitur *notification*, aplikasi dengan fitur yang sebelumnya dibuat, ditambah dengan fitur *notification* yang ditampilkan saat pesan sedang dikirimkan dan saat pesan berhasil dikirim dan diterima *recipient* atau saat pesan gagal terkirim, dan hasil *testing* untuk fitur tersebut.

	Analisis	Design	Implementasi	Testing
Increment 7 [Fitur template]	<ul style="list-style-type: none"> • Menentukan cara untuk : <ul style="list-style-type: none"> - membuat <i>template</i> untuk <i>template</i> biasa dan <i>template</i> dengan <i>string</i> khusus - menyimpan atribut <i>template</i> ke <i>database</i> - menggunakan <i>template</i> untuk mengisi <i>content message</i> - mengambil tanggal ulang tahun atau hari jadi dari kontak yang ada pada <i>device</i> - menghapus <i>template</i> dari <i>database</i> 	<ul style="list-style-type: none"> • Membuat rancangan algoritma untuk : <ul style="list-style-type: none"> - membuat <i>template</i> baru, baik untuk <i>template</i> biasa atau <i>template</i> dengan <i>string</i> khusus - menyimpan atribut <i>template</i> baru ke <i>database</i> - mengakses kontak dari <i>device</i> - Mengambil data tanggal ulang tahun atau <i>anniversary</i> dari kontak • Mengintegrasikan <i>template</i> dengan fitur membuat <i>schedule</i> baru atau fitur mengubah <i>schedule</i> • <i>User interface</i> untuk : <ul style="list-style-type: none"> - Membuat <i>template</i> baru - Menampilkan <i>list template</i> 	<ul style="list-style-type: none"> • Membuat <i>template</i> baru, baik untuk <i>template</i> biasa atau <i>template</i> dengan <i>string</i> khusus • Menyimpan atribut <i>template</i> baru ke <i>database</i> • Mengakses kontak dari <i>device</i> • Mengambil data tanggal ulang tahun atau <i>anniversary</i> dari kontak • Mengintegrasikan <i>template</i> dengan fitur membuat <i>schedule</i> baru atau fitur mengubah <i>schedule</i> • <i>User interface</i> 	<ul style="list-style-type: none"> • Melakukan tes untuk : <ul style="list-style-type: none"> - membuat <i>template</i> baru, baik untuk <i>template</i> biasa atau <i>template</i> dengan <i>string</i> khusus - menyimpan atribut <i>template</i> baru - mengakses kontak pada <i>device</i> • mengambil data ulang tahun atau <i>anniversary</i> dari kontak <i>device</i> • membuat atau mengubah <i>schedule</i> dengan <i>content</i> berupa <i>typical message</i> • menjalankan komponen <i>user interface</i>

Artifak dari *increment 7* adalah hasil analisis dan *design* untuk proses-proses pada fitur *template*, aplikasi dengan fitur yang sebelumnya dibuat, ditambah dengan fitur *template* yang terintegrasi dengan fitur membuat *schedule* baru dan mengubah *schedule*, sehingga *template* yang sebelumnya sudah dibuat dapat digunakan ketika membuat *schedule* baru atau mengubah *schedule*, termasuk hasil *testing* untuk fitur tersebut.

	Analisis	Design	Implementasi	Testing
Increment 8 [Fitur pause schedule]	<ul style="list-style-type: none"> • Menentukan cara untuk : 	<ul style="list-style-type: none"> • Membuat rancangan algoritma untuk : <ul style="list-style-type: none"> - melakukan <i>pause schedule</i> 	<ul style="list-style-type: none"> • Menghentikan sementara pengiriman <i>schedule</i> (<i>pause</i>) 	<ul style="list-style-type: none"> • Melakukan tes untuk : <ul style="list-style-type: none"> - melakukan <i>pause schedule</i>

	Analisis	Design	Implementasi	Testing
Increment 8 [Fitur pause schedule]	<ul style="list-style-type: none"> - menghentikan sementara pengiriman <i>schedule (pause)</i> - me-resume <i>schedule</i> - menangani perubahan waktu pengiriman <i>schedule</i> yang sudah tersusun sebelumnya, dikarenakan user melakukan <i>paused</i> atau <i>resume schedule</i>. 	<ul style="list-style-type: none"> - me-resume <i>schedule</i> - melakukan validasi waktu pengiriman <i>schedule</i> di <i>resume</i> - menangani perubahan waktu pengiriman ketika <i>schedule</i> di <i>pause</i> atau di <i>resume</i> 	<ul style="list-style-type: none"> • Me-resume <i>schedule</i> • Validasi waktu pengiriman <i>schedule</i> di <i>resume</i> • Menangani perubahan waktu pengiriman ketika <i>schedule</i> di <i>pause</i> atau di <i>resume</i> 	<ul style="list-style-type: none"> - melakukan <i>resume schedule</i> - validasi waktu pengiriman <i>schedule</i> di <i>resume</i> - penanganan perubahan waktu pengiriman ketika <i>schedule</i> di <i>pause</i> atau di <i>resume</i>

Artifak dari *increment 8* adalah hasil analisis dan *design* untuk proses-proses pada fitur *pause schedule*, aplikasi dengan fitur yang sebelumnya dibuat, ditambah dengan fitur *pause schedule* dan *resume schedule* yang sebelumnya di-*pause*, dan hasil *testing* untuk fitur tersebut.

	Analisis	Design	Implementasi	Testing
Increment 9 [Fitur pengelompokan kategori pesan]	<ul style="list-style-type: none"> • menentukan cara untuk : <ul style="list-style-type: none"> - menampilkan <i>schedule</i> dan pesan yang sudah tersimpan di <i>database</i> sesuai dengan kategorinya, yaitu kategori <i>scheduled</i>, <i>sent</i>, <i>failed</i>, dan <i>paused</i>. - menangani beberapa opsi pada masing-masing kategori (integrasi dengan <i>edit</i>, <i>delete</i>, <i>resend</i>, <i>paused</i>, dan <i>resume schedule</i>) 	<ul style="list-style-type: none"> • Membuat rancangan algoritma untuk mengambil pesan dari <i>database</i> per kategorinya • <i>User interface</i> untuk : <ul style="list-style-type: none"> - Menampilkan <i>list</i> pesan dengan kategori <i>scheduled</i> - Menampilkan <i>list</i> pesan dengan kategori <i>sent</i> - Menampilkan <i>list</i> pesan dengan kategori <i>failed</i> - Menampilkan <i>list</i> pesan dengan kategori <i>paused</i> 	<ul style="list-style-type: none"> • Mengambil pesan dari <i>database</i> per kategorinya • Mengintegrasikan dengan opsi <i>pause</i>, <i>resume</i>, <i>delete</i>, <i>resend</i>, <i>new schedule</i>, dan <i>edit schedule</i>. • <i>User interface</i> 	<ul style="list-style-type: none"> • Melakukan tes untuk : <ul style="list-style-type: none"> - menampilkan pesan per kategori - menjalankan opsi-opsi yang terintegrasi, yaitu <i>pause</i>, <i>resume</i>, <i>delete</i>, <i>resend</i>, <i>new schedule</i>, dan <i>edit schedule</i> - menjalankan komponen <i>user interface</i>

Artifak dari *increment 9* adalah hasil analisis dan *design* untuk proses-proses pada fitur pengelompokan kategori, aplikasi dengan fitur yang sebelumnya dibuat, ditambah dengan fitur pengelompokan kategori pesan, yaitu untuk pesan yang masih ter-*scheduled*, pesan yang berhasil dikirim dan berhasil diterima recipient, pesan yang gagal terkirim, pesan yang sedang di-*pause* serta integrasi dengan opsi *pause*, *resume*, *delete*, *resend*, *new schedule*, dan *edit schedule*, dan hasil *testing* untuk fitur tersebut.

	Analisis	Design	Implementasi	Testing
Increment 10 [Fitur setting]	<ul style="list-style-type: none"> • Menentukan cara untuk : <ul style="list-style-type: none"> - membuat <i>default setting</i> - mengakses <i>setting</i> pada <i>device</i> 	<ul style="list-style-type: none"> • Membuat rancangan algoritma untuk : <ul style="list-style-type: none"> - membuat <i>default setting</i> 	<ul style="list-style-type: none"> • Mengakses nada dering pada <i>device</i> • Mengambil dan menggunakan <i>setting</i> yang digunakan pada <i>device</i> 	<ul style="list-style-type: none"> • Melakukan tes untuk : <ul style="list-style-type: none"> - mengakses nada dering pada <i>device</i>

	Analisis	Design	Implementasi	Testing
Increment 10 [Fitur setting]	<ul style="list-style-type: none"> - mengakses nada dering pada <i>device</i> - melakukan penyimpanan <i>setting</i> (apakah di <i>database</i> atau di <i>file system</i>). - mengintegrasikan <i>setting</i> dengan <i>alert</i> dan <i>notification</i>. 	<ul style="list-style-type: none"> - mengambil <i>setting</i> yang digunakan pada <i>device</i> - mengambil nada dering - menyimpan pengaturan <i>setting</i> pada <i>file</i> dalam <i>device system</i> maupun pada <i>database</i>. • <i>User interface</i> untuk : <ul style="list-style-type: none"> - Menampilkan menu <i>setting</i> - Menampilkan <i>list</i> nada dering yang ada pada <i>device</i> - Menampilkan pilihan waktu <i>alert</i> 	<ul style="list-style-type: none"> • Membuat <i>default setting</i> • Menyimpan <i>setting</i> yang diinput pada <i>file</i> dalam <i>device system</i> atau <i>database</i> aplikasi • Mengintegrasikan dengan fitur <i>alert</i> dan <i>notification</i> 	<ul style="list-style-type: none"> - mengambil dan menggunakan <i>setting</i> yang digunakan pada <i>device</i> - menyimpan <i>setting</i> yang diinput pada <i>file</i> dalam <i>device system</i> atau <i>database</i> aplikasi - menjalankan <i>default setting</i> - menjalankan <i>alert</i> dan <i>notification</i> sesuai dengan <i>setting</i> yang sudah disimpan - menjalankan komponen <i>user interface</i>

Artifak dari *increment 10* adalah hasil analisis dan *design* untuk proses-proses pada fitur *setting*, aplikasi dengan fitur yang sebelumnya dibuat, ditambah dengan fitur *setting* yang telah terintegrasi dengan fitur *alert* dan *notification*, dan hasil *testing* untuk fitur tersebut.

1.5 Batasan Aplikasi

Batasan untuk aplikasi SMS Scheduler ini adalah :

1. Aplikasi dibuat pada *platform* Android (versi 2.3 (*Gingerbread*) sampai dengan 4.0 (*ICS*)), Blackberry (versi OS 7), Java ME/ J2ME, dan iOS (versi OS 5 sampai dengan OS 6).
2. Panjang halaman yang disediakan untuk satu kali pengiriman maksimal 1 halaman dengan jumlah karakter 160 karakter.
3. Banyaknya kontak yang dapat digunakan untuk satu kali pengiriman maksimal 100 kontak (baik kontak dari *device* maupun yang diinput secara manual).
4. Aplikasi bersifat satu arah, artinya aplikasi tidak menangani penerimaan balasan SMS dari penerima pesan.
5. Aplikasi hanya memberikan notifikasi bahwa pesan sudah diterima pada *device recipient* dan bukan memberikan informasi bahwa pesan sudah dibaca oleh *recipient*.
6. Aplikasi hanya menampung *history* pesan sebanyak 50 pesan. Jika *history* sudah lebih dari 50 pesan, maka akan dihapus secara otomatis.

1.6 Lingkup Sistem

Dengan banyaknya *platform* untuk aplikasi *mobile* yang ada saat ini, ruang lingkup aplikasi ini pun dibatasi dengan pembuatan aplikasi yang hanya dilakukan pada *platform* Android, Blackberry, iOS dan Java ME/J2ME. Pada setiap *platformnya* pun masih dibagi lagi menjadi beberapa versi. Oleh karena itu, aplikasi ini pun hanya dibuat untuk versi, yaitu Android (versi 2.3 (*Gingerbread*) sampai dengan 4.0 (ICS)), Blackberry (versi OS 7), Java ME/ J2ME, dan iOS (versi OS 5 sampai dengan OS 6).

BAB II

Tinjauan Pustaka

Guna mendukung pembuatan aplikasi SMS *Scheduler*, ada beberapa pengetahuan pendukung dan teknologi pendukung yang diperlukan dan digunakan, beberapa pengetahuan yang diperlukan adalah pengetahuan mengenai SMS, SMS *Scheduler*, serta metode pengembangan aplikasi yang digunakan, yaitu metode *Incremental*. Dengan adanya tuntutan untuk membuat aplikasi ini di beberapa *platform*, diperlukan juga pengetahuan mengenai teknik pembuatan aplikasi, yaitu *native development*, pengetahuan mengenai *platform* yang digunakan, yaitu Android, Blackberry, Java ME/ J2ME dan iOS, serta bahasa pemrograman yang digunakan untuk membuat aplikasi di beberapa *platform* tersebut, yaitu bahasa pemrograman Java, serta *mapping tools*. Selain itu, diperlukan pula teknologi pendukung yang digunakan untuk membuat aplikasi di beberapa *platform* tersebut, yaitu Eclipse IDE, Netbeans IDE, Xcode, ADT plugin, Blackberry plugin, Android SDK, Blackberry SDK, dan iOS SDK.

2.1 Teori Pendukung

Dalam pembuatan aplikasi SMS *Schduler* ini, terdapat beberapa teori pendukung yang diperlukan dan digunakan. Teori-teori tersebut adalah mengenai *Short Message Services* (SMS), model incremental, metode *interrupt*, *predictive search*, konsep *object oriented*, *native development*. Selain itu, ada pula teori mengenai platform yang akan digunakan dalam pembuatan aplikasi (android, blackberry, Java ME, iOS), bahasa pemrograman Java, Objective C dan *mapping tools*.

2.1.1 *Short Message Services* (SMS)

Short Message Services atau SMS merupakan sebuah layanan yang banyak diaplikasikan pada sistem komunikasi nirkabel yang memungkinkan melakukan pengiriman pesan dalam bentuk *alphanumeric* antara terminal pelanggan dengan sistem eksternal (*e-mail*, *paging*, *voice mail*, dan lain-lain). Keunggulan dari teknologi SMS antara lain :

- Harganya cukup terjangkau.
- *Delivery oriented service*, maksudnya pesan selalu diusahakan dikirim ke tujuan walaupun nomor tujuan sedang tidak aktif, pesan akan disimpan di SMSC (*SMS Center*) dan akan dikirimkan kembali ketika nomor tujuan sudah aktif kembali.
- Dapat dikirim ke banyak penerima secara bersamaan.
- Pesan dapat dikirim ke berbagai jenis tujuan, seperti *e-mail*, IP dan lainnya.

Pengiriman pesan yang dilakukan oleh aplikasi SMS *Scheduler* sama seperti pengiriman SMS, yaitu ketika *user* mengirimkan pesan, pesan tersebut tidak langsung dikirim ke nomor tujuan, melainkan dikirim ke SMSC (operator telepon) terlebih dahulu, kemudian pesan baru diteruskan ke nomor tujuan. SMSC membantu *user* untuk mengetahui apakah pesan telah sampai atau gagal. Jika *device* penerima berada dalam kondisi aktif, maka pesan bisa langsung diterima dan *device* akan mengirimkan pesan konfirmasi ke SMSC yang kemudian akan diteruskan ke pengirim pesan (*user*). Namun, jika *device* penerima dalam keadaan mati atau diluar jaringan, maka pesan yang dikirim akan disimpan di SMSC sampai *period-validity* terpenuhi.

Period-validity artinya tenggang waktu yang diberikan si pengirim pesan sampai pesan dapat diterima oleh si penerima. Hal ini dapat diatur pada ponsel, mulai dari 1 jam sampai lebih dari 1 hari. Setiap detiknya, ponsel saling bertukar informasi dengan tower si pengirim paket data untuk memastikan bahwa semua berjalan sebagaimana mestinya (Arief, 2010).

SMS dikembangkan menjadi EMS (*Enhanced Message Service*). EMS memungkinkan *user* untuk mengirim pesan dengan jumlah karakter yang lebih banyak dan juga pesan non-karakter. Dengan EMS, pesan yang dikirim lebih dari 160 karakter akan dipecah menjadi beberapa pesan (masing-masing pesan 160 karakter) dan dikirimkan secara terpisah. Namun, saat diterima oleh penerima pesan, pesan akan digabungkan menjadi satu SMS lagi.

2.1.2 SMS *Scheduler*

SMS *Scheduler* merupakan aplikasi yang dibuat untuk mengirimkan pesan pada waktu yang akan datang. Dengan menggunakan aplikasi ini, *user* hanya perlu membuat jadwal pengiriman pesan dengan memasukkan nomor-nomor penerima pesan, isi pesan, kapan waktu pengirimannya dilaksanakan dan berapa frekuensi pengiriman pesan tersebut.

Nantinya, ketika waktu pengirimannya sudah sesuai dengan waktu sistem (*device*), aplikasi akan mengirimkan pesan yang sudah dijadwal tadi secara otomatis ke semua nomor penerima pesan, berdasarkan *schedule* yang telah dibuat.

Kebanyakan para *developer* aplikasi SMS *Scheduler* membuat aplikasi tersebut dengan tujuan untuk membantu orang-orang yang sibuk, memberikan ucapan selamat ulang tahun atau hari jadi (*anniversary*) di tengah malam, mengingatkan hal-hal penting kepada teman atau keluarga, dan lainnya. Dapat disimpulkan bahwa aplikasi SMS *Scheduler* ini dibuat untuk membantu menyampaikan informasi, terutama informasi-informasi penting bagi seseorang, yang harus disampaikan kepada orang lain pada waktu yang akan datang. “Aplikasi ini merupakan aplikasi yang dirancang bagi orang-orang dengan keterbatasan waktu untuk melakukan hal-hal yang sangat kecil namun penting (Derie, 2011:p.1).”

Sebetulnya tujuan dari SMS *Scheduler* sangat terkait dengan tujuan *scheduling* secara umum, yaitu untuk membantu seseorang atau lebih dalam mencapai tujuannya dengan cara yang terorganisir. Dengan SMS *Scheduler*, diharapkan dapat membantu tercapainya tujuan untuk menyampaikan suatu informasi, terutama informasi yang harus disampaikan di waktu yang akan datang, dengan cara yang terorganisir, yaitu dengan pengiriman pesan yang dijadwalkan.

2.1.3 Software Development Life Cycle (SDLC)

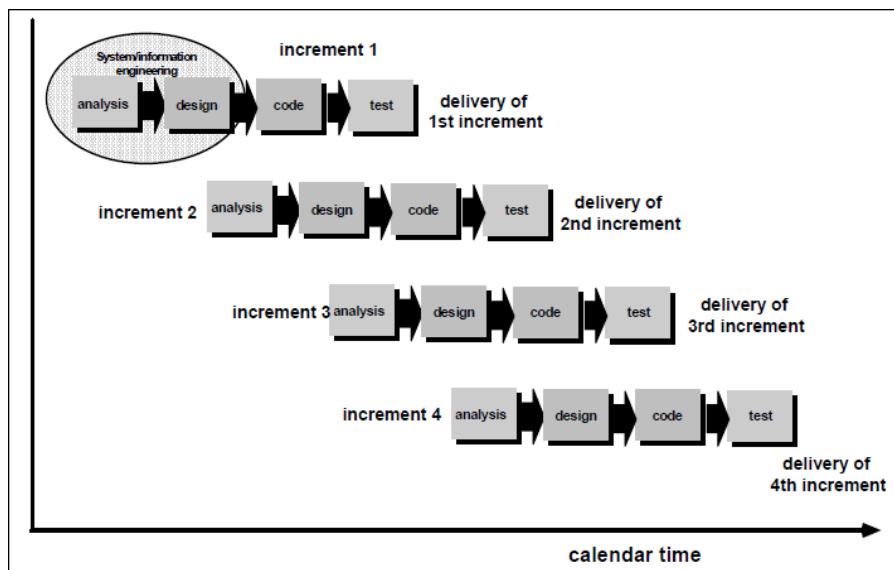
Dalam rekayasa perangkat lunak, terdapat berbagai model atau metode pengembangan yang dapat digunakan, seperti *waterfall*, *iterative*, *incremental*, *prototyping*, *spiral*, *RAD*, dan *agile*. Untuk memutuskan model atau metode pengembangan aplikasi yang akan digunakan dapat dilihat dari beberapa aspek, seperti lingkup *project*, waktu pengerjaan *project*, dan *deliverable* yang dibutuhkan. Lingkup *project* akan mempengaruhi apakah suatu *project* termasuk dalam *project* besar atau tidak dan berapa lama waktu yang dibutuhkan untuk mengerjakan *project* tersebut. Terakhir melihat dari *deliverable* yang dibutuhkan, apakah terdapat tuntutan agar dalam waktu dekat sudah harus menghasilkan suatu produk.

Jika melihat dari beberapa aspek tersebut untuk pembuatan aplikasi SMS *Scheduler*, lingkup aplikasi tidaklah terlalu besar, dikarenakan fungsi-fungsi yang akan ditangani dalam aplikasi tidak terlalu banyak.

Selain karena lingkup *project* tidak terlalu besar, waktu penggerjaan *project* pun terbatas karena telah ditentukan dalam jadwal penggerjaan aplikasi di awal penggerjaan *project*. Dengan masa penggerjaan yang tidak terlalu lama, ada pula tuntutan untuk dapat memperlihatkan *progress* pekerjaan dalam waktu dekat, yaitu setiap 2 bulan (sesuai dengan jadwal project 5 dan 6).

Dengan kondisi seperti yang disebutkan sebelumnya, pilihan model pengembangan aplikasi yang dapat dipilih antara model *incremental* dengan model *iterative-incremental*. Sebetulnya masing-masing pemodelan memiliki keuntungan dalam penggerjaan aplikasi. Namun, akhirnya diputuskan model *incremental*-lah yang digunakan, dikarenakan dengan model ini penggerjaan aplikasi dapat dilakukan secara parsial (dibagi ke dalam beberapa *increment*) dan setiap *increment* difokuskan untuk menghasilkan produk yang sudah dapat digunakan, sehingga tidak perlu menunggu penggerjaan aplikasi secara keseluruhan selesai dikerjakan untuk menghasilkan produk.

“The incremental model combines elements of the linear sequential model (applied repetitively) with the iterative philosophy of prototyping (Pressman, 2001:p.35).” Model *incremental* juga dapat diartikan sebagai model pengembangan sistem pada rekayasa perangkat lunak berdasarkan kebutuhan-kebutuhan yang dibagi menjadi beberapa bagian sehingga model pengembangannya dilakukan secara bertahap. Gambar 1 menunjukkan daur hidup dalam model *incremental*.



Gambar 1 . Model SDLC *Incremental*

Sumber : Pressman (2001, p.35)

Seperti yang digambarkan pada Gambar 1, tahap awal yang dikerjakan dalam model *incremental* adalah analisis untuk *increment* pertama, kemudian dilanjutkan dengan perancangan sampai dengan *testing* dan menghasilkan *deliverable* untuk *increment* pertama. Namun, pada saat tahap perancangan untuk *increment* pertama, tahap analisis untuk *increment* kedua sudah dapat dilakukan. Inilah yang membuat penggerjaan aplikasi bisa lebih cepat diselesaikan.

Selain itu, model *incremental* juga cocok untuk *project* yang penggerjaannya tidak harus menyelesaikan satu fase untuk melanjutkan ke fase berikutnya. Misalnya, untuk fase analisis harus dilakukan secara menyeluruh untuk satu aplikasi, barulah dapat dilanjutkan pada fase selanjutnya. Dengan model *incremental* penggerjaan *project* dapat dilakukan secara parsial, yaitu dengan membagi kebutuhan-kebutuhan aplikasi menjadi beberapa bagian penggerjaan. Oleh karena itu, pada aplikasi ini kebutuhan-kebutuhan aplikasi dibagi dalam beberapa bagian yaitu dibagi berdasarkan fitur aplikasi, dimana fitur yang dibuat pada *increment* awal adalah fitur utama atau *core product* yang sudah dapat digunakan dan dapat memberikan gambaran mengenai produk akhir.

“*When an incremental model is used, the first increment is often a core product* (Pressman, 2001:p.35). *The incremental model focuses on the delivery of an operational product with each increment. Early increments are stripped down versions of the final product, but they do provide capability that serves the user and also provide a platform for evaluation by the user* (Pressman, 2001:p.36).”

Untuk penggerjaan pada tahap *increment* berikutnya dilakukan berdasarkan prioritas dan kompleksitas pada suatu fitur. Fitur dengan prioritas dan kompleksitas tinggi akan dikerjakan terlebih dahulu pada *increment* selanjutnya. Prioritas dan kompleksitas fitur ditentukan berdasarkan *requirements* untuk masing-masing fitur yang telah terdefinisi di tahap analisis.

Karena penggerjaan aplikasi dapat dilakukan secara parsial, resiko kegagalan *project* pun dapat diminimalisir karena kesalahan sudah dapat ditemukan di awal *project*, baik dari tahap *testing* dari *increment* awal, maupun dari hasil evaluasi *user* berdasarkan produk yang sudah dihasilkan dari *increment* sebelumnya. Keuntungan lainnya yaitu model *incremental* mampu menangani perubahan *user requirement* yang dapat berubah sewaktu-waktu selama penggerjaan *project*. Sehingga jika dari produk yang sudah dihasilkan atau pada saat penggerjaan ternyata ada kebutuhan yang berubah, baik ditambahkan atau

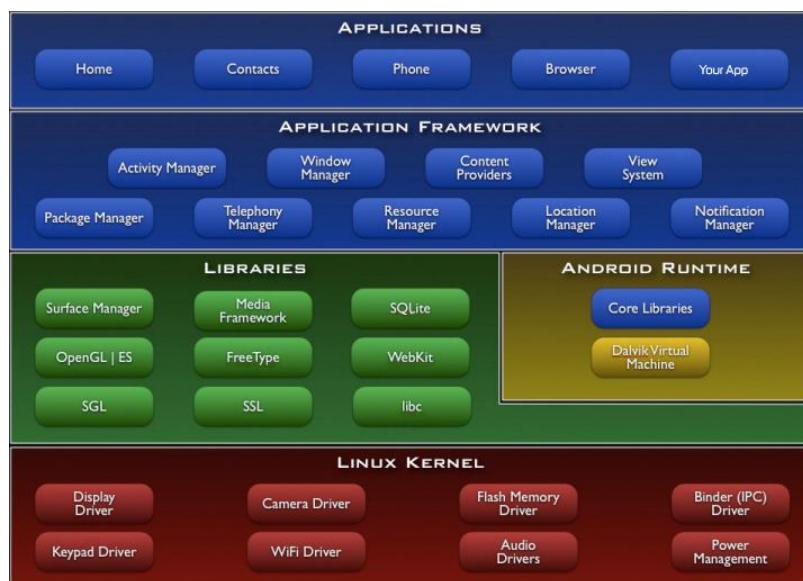
dikurangi, maka perubahan tersebut dapat ditangani pada *increment* berikutnya sampai menghasilkan produk yang sudah lengkap atau memenuhi semua kebutuhan.

Sementara jika menggunakan metode *iterative-incremental*, untuk menghasilkan suatu produk tetap harus menunggu paling tidak sampai tahap *construction*. Karena pada dua tahap sebelumnya, yaitu *inception* dan *elaboration*, pengembangan aplikasi difokuskan pada pemodelan bisnis sampai dengan perancangan aplikasi.

2.1.4 Android

Salah satu *platform* yang digunakan untuk mengembangkan aplikasi SMS *Scheduler* ini adalah Android. Android juga merupakan sistem operasi yang digunakan pada *device* Android *smartphone*. Device Android adalah *mobile device* berbasis Linux dan menggunakan bahasa pemrograman Java.

Arsitektur sistem operasi Android dapat digambarkan atau diibaratkan sebagai tumpukan *software* dengan beberapa *layer* yang berbeda, dimana masing-masing *layer* tersebut merupakan kumpulan dari beberapa komponen program.



Gambar 2 . Arsitektur Android

Sumber : Android App Market (<http://www.android-app-market.com/android-architecture.html>)

Masing-masing layer pada gambar 2 dijelaskan sebagai berikut.:

- **Linux Kernel**

Android mengandalkan Linux untuk layanan utamanya, yaitu memory management, security setting, power management, dan beberapa driver hardware (camera, display, keypad, wifi, flash memory, audio, IPC).

- **Libraries and Android Runtime**

Lapisan selanjutnya adalah Libraries dan Android runtime. Layer ini memungkinkan *device* untuk menangani berbagai jenis data yang berbeda. Libraries ditulis dalam bahasa C atau C++ dan spesifik untuk hardware tertentu. Beberapa native libraries penting dalam Libraries Android, yaitu surface manager, media framework, SQLite (database engine yang digunakan dalam Android untuk penyimpanan data), WebKit, dan OpenGL.

Sementara itu, Android Runtime terdiri dari Dalvik Virtual Manager (DVM) dan Core Java Libraries. DVM merupakan Java Virtual Machine yang digunakan untuk pada device android untuk menjalankan aplikasi dan mengoptimalkan *low processing power* dan *low memory environments* (optimalkan telepon selular). Sedangkan Core Java Libraries berisi *library* dasar yang menyediakan sebagian besar fungsi-fungsi yang ada pada Java SE *libraries*.

- **Application Framework**

Lapisan yang memberikan akses ke kerangka API yang digunakan oleh aplikasi inti dan memungkinkan pengembang untuk menggunakan komponen API tersebut. Application Framework berisi serangkaian *tool* dasar seperti alokasi *resource* pada *smartphone*, aplikasi telepon, pergantian antar proses atau program, dan pelacakan lokasi fisik telepon. Komponen-komponen pada Application Framework antara lain :

- *Activity Manager* : mengatur siklus hidup dari aplikasi.
- *Content Providers* : mengatur data sharing antar aplikasi sehingga memungkinkan aplikasi untuk mengakses data, misalnya kontak.
- *Notification Manager* : memungkinkan semua aplikasi untuk menampilkan *custom alert* pada status bar.
- *Resource Manager* : mengatur berbagai jenis *resource* yang dibutuhkan aplikasi.

- Application

Merupakan *layer* teratas pada arsitektur Android dan merupakan lapisan yang paling sering diakses oleh *user* melalui *user interface*. Pada *layer* ini terdapat fungsi-fungsi dasar *smartphone*, misalnya *dialer*, SMS *client app*, *web browser*, dan *contact manager*. Aplikasi berbasis Android dibuat dengan bahasa pemrograman Java. Selain itu, diperlukan pula Android SDK yang akan melakukan *compile* dan *convert* aplikasi menjadi Android Package (apk). Instalasi apk akan dilakukan oleh keseluruhan kemampuan Android-*Device* menjadi aplikasi berbasis Android. Struktur aplikasi berbasis Android sebagai berikut :

- *Activities* : direpresentasikan sebagai tempat menyimpan UI untuk kebutuhan aplikasi terhadap suatu form yang memiliki layout tertentu. Suatu aplikasi dapat memiliki banyak activities dan tidak diperkenankan jika suatu aplikasi tidak memiliki activity.
- *Services* : berjalan dalam suatu background untuk melakukan suatu operasi yang relatif panjang waktunya. Service tidak diperbolehkan untuk menyediakan UI.
- *Content Providers* : menyediakan hak akses ke dalam database SQLite, baik itu untuk local akses atau web.
- *Broadcast Receivers* : menyediakan pesan atau data untuk kebutuhan aplikasi lain. Misalnya suatu aplikasi memakan cukup banyak energi baterai sehingga baterai cepat habis, maka aplikasi yang sedang berjalan akan melakukan broadcast receiver untuk menginformasikan bahwa baterai akan habis atau kondisi baterai saat ini sudah mencapai 30%.
- *Intent* : merupakan pemanggil *activity*, baik di awal aplikasi berjalan maupun pada saat aplikasi sedang berjalan.

Android memungkinkan *developer* untuk mengembangkan aplikasi sendiri. Disamping itu, Android juga menyediakan *tools* dan APIs yang dibutuhkan selama pembuatan aplikasi. Sejak pertama dirilis, telah dilakukan beberapa kali pembaharuan pada sistem operasi Android, yaitu :

1. Android versi 1.6
2. Android versi 1.5 (*Cupcake*)
3. Android versi 1.6 (*Donut*)
4. Android versi 2.0/ 2.1 (*Eclair*)
5. Android versi 2.2 (*Froyo* : *Frozen Yoghurt*)

6. Android versi 2.3 (*Gingerbread*)
7. Android versi 3.0/3.1 (*Honeycomb*)
8. Android versi 4.0 (*ICS : Ice Cream Sandwich*)
9. Android versi 5.0 (*Jellybean*)

2.1.5 Blackberry

Salah satu *platform* yang digunakan untuk mengembangkan aplikasi SMS *Scheduler* ini adalah Blacberry. Blackberry juga merupakan sistem operasi *mobile* yang dikembangkan oleh *Research In Motion* (RIM) untuk digunakan pada *device* Blackberry *smartphone*. Sistem operasi blackberry memiliki kelebihan dalam hal keamanan, kemampuan mutitasking, dan interoperabilitas dengan infrastruktur email perusahaan. Selain itu, sistem operasi blackberry dirancang untuk penggunaan perangkat input seperti *trackwheel*, *trackball*, dan *trackpad*. Kelebihan lain yang dimiliki sistem operasi ini adalah adanya fitur BlackBerry Messenger (BBM) yang memungkinkan pengguna untuk berkomunikasi melalui pesan instan secara *realtime*.

Sejak pertama dirilis, Blackberry OS telah mendapatkan beberapa kali pembaharuan, berikut ini versi-versi Blackberry OS :

1. Untuk *device* Blackberry seri 8000
 - a. Blackberry OS versi 4.5
 - b. Blackberry OS versi 4.6
 - c. Blackberry OS versi 5.0
2. Untuk *device* Blackberry seri 9000
 - a. Blackberry OS versi 5.0
 - b. Blackberry OS versi 6.0
 - c. Blackberry OS versi 7.0
 - d. Blackberry OS versi 7.1
 - e. Blackberry OS versi 10

2.1.6 iOS

Salah satu *platform* yang digunakan untuk mengembangkan aplikasi SMS *Scheduler* ini adalah iOS. Sama seperti Android dan Blackberry, iOS juga merupakan sistem operasi untuk perangkat *mobile* yang dikembangkan dan didistribusikan oleh Apple Inc untuk digunakan pada *mobile device* milik Apple, yaitu iPhone, iPad, dan iPod. IOS hanya dapat digunakan pada perangkat Apple, karena Apple tidak melisensikan iOS untuk diinstal pada perangkat non-Apple.

iOS dibangun dengan menggunakan bahasa pemrograman Objective-C dan kompiler C/C++ untuk performa *native*. Arsitektur iOS terdiri dari empat lapisan abstraksi, yaitu *Core OS*, *Core Services*, *Media* dan *Cocoa Touch*. *User interface* pada iOS merupakan kombinasi antara *Cocoa Touch* dan *Interface Builder developer tools*. Versi iOS yang pertama adalah iOS 1.0 dan yang terbaru adalah iOS 6.0. Namun, aplikasi SMS *Scheduler* ini dibuat untuk iOS versi 4 sampai versi 6 saja.

2.1.7 Java ME/ J2ME

Salah satu *platform* yang digunakan untuk mengembangkan aplikasi SMS *Scheduler* ini adalah Java ME/ J2ME. *Java Platform Micro Edition* (Java ME) atau J2ME merupakan salah satu *platform* Java yang dirancang untuk aplikasi *mobile* dan *embedded devices*. Penggunaan Java ME digunakan untuk melakukan pengembangan aplikasi di lingkungan yang lebih kecil (memori yang terbatas, layar yang kecil dan kapasitas daya yang minim) dibandingkan dengan lingkup PC, yaitu *handphone*, PDA dan *mobile device* lainnya.

Java ME terdiri dari elemen-elemen berikut :

1. Konfigurasi (*configuration*), menyediakan kumpulan *library* utama dan kemampuan *virtual machine* untuk berbagai perangkat. Dua konfigurasi utama pada J2ME, yaitu konfigurasi untuk *small mobile device* yang disebut *Connected Limited Device Configuration* (CLDC) dan konfigurasi untuk *device* yang memiliki kemampuan lebih seperti *smartphone* yang disebut *Connected Device Configuration* (CDC).
2. Profil (*profile*) adalah kumpulan API yang mendukung berbagai *small mobile device*. Sebuah profile berdiri di atas sebuah *configuration*. Profile yang paling dikenal adalah MIDP (*Mobile Information Device Profile*).
 - MIDP 1.0

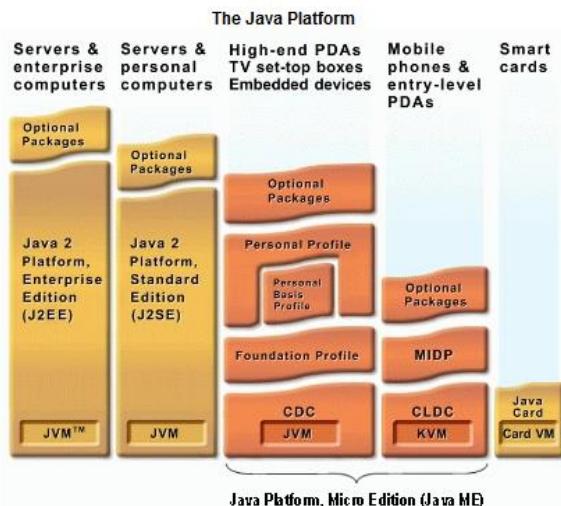
MIDP ini hampir didukung oleh semua ponsel J2ME. API yang ada pada MIDP ini adalah API standar atau sangat general sehingga dapat didukung oleh semua ponsel J2ME. Keterbatasan API standar MIDP 1.0 ini antara lain tidak ada API untuk mengakses address book dan SMS user (termasuk mengirim SMS). Batasan minimum device yang didukung oleh profile ini adalah device dengan display monokrom atau berwarna (minimal 96x54 piksel), input touch screen, keypad ITU,

atau keyboard, koneksi wireless bandwidth terbatas (tidak selalu online), dan batasan memori minimum CLDC (memori non volatile 128 Kb, memori volatile 8 kb untuk data persisten, dan 32 Kb untuk runtime Java)

- MIDP 2.0

MIDP ini merupakan MIDP 1.0 yang ditambah dengan API untuk multimedia, misalnya akses ke kamera dan perekam video, mendukung berbagai format audio dan video, memiliki batasan minimum hardware yang lebih tinggi. Selain itu, komponen user interface yang tersedia sudah mendukung penggunaan form, list, dan textbox.

Gambaran umum komponen teknologi Java ME dan bagaimana keterkaitannya dengan teknologi Java yang lain dapat dilihat pada gambar dibawah ini.

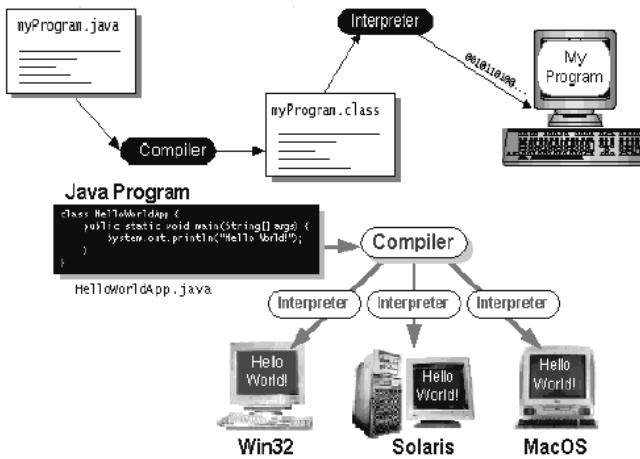


Gambar 3 . Java Platform, Micro Edition (Java ME)

Sumber : Oracle (<http://www.oracle.com/technetwork/java/javame/about-java-me-395899.html>)

2.1.8 Bahasa Pemrograman Java

Untuk membuat aplikasi pada *platform* android, blackberry, dan java, bahasa pemrograman yang digunakan adalah Java. Java merupakan bahasa pemrograman berorientasi objek (OOP) dan dapat dijalankan pada beberapa *platform* sistem operasi. Dalam bahasa pemrograman Java terdapat penyederhanaan untuk menghilangkan fitur bahasa yang menyebabkan kesalahan umum dalam pemrograman. Cara kerja Java dapat digambarkan sebagai berikut.



Gambar 4 . Cara kerja Java

Sumber : Noviyanto (p. 1)

Source code Java (file dengan ekstensi .java) dikompilir kedalam format yang disebut *bytecode* (file dengan ekstensi *class*) yang kemudian dieksekusi oleh Java *interpreter* untuk kemudian hasilnya dimunculkan ke layar.

Karakteristik bahasa pemrograman Java antara lain sederhana, berorientasi objek, terdistribusi, interpreted, robust, aman, architecture neutral, portable, performance, multithreaded dan dinamis.

Kelebihan dari bahasa pemrograman Java antara lain :

1. *Multiplatform* atau *open system*
2. Mendukung OOP (*Object Oriented Programming*)
3. Mengadopsi gaya C++
4. Mudah saat mendistribusikan *software*

2.1.9 Predictive Search

Predictive search adalah teknologi input yang digunakan dimana satu huruf atau angka merepresentasikan banyak kata. Setiap huruf atau kata yang diinputkan akan menghasilkan prediksi/ kemungkinan terhadap kumpulan kata yang mengandung huruf atau angka tersebut. *Predictive search* digunakan pada fitur sistem *filtering contact*. Dengan *predictive search*, user hanya perlu menginputkan *keyword*-nya saja sehingga user tidak harus menuliskan nama kontak atau nomor kontak secara jelas untuk melakukan pencarian.

2.1.10 Native Development

Dengan adanya keinginan untuk membuat aplikasi ini di beberapa *platform*, maka terdapat beberapa teknik pengembangan aplikasi yang dapat digunakan untuk aplikasi ini, diantaranya adalah *native development* dan *hybrid development*.

Native development merupakan salah satu teknik yang digunakan untuk membuat atau mengembangkan suatu aplikasi, terutama untuk aplikasi *mobile*. Selain *native*, ada pula web dan *hybrid development*. Dengan *native development*, sebuah aplikasi dibuat dengan menggunakan pemrograman aslinya. Maksudnya, jika sebuah aplikasi dibuat untuk beberapa *platform* yang berbeda, maka pemrograman aplikasi dilakukan “satu per satu”, dan bisa jadi dengan menggunakan bahasa pemrograman yang berbeda, sesuai dengan masing-masing *platform*-nya. Berbeda dengan *native*, jika teknik yang digunakan adalah *hybrid development*, maka pemrograman aplikasi hanya dilakukan satu kali, dengan satu bahasa pemrograman saja. Namun, antara *native* dan *hybrid development* ini memiliki kelebihan dan kekurangannya masing-masing.

Di dunia industri, *native development* memang tidak dianjurkan karena penggunaan teknik ini akan memakan banyak waktu dan biaya. Berbeda dengan *hybrid*, dimana dengan kelebihannya membuat satu *code* untuk beberapa *platform* sekaligus tentu membuat biaya dan waktu yang diperlukan pun lebih sedikit. Namun, untuk kasus pembuatan aplikasi pada Tugas Akhir ini, penggunaan *native development* justru dianjurkan. Alasannya adalah *developer* akan mendapatkan lebih banyak ilmu dengan mempelajari lebih dari satu *platform*. Selain itu, dengan menggunakan *native development*, terdapat beberapa fungsi pada *device* yang dapat diakses, seperti fungsi kontak, kalender, *messaging*, *alarm*, dll, dimana fungsi-fungsi tersebut sangat diperlukan dalam pembuatan aplikasi SMS *Scheduler* ini.

2.1.11 Mapping Tools (Pemetaan Tools)

Dikarenakan aplikasi ini dibuat untuk beberapa *platform*, maka *tools* yang digunakan pun sudah pasti sangat beragam. Jika *developer* harus melakukan instalasi semua *tools* untuk semua *platform*, sudah pasti kurang efektif, karena selain membutuhkan memori yang cukup besar, setiap kali akan membuat satu aplikasi untuk *platform* yang

berbeda, maka harus berpindah tools, padahal beberapa *tools* yang digunakan untuk mendevelope pada beberapa *platform* memiliki kesamaan.

Oleh karena itu, *mapping tools* digunakan untuk memudahkan *developer* untuk mengetahui *tools-tools* apa saja yang dapat digunakan dalam membuat suatu aplikasi, terutama jika pengembangan aplikasi tersebut akan dibuat pada beberapa *platform*, dimana setiap *platform* pasti memiliki banyak *tools* yang dapat digunakan.

Selain itu, dengan menggunakan *mapping tools*, dapat diketahui kesamaan yang ada diantara satu *tool* pengembangan dengan *tool* pengembangan lainnya. Dengan demikian, penggunaan *tools* dapat diminimalisir. Karena, jika telah diketahui *tools* untuk *platform* mana saja yang memiliki kesamaan, maka *tools* tersebutlah yang akan digunakan dan *developer* tidak harus melakukan instalasi semua *tools* untuk semua *platform*. *Mapping tools* dapat dilihat pada Lampiran A.

2.1.12 Teknologi Pendukung

Dengan adanya tuntutan untuk membuat aplikasi pada beberapa *platform*, diperlukan beberapa teknologi untuk mendukung pembuatan aplikasi. Teknologi tersebut adalah Eclipse IDE, Netbeans IDE, Xcode IDE, Android SDK, Blackberry SDK dan iOS SDK.

2.1.13 Eclipse IDE

Untuk membuat aplikasi SMS *Scheduler* pada *platform* android dan blackberry OS, *tools* yang digunakannya adalah Eclipse IDE. Aplikasi ini merupakan salah satu Java IDE (*Integrated Development Environment*) yang digunakan untuk melakukan pengembangan perangkat lunak dan bersifat *open source*. Beberapa kelebihan Eclipse IDE antara lain :

1. Dapat dikembangkan oleh pengguna dengan *plug-in*, misalnya plugin untuk android dan blackberry yang dapat digunakan pada pengembangan *mobile application* untuk Android dan Blackberry.
2. Dapat beroperasi diatas beberapa sistem operasi, seperti Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
3. Mendukung pengembangan aplikasi berbasis bahasa pemrograman selain Java, seperti C/C++, Cobol, Python, Perl, PHP dan lainnya walaupun sebetulnya Eclipse IDE dikembangkan dengan menggunakan bahasa pemrograman Java.

4. Dapat digunakan juga untuk aktivitas dalam *software development lifecycle*, seperti dokumentasi, testing, pengembangan web dan lainnya.

Sejak tahun 2006, Eclipse *Foundation* melakukan peluncuran secara rutin yang dikenal dengan *Simultaneous Release* dan telah menghasilkan beberapa versi, yaitu :

- | | |
|---------------------|---------------------|
| 1. Eclipse 3.0 | 5. Eclipse Ganymede |
| 2. Eclipse 3.1 | 6. Eclipse Galileo |
| 3. Eclipse Callisto | 7. Eclipse Helios |
| 4. Eclipse Europa | 8. Eclipse Juno |

Namun, versi *eclipse* yang mendukung untuk pembuatan aplikasi android minimal menggunakan Eclipse Helios. Dan untuk pembuatan aplikasi blackberry minimal menggunakan Eclipse Indigo.

2.1.14 Netbeans IDE

Untuk membuat aplikasi SMS *Scheduler* pada *platform Java ME/ J2ME*, *tools* yang digunakannya adalah Netbeans IDE. Netbeans IDE adalah IDE *open source* yang digunakan untuk membuat *software* aplikasi komputer maupun *mobile*. Netbeans IDE mendukung pengembangan semua tipe aplikasi JAV, seperti J2SE, web, EJB dan aplikasi *mobile* (Java ME/ J2ME). Selain untuk membuat aplikasi, Netbeans juga digunakan untuk melakukan *testing*. Netbeans IDE ditulis dalam Java, namun Netbeans tetap dapat mendukung bahasa pemrograman lainnya. Fitur lain yang dimiliki Netbeans IDE adalah sistem proyek berbasis Ant, *versioning control* dan *refactoring*.

Berikut ini adalah beberapa versi dari Netbeans IDE yang ada hingga saat ini antara lain :

1. Netbeans IDE versi 5 (5.0, 5.1, 5.5, 5.5.1)
2. Netbeans IDE versi 6 (6.0.1, 6.5, 6.5.1, 6.6, 6.7, 6.7.1, 6.8, 6.9, 6.9.1)
3. Netbeans IDE versi 7 (7.0, 7.0.1, 7.1, 7.1.1, 7.1.2, 7.2, 7.3)

Untuk membuat aplikasi Java *mobile*, Netbeans IDE yang digunakan minimal adalah Netbeans IDE versi 6.

2.1.15 Xcode IDE

Untuk membuat aplikasi SMS *Scheduler* pada *platform iOS*, *tools* yang digunakannya adalah Xcode. Xcode merupakan *suite of tools* yang dikembangkan oleh

Apple untuk mengembangkan *software* yang dibuat pada platform Mac OS X dan iOS. Xcode hanya dapat digunakan dengan menggunakan Mac OS X, artinya Xcode tidak dapat diinstal pada sistem operasi seperti Microsoft Windows, Linux, dll. Didalam Xcode sudah termasuk instrumen *analysis tool*, iOS Simulator, dan Mac OS X versi sebelumnya beserta iOS SDK. Xcode IDE menganalisis rincian *project* untuk mengidentifikasi kesalahan antara sintaks dan logika, bahkan dapat membantu untuk memperbaiki kode.

2.1.16 ADT *Plugin*

Salah satu *tools* yang digunakan untuk membuat aplikasi pada *platform* Android adalah ADT (*Android Developer Tools*) plugin. ADT merupakan *plugin* untuk Eclipse yang menyediakan sekumpulan tools yang terintegrasi dengan Eclipse IDE. ADT plugin dapat digunakan developer untuk mengakses berbagai fitur yang membantu dalam pengembangan atau pembuatan aplikasi Android. ADT juga menyediakan akses GUI ke berbagai command line SDK tools seperti UI design tools untuk rapid prototyping, perancangan dan pembuatan user interface aplikasi. Beberapa fitur penting dari Eclipse dan ADT antara lain :

- Meng-integrasi antara Android project creation, building, packaging, installation and debugging
- Integrasi dengan SDK tools
- Editor bahasa pemrograman Java dan XML
- Meng-integrasi dokumen untuk Android framework APIs.

2.1.17 Blackberry *Plugin*

Salah satu *tools* yang digunakan untuk membuat aplikasi pada *platform* Blackberry adalah Blackberry plugin. Sama seperti ADT, Blackberry plugin ini pun merupakan sekumpulan tools yang terintegrasi dengan IDE yang digunakan developer, baik itu Eclipse maupun Netbeans IDE. Blackberry plugin memberikan akses bagi developer untuk membuat/ membangun, melakukan pengujian dan memperbaiki kesalahan (debug) aplikasi Blackberry dengan menggunakan Eclipse IDE atau Netbeans IDE.

2.1.18 Android SDK

Tools lain yang digunakan untuk membuat aplikasi SMS *Scheduler* di *platform* android adalah Android SDK. Android SDK (*Software Development Kit*) adalah *tools API (Application Programming Interfaces)* yang diperlukan untuk mengembangkan aplikasi pada *platform* android. Android SDK ini merupakan *software* yang berguna untuk mensimulasikan OS Android pada PC. Selain itu, Android SDK juga dapat digunakan untuk mencoba aplikasi-aplikasi yang disediakan oleh para *developer*. Android SDK disediakan sesuai dengan versi dari Android OS, yaitu Android SDK 1.6 sampai dengan Android SDK 4.2.

2.1.19 Blackberry SDK

Tools lain yang digunakan untuk membuat aplikasi SMS *Scheduler* di *platform* Blackberry OS adalah Blackberry SDK. Blackberry SDK sama seperti Android SDK, yaitu tools API yang diperlukan untuk mengembangkan aplikasi pada *platform* Blackberry OS. Blackberry SDK ini juga merupakan *software* yang digunakan untuk mensimulasikan OS Blackberry pada PC. Sehingga untuk mencoba hasil aplikasi yang telah dibuat dapat dijalankan pada simulator Blackberry terlebih dahulu.

2.1.20 iOS SDK

Tools lain yang digunakan untuk membuat aplikasi SMS *Scheduler* di *platform* iOS adalah iOS SDK. Sama seperti SDK untuk android dan blackberry, iOS SDK juga digunakan untuk dapat mensimulasikan iOS, sehingga aplikasi-aplikasi yang telah dibuat dapat dijalankan pada simulator yang disediakan pada iOS SDK. iOS SDK terdiri dari *development tools* yaitu, Xcode, iOS Simulator, *Instrument* dan *Interface Builder*.

BAB III

Hasil Analisis dan Pembahasan

Pada tahap ini akan dilakukan analisis terhadap aplikasi yang ada saat ini untuk mengetahui fitur atau fungsi yang ada dan mengetahui perilaku aplikasi. Dari hasil analisis tersebut, akan dilakukan evaluasi untuk mengetahui fitur atau fungsi apa yang masih bisa dikembangkan. Diharapkan dari kegiatan tersebut didapatkan kebutuhan-kebutuhan yang diperlukan untuk melakukan pengembangan aplikasi (fungsi atau fitur apa yang akan dikembangkan) dan perilaku dari aplikasi yang akan dibuat. Untuk mencapai hal tersebut dilakukan :

- Identifikasi pada beberapa aplikasi yang ada saat ini untuk mendapatkan fungsi atau fitur yang ada, termasuk fungsi atau fitur tambahan yang ada pada masing-masing aplikasi. Dari fungsi atau fitur tersebut, didapatkan perilaku aplikasi yang digambarkan dengan menggunakan *tools use case diagram*.
- Evaluasi untuk mendapatkan kebutuhan-kebutuhan yang diperlukan untuk mengembangkan aplikasi.
- Survei pada para pengguna *mobile device* dan pengguna SMS untuk memastikan apakah kebutuhan yang ditemukan benar-benar dibutuhkan oleh *user*. Dari kebutuhan-kebutuhan tersebut, didapatkan fungsi atau fitur untuk aplikasi yang akan dibuat, serta perilaku aplikasi yang akan dibuat (perilaku aplikasi yang akan dibuat digambarkan dengan menggunakan *tools use case diagram*).
- Analisis terhadap teknologi-teknologi dan teknik pembuatan aplikasi yang dapat digunakan ketika melakukan pengembangan atau pembuatan aplikasi.

3.1 Analisis dan Evaluasi Sistem Saat Ini

Analisis sistem yang saat ini sedang berjalan (*current system*) dilakukan pada tahap analisis *di-increment* pertama. Setelah dilakukan analisis terhadap *current system*, selanjutnya dilakukan analisis aplikasi yang akan dibuat serta analisis *requirement* yang dilakukan juga pada tahap analisis *di-increment* pertama.

3.1.1 Analisis Sistem Saat Ini

Dikarenakan aplikasi-aplikasi SMS *Scheduler* yang ada saat ini bukanlah aplikasi yang *open source*, sehingga fungsi utama maupun fungsi pendukung aplikasi tidak dapat dipelajari dari *script*-nya. Oleh karena itu, analisis sistem yang ada saat ini dilakukan dengan melakukan identifikasi fungsi-fungsi atau fitur-fitur yang tersedia, baik fungsi utama maupun fungsi pendukung aplikasi. Kemudian, dilakukan identifikasi perilaku yang ada dari fitur-fitur tersebut dilihat berdasarkan visualisasinya (yang terlihat), karena teknis dari proses-proses yang ada pada aplikasi tersebut tidak dapat dilihat lebih mendalam.

Aplikasi-aplikasi SMS *Scheduler* yang dianalisis, yaitu :

1. SMS *Scheduler* (*Author* : Lylynx.com)
2. SMS *Scheduler* (*Author* : Cuunbonandroid)
3. SMS *Scheduler* (*Author* : ZYXSolutions)
4. Auto SMS (*Author* : Thein Min Naing)
5. SMS *Scheduler* (*Author* : DiorDNA Apps)
6. SMS *Scheduler* (*Author* : Globalapps R)
7. SMS *Scheduler* (*Author* : Vinsol)
8. SMS *Scheduler* (*Author* : Sharperstudio)
9. SMS *Scheduler* (*Author* : Liuge, Good App)
10. Holo SMS *Scheduler* (*Author* : Casey Vockrodt, Destruction Labs)

Berikut ini akan dibahas beberapa aplikasi SMS *Scheduler* yang ada saat ini secara detail. Untuk aplikasi yang tidak dijelaskan, artinya aplikasi tersebut memiliki kemiripan atau kesamaan dengan aplikasi yang dibahas.

3.1.1.1 SMS *Scheduler* (*Author* : Lylynx.com)

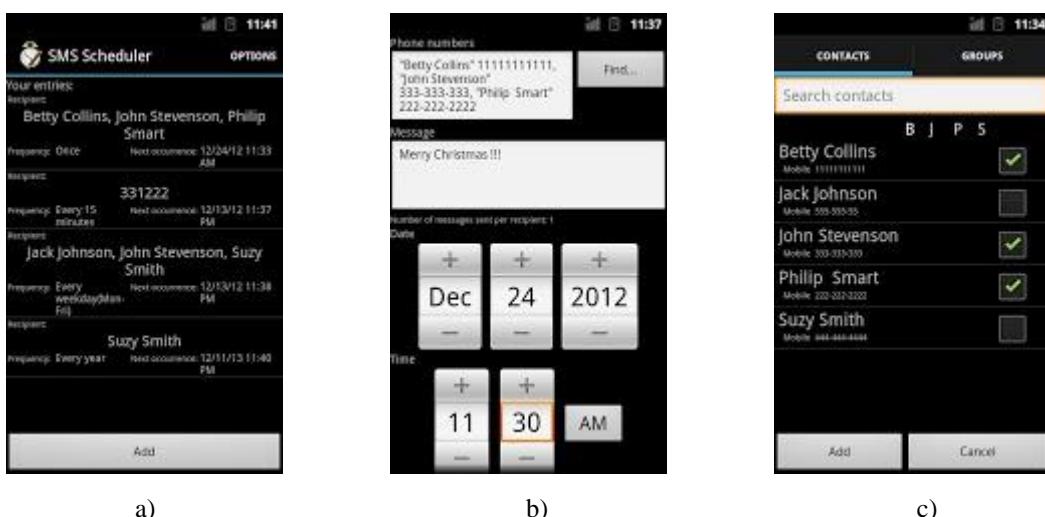
Fitur-fitur utama dari aplikasi SMS *Scheduler* ini antara lain :

1. Nomor penerima pesan dapat ditulis secara manual oleh *user* maupun dipilih dari list kontak dari *email* yang tersinkronisasi dengan *device*.
2. Penerima pesan dapat dipilih lebih dari satu (*multiple recipients*).
3. Terdapat fitur untuk mengatur frekuensi pengiriman pesan, diantaranya untuk satu kali pengiriman, setiap 5 menit, setiap 15 menit, setiap jam, sampai setiap tahun.

4. Terdapat notifikasi pada *status bar* ketika pesan dikirim dan kemudian akan ditampilkan berita pengirimannya (*delivery reports*).
5. Terdapat *history* untuk pesan yang sudah dikirim.

Selain fitur-fitur diatas, terdapat beberapa hal lain yang dilakukan aplikasi ini, yaitu :

1. Setiap jadwal yang sudah dibuat *user* akan ditampilkan dalam bentuk list pada halaman awal aplikasi.
2. Setiap jadwal yang sudah dibuat *user* dapat di edit maupun dihapus.
3. Saat memilih *recipient*, *user* tidak hanya dapat memilih dari list kontak, tetapi dapat pula mengambil dari grup, baik itu grup pada *device* maupun dari grup pada *email* yang tersinkronisasi dengan *device*.
4. *History* tidak dapat dilihat secara *detail* (hanya ada *recipient*, *event type*, *schedule time*, *delivery time*, dan *status*), sementara isi pesannya apa dan siapa saja *recipient*-nya (jika *recipient* lebih dari satu) tidak dapat dilihat.
5. Terdapat menu *preference* untuk mengatur apakah *sending notifications* akan ditampilkan atau tidak, *delivery notifications* akan ditampilkan atau tidak, *conversation threads*, dan *date time pattern*.



Gambar 5 . Screenshot aplikasi SMS Scheduler (Author : Lylynx.com)

- a) halaman awal aplikasi, b) halaman untuk membuat jadwal pengiriman pesan, dan c) halaman untuk memilih *recipient*.

3.1.1.2 Auto SMS (*Author : Thein Min Naing*)

Aplikasi Auto SMS ini berbeda dengan aplikasi SMS *Scheduler* yang lainnya, karena aplikasi ini tidak hanya menangani SMS *Scheduler* saja, tetapi ada beberapa fitur lain seperti SMS *Auto Reply*, SMS *Reader*, Instant SMS, dan Forward SMS.

1. Fitur SMS *Auto Reply*

- a. Secara otomatis membalas dengan SMS, dimana pesannya sudah ditentukan atau di-*setting*, ketika ada SMS masuk atau panggilan masuk yang tidak sempat untuk dibalas/ dijawab (misalnya sedang meeting, sedang tidur, sedang sibuk, dll).
- b. *Reply Message* (dapat merubah *auto-response* SMS *message* untuk *missed calls* dan SMS).
- c. *Widget* (widget untuk menghidupkan/ mematikan *auto-response*).
- d. Auto On/Off (dapat mengatur durasi menghidupkan *auto-response*, sehingga tidak perlu khawatir jika lupa untuk mematikannya).
- e. *Silent Mode* (dapat mengatur Ringer Mode menjadi silent mode selama aplikasi dihidupkan dan mengembalikannya ke Ringer Mode secara otomatis setelah aplikasi dimatikan).
- f. *Battery Efficient* (Auto SMS tidak akan “memakan” baterai *device* ketika aplikasi sedang dimatikan atau ketika tidak ada *schedule* pengiriman pesan, sehingga akan membantu untuk menghemat baterai).

2. SCHEDULE SMS

User dapat membuat jadwal untuk mengirimkan SMS di masa yang akan datang, misalnya pesan yang harus dikirimkan untuk acara penting atau untuk orang yang disayangi.

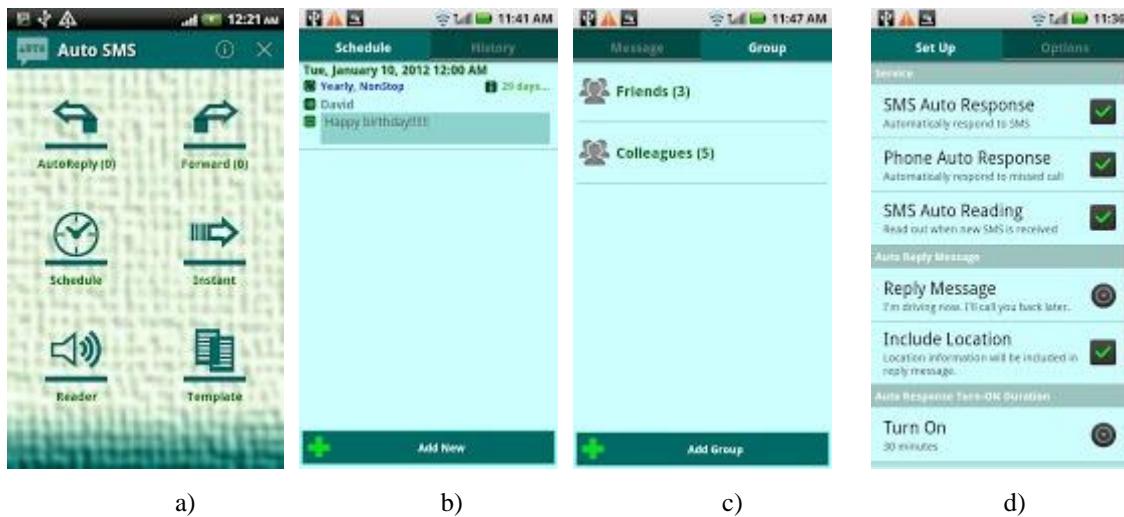
3. SMS *READER*

Fitur ini sangat berguna ketika *user* sedang tidak ingin atau tidak dapat melihat layar dan membaca SMS, misalnya ketika sedang berkendara, sehingga *user* hanya tinggal mendengarkan aplikasi ini membacakan pesannya.

4. FORWARD SMS

Fitur ini akan meneruskan SMS yang diterima ke nomor yang sudah ditentukan secara otomatis. Fitur ini dapat digunakan untuk kegiatan bisnis seperti berbagi informasi dengan anggota tim.

Selain itu, aplikasi ini juga memiliki beberapa tema yang dapat digunakan oleh *user* untuk membuat tampilannya lebih menarik.



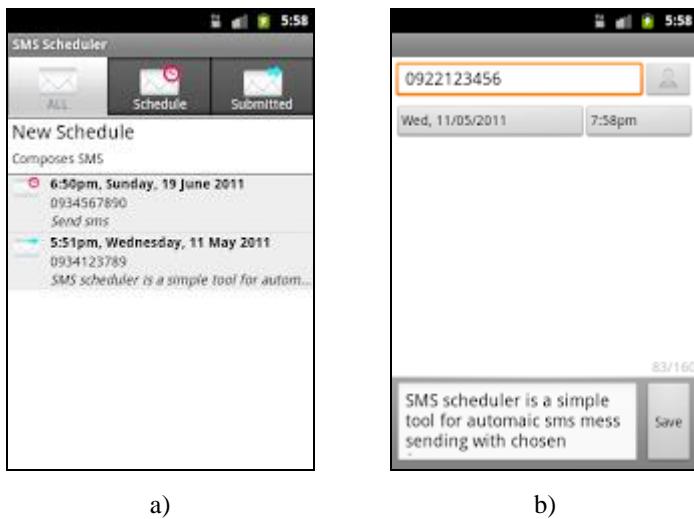
Gambar 6 . Screenshot aplikasi Auto SMS (*Author* : Thein Min Naing)

- a) halaman awal aplikasi, b) list jadwal yang dibuat, c) halaman untuk memilih *recipient*, dan d) halaman *setup* aplikasi.

3.1.1.3 SMS Scheduler (*Author* : DiorDNA Apps)

Aplikasi ini menawarkan editor pesan yang *simple* untuk membuat pesan baru atau memuat (*load*) SMS/Draft dari aplikasi *messaging* pada *device*. Setelah mengatur waktu dan tanggal pengiriman pesan, SMS akan ditambahkan pada *list schedule* dan dikirim secara otomatis sesuai jadwalnya. Penerima pesan dapat ditulis secara manual maupun dipilih dari kontak yang ada pada *device*.

Pada aplikasi ini, terdapat pengelompokan untuk beberapa *list*, yaitu *list schedule* (*list* untuk jadwal yang dibuat), *list submitted* (*list* pesan yang sudah dikirim/ jadwal sudah dilaksanakan), *list paused* (*list* jadwal yang ditunda), *list all* (penggabungan dari *list schedule*, *submitted* dan *paused*).



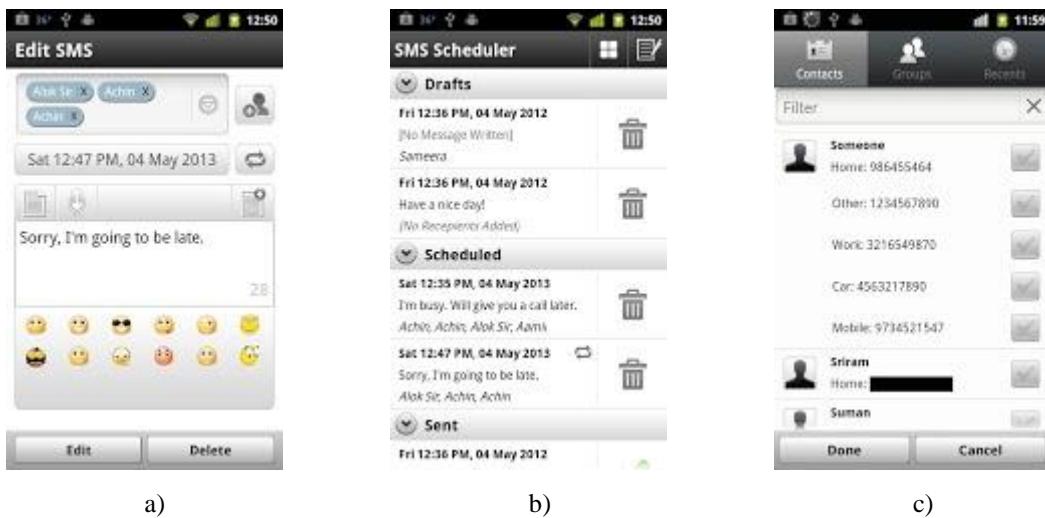
Gambar 7 . Screenshot aplikasi SMS Scheduler (Author : DiorDNA Apps)

a) halaman awal aplikasi sekaligus *interface* untuk *list all*, dan b) halaman untuk membuat jadwal pengiriman pesan.

3.1.1.4 SMS Scheduler (*Author : Vinsol*)

Secara umum, aplikasi SMS Scheduler ini sama-sama seperti aplikasi-aplikasi sebelumnya, yaitu dapat melakukan pengiriman pesan sesuai dengan jadwal yang dibuat, namun aplikasi ini memiliki fitur-fitur yang dijelaskan secara lengkap atau *detail* seperti berikut ini :

1. dapat memilih teks pesan dari *template* yang sudah dibuat,
2. dapat menambahkan pesan ke *template*,
3. dapat menambahkan grup,
4. nomor penerima pesan dapat ditulis secara manual maupun dipilih dari kontak *device*,
5. dapat memilih lebih dari satu penerima pesan (*multiple recipients*),
6. dapat melihat *history* dari pesan-pesan yang sudah dikirim,
7. dapat memasukkan pesan dengan menggunakan Google *voice*,
8. dapat mengatur frekuensi pengiriman pesan, misalnya pesan dikirim untuk satu kali pengiriman, setiap 5 menit, setiap 15 menit, setiap jam, sampai setiap tahun,
9. terdapat *emoticon* ketika sedang menuliskan pesan, dan
10. pesan diklasifikasikan menjadi *Scheduled*, *Sent* dan *Draft*.



Gambar 8 . Screenshot aplikasi SMS Scheduler (Author : Vinsol)

a) halaman untuk membuat jadwal pengiriman pesan, b) halaman yang menampilkan pengelompokan list, dan c) halaman untuk memilih *recipient*.

3.1.1.5 Holo SMS Scheduler (Author : Casey Vockrodt, Destruction Labs)

Aplikasi Holo SMS Scheduler ini sebetulnya berkerja seperti aplikasi SMS Scheduler lainnya. Yang membedakan aplikasi ini dengan aplikasi lainnya hanya pada *user interface*-nya saja, karena aplikasi ini dibuat dengan menggunakan *the Holo theme standard* untuk *Ice Cream Sandwich* (ICS) namun *compatible* dengan android 2.1 dan setelahnya (tidak hanya untuk android versi ICS saja). Fitur-fitur yang ada pada aplikasi ini pun hampir sama dengan aplikasi lainnya, yaitu :

1. nomor penerima pesan dapat ditulis secara manual maupun dipilih dari kontak *device*,
2. dapat memilih lebih dari satu penerima pesan (*multiple recipients*),
3. dapat mengatur frekuensi pengiriman pesan, misalnya pesan dikirim untuk satu kali pengiriman, setiap 5 menit, setiap 15 menit, setiap jam, sampai setiap tahun,
4. terdapat beberapa menu, yaitu *compose*, *history* dan *Draft*, dan
5. dapat melihat *history* dari pesan-pesan yang sudah dikirim.



Gambar 9 . Screenshot aplikasi Holo SMS Scheduler (Author : Casey Vockrodt, Destruction Labs)
 a) halaman untuk membuat jadwal pengiriman pesan, dan b) halaman yang menampilkan *list* pesan yang ada pada *Draft*.

3.1.2 Evaluasi Sistem Saat Ini

Dari pembahasan sebelumnya, diketahui bahwa aplikasi - aplikasi SMS *Scheduler* yang ada saat ini dapat :

1. membuat jadwal baru untuk mengirimkan pesan,
2. mengirimkan pesan berdasarkan jadwal yang sudah dibuat,
3. bersifat satu arah, maksudnya aplikasi hanya dapat mengirimkan pesan saja, tetapi tidak dapat menerima balasan pesan (balasan pesan tetap masuk ke kotak masuk *device*),
4. penerima pesan dapat ditulis secara manual maupun memilih dari kontak,
5. penerima pesan dapat dipilih lebih dari satu (*multiple*),
6. tersinkronisasi dengan kontak-kontak yang ada pada *device*,
7. mengirimkan pesan kepada nomor penerima yang diawali dengan menggunakan kode negara (misalnya 62 atau +62 untuk kode negara Indonesia),
8. melakukan pencarian nama atau nomor penerima pesan ketika sedang memilih dari kontak,
9. menampilkan notifikasi ketika pesan dikirim atau jadwal dilaksanakan,
10. menampilkan *history* pengiriman pesan,
11. jadwal yang sudah dibuat dapat diedit atau dihapus,

12. memberikan pilihan frekuensi pengiriman pesan (dikirim satu kali, setiap 5 menit, setiap hari, sampai dengan setiap tahun), dan
13. pengaturan awal untuk tanggal dan waktu yang ditampilkan ketika akan membuat jadwal (tanggal dan waktu sebelum *diset* oleh *user*) disesuaikan dengan tanggal dan waktu sistem.

Selain itu, beberapa aplikasi memiliki kelebihan-kelebihan berikut ini :

1. dapat membedakan *list* untuk jadwal pengiriman pesan, *draft*, dan pesan yang sudah dikirim,
2. memberikan fitur tambahan untuk mengatur tema *interface*-nya,
3. memiliki fitur lain selain SMS Scheduler, yaitu SMS Auto Reply, SMS Reader, Instant SMS, dan Forward SMS,
4. menyediakan *template* pesan, seperti (*i'm bussy, meeting, dll*),
5. menyediakan tambahan *smile/ emoticon* ketika menulis isi pesan,
6. dapat mengakses kontak yang ada pada *email* yang tersinkronisasi dengan *device*, dan
7. dapat mengakses grup yang ada pada kontak *device* maupun pada *email* yang tersinkronisasi dengan *device*.

Namun, ada pula beberapa hal yang tidak ditangani oleh aplikasi terkait dengan perilaku dari perangkat keras (hardware) yang digunakan, yaitu :

1. Aplikasi tidak menangani masalah kegagalan pengiriman yang diakibatkan oleh *device* dalam keadaan mati, *hang*, atau rusak, baik itu kerusakan sistem maupun kerusakan *hardware device*. Hal ini dikarenakan sistem pada *device* juga tidak berjalan semestinya dan kemungkinan sinyal pada *device* tidak tersedia.
2. Aplikasi tidak menangani masalah kegagalan pengiriman yang diakibatkan oleh tidak adanya pulsa yang cukup untuk melakukan pengiriman pesan, adanya gangguan sinyal telepon, adanya gangguan pada operator atau *provider*. Jika pulsa tidak cukup untuk melakukan pengiriman, maka provider tidak bisa melayani pengiriman pesan. Jika tidak terdapat sinyal pada telepon atau terdapat gangguan pada operator telepon, maka pengiriman pesan juga tidak dapat dilakukan.
3. Aplikasi tidak menangani masalah keterlambatan pengiriman pesan atau pengiriman pesan terlalu cepat, yang diakibatkan oleh waktu *device* yang tidak dalam kondisi

benar, artinya waktu *device* tidak sesuai dengan waktu yang sedang berlangsung. Hal ini dikarenakan aplikasi akan melakukan pengiriman jika waktu pengiriman sudah sama dengan waktu *device*.

Dari hasil evaluasi diatas, didapatkan hasil analisis dari sepuluh aplikasi-aplikasi SMS *Scheduler* diatas.

Tabel 2. Hasil analisis Aplikasi SMS *Scheduler* yang ada saat ini

Fungsi	(1) Lylynx.com	(2) cuubonandroid	(3) ZYXSolutions	(4) Thein Min Naing	(5) DiorDNA Apps	(6) Globalapps R	(7) VinSol	(8) Shaper- studio	(9) Liuge, Good App	(10) Casey Vockrodt
Dapat menginput nomor <i>recipient</i> dengan kode negara (+62)/(62)	√	√	√	√	√	√	√	√	√	√
Dapat mengambil nomor dari kontak <i>device</i>	x	x	√	√	√	√	√	√	√	√
Dapat mengambil nomor dari kontak <i>email</i>	√	x	x	x	x	x	x	x	x	x

Keterangan : √ : Ya

x : Tidak

- : Tidak Jelas

Tabel 3. Hasil analisis Aplikasi SMS Scheduler yang ada saat ini (Lanjutan)

Fungsi	(1) Lylynx.com	(2) cuubonandroid	(3) ZYXSolutions	(4) Thein Min Naing	(5) DiorDNA Apps	(6) Globalapps R	(7) VinSol	(8) Shaper- studio	(9) Liuge, Good App	(10) Casey Vockrodt
Dapat melakukan filtering saat melakukan pemilihan recipient	√	x	x	√	√	√	√	x	x	√
Dapat melakukan pengiriman ke lebih dari satu recipient	√	√	√	√	√	√	√	√	√	√
Dapat melakukan pengiriman ke nomor dan pada jam yang sama, tetapi dengan jadwal yang berbeda	√	√	√	√	√	√	√	√	√	√

Keterangan :

√ : Ya

x : Tidak

- : Tidak Jelas

Tabel 4. Hasil analisis Aplikasi SMS Scheduler yang ada saat ini (Lanjutan)

Fungsi	(1) Lylynx.com	(2) cuubonandroid	(3) ZYXSolutions	(4) Thein Min Naing	(5) DiorDNA Apps	(6) Globalapps R	(7) VinSol	(8) Shaper- studio	(9) Liuge, Good App	(10) Casey Vockrodt
Memberikan pilihan frekuensi pengiriman pesan (dikirim satu kali, setiap 5 menit, setiap hari, sampai dengan setiap tahun)	√	√	√	√	√	√	√	√	√	√
Terdapat notifikasi ketika pesan dikirim	√	√	√	√	√	√	√	√	√	√
Terdapat keterangan berhasil dikirim atau gagal	√	√	x	x	√	-	x	x	√	x

Keterangan : √ : Ya

x : Tidak

- : Tidak Jelas

Tabel 5. Hasil analisis Aplikasi SMS Scheduler yang ada saat ini (Lanjutan)

Fungsi	(1) Lylynx.com	(2) cuubonandroid	(3) ZYXSolutions	(4) Thein Min Naing	(5) DiorDNA Apps	(6) Globalapps R	(7) VinSol	(8) Shaper- studio	(9) Liuge, Good App	(10) Casey Vockrodt
Dapat melakukan pengiriman ulang untuk <i>schedule</i> yang gagal.	x	√	x	x	√	-	x	x	x	x
Terdapat <i>history</i>	√	x	x	√	√	-	x	x	√	√
Menjalankan fungsi <i>schedule</i>	√	√	√	√	√	x	√	√	√	√
Jadwal yang sudah dibuat dapat diedit atau dihapus	√	√	√	√	√	√	√	√	√	√
Mengkategorikan pesan (<i>schedule</i> , <i>sent</i> , <i>failed</i> , dll)	x	x	x	x	√	x	√	x	x	x

Keterangan :

√ : Ya

x : Tidak

- : Tidak Jelas

Tabel 6. Hasil analisis Aplikasi SMS Scheduler yang ada saat ini (Lanjutan)

Fungsi	(1) Lylynx.com	(2) cuubonandroid	(3) ZYXSolutions	(4) Thein Min Naing	(5) DiorDNA Apps	(6) Globalapps R	(7) VinSol	(8) Shaper- studio	(9) Liuge, Good App	(10) Casey Vockrodt
Menyediakan <i>template</i> pesan	x	x	x	√	x	x	√	x	x	x
Pengiriman jadwal untuk waktu yg sudah lampau dilanjutkan setelah <i>device</i> dihidupkan kembali.	x	x	x	x	x	x	√	x	x	x
Pengiriman jadwal untuk waktu yg belum lampau dilanjutkan setelah <i>device</i> dihidupkan kembali.	√	√	√	x	x	√	√	x	√	x

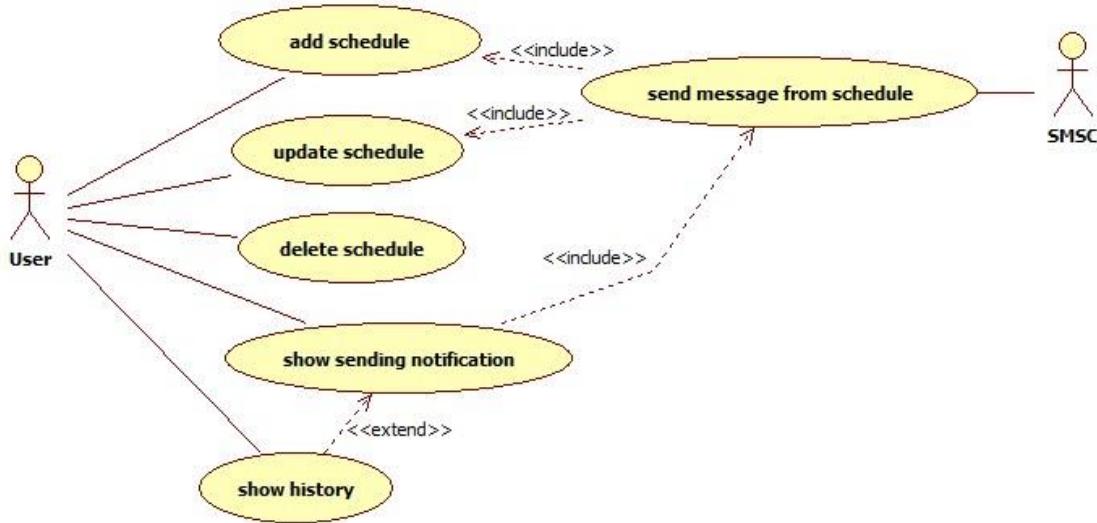
Keterangan :

√ : Ya

x : Tidak

- : Tidak Jelas

Dari hasil analisis *current system* diatas, dapat dijelaskan pula mengenai interaksi antara aktor dengan *current system* yang digambarkan dalam bentuk *use case diagram* pada gambar 10.



Gambar 10 . Use case diagram current system SMS Scheduler

Aktor merupakan sesuatu atau seseorang yang men-stimulate sistem untuk memberikan respon terhadap *request* yang diberikannya. Aktor-aktor yang terlibat dalam *use case diagram* diatas dijelaskan pada tabel 7.

Tabel 7. Deskripsi aktor

No	Aktor	Deskripsi
A01	User	Merupakan orang yang akan mengirim pesan. User juga merupakan orang yang akan mengatur kapan dan kemana pesan akan dikirimkan
A02	SMSC	Merupakan provider sesuai dengan SIM card yang digunakan pada device

Sementara itu, *use case* menggambarkan fungsi yang dilakukan sistem sesuai dengan keinginan aktor. Tabel 8 menjelaskan tentang *use cases* pada *current system* beserta deskripsinya. Deskripsi *use case* merupakan deskripsi singkat mengenai penggunaan proses.

Tabel 8. Deskripsi case

No	Use Case	Deskripsi
UC01	Add schedule	Case ini digunakan ketika user membuat <i>schedule</i> baru. Case ini akan menerima data yang diinput user dan menyimpan data tersebut sebagai data <i>schedule</i> baru ke <i>database</i> .
UC02	Update schedule	Case ini digunakan ketika user melakukan perubahan pada <i>schedule</i> yang sudah ada sebelumnya. Case ini akan menerima perubahan data yang diinput user dan melakukan <i>update</i> ke <i>database</i> sesuai dengan <i>schedule</i>

No	Use Case	Deskripsi
UC03	<i>Delete schedule</i>	yang akan diubah. <i>Case</i> yang digunakan ketika <i>user</i> menghapus <i>schedule</i> pengiriman pesan yang sudah dibuat sebelumnya. <i>Case</i> ini akan menghapus <i>schedule</i> yang dipilih <i>user</i> .
UC04	<i>Send message from schedule</i>	<i>Case</i> ini digunakan untuk mengirimkan pesan sesuai urutan yang dihasilkan proses pengecekan waktu pengiriman.
UC05	<i>Show sending notification</i>	<i>Case</i> ini akan menampilkan notifikasi ketika proses pengiriman pesan dilakukan atau sedang berlangsung.
UC06	<i>Show history</i>	<i>Case</i> ini digunakan untuk menampilkan laporan pengiriman jadwal yang telah dilakukan, baik pengiriman jadwal yang berhasil maupun yang gagal.

Skenario *use case* digunakan untuk menjelaskan tahapan proses yang ada pada masing-masing *use case*. Skenario yang terdapat dalam masing-masing *case* pada *use case* diagram dijelaskan pada tabel 9 sampai dengan tabel 17.

Tabel 9. Skenario *use case add schedule*

Case ID	UC01		
Nama Case	<i>Add new schedule</i>		
Aktor yang berperan	<i>User</i>		
Deskripsi	<i>Case</i> ini digunakan ketika <i>user</i> membuat <i>schedule</i> baru. <i>Case</i> ini akan menerima data yang diinput <i>user</i> dan menyimpan data tersebut sebagai data <i>schedule</i> baru ke <i>database</i> .		
Main Success Scenario	Aktor	Sistem	
	1. <i>User</i> memilih menu untuk membuat <i>schedule</i> baru	2. Sistem menampilkan <i>form</i> untuk membuat <i>schedule</i> baru yang terdiri dari data <i>recipient</i> , <i>date and time</i> , <i>sending frequency</i> dan <i>message</i> .	
	3. <i>User</i> mengisi <i>form</i> dan melakukan konfirmasi pembuatan <i>schedule</i> baru.	4. Sistem menerima masukan dari <i>user</i> berupa data-data yang akan disimpan ke <i>database</i> .	
		5. Sistem melakukan validasi pada setiap data yang diinputkan <i>user</i> untuk memastikan bahwa tidak ada data yang masih kosong.	
		6. Sistem melakukan validasi pada waktu pengiriman untuk memastikan waktu yang diinputkan bukan waktu yang lampau.	
		7. Sistem mengirimkan <i>form</i> yang sudah terisi data-data yang valid ke <i>database</i> untuk selanjutnya dilakukan proses penyimpanan <i>schedule</i> .	
		8. Apabila <i>form</i> sudah terisi semua dan data yang dimasukkan sesuai dengan aturan, maka sistem akan memasukkan data-data yang diinput <i>user</i> ke <i>database</i> .	
		9. <i>Form</i> yang sudah lengkap dan dengan waktu pengiriman yang valid disimpan ke tabel <i>schedule</i> yang ada di <i>database</i> .	

Tabel 10. Skenario case *add schedule* (lanjutan)

Case ID	UC01	
Nama Case	<i>Add schedule</i>	
Skenario Alternatif	Aktor	Sistem
3.a. <i>User</i> menggunakan <i>template</i> pada <i>content message</i>	3.a.1 <i>User</i> memilih <i>template</i> untuk mengisi <i>content message</i>	3.a.2 Sistem menampilkan <i>template</i> yang dipilih <i>user</i> di <i>content message</i> .
Skenario Alternatif	3.b.1 <i>User</i> melakukan konfirmasi pembuatan <i>schedule</i> dengan <i>form</i> yang belum lengkap	3.b.2 Sistem menampilkan pesan <i>error</i> dan meminta <i>user</i> untuk melengkapi data yang masih kosong.
Skenario Alternatif	3.c.1 <i>User</i> melakukan konfirmasi pembuatan <i>schedule</i> dengan waktu pengiriman <i>schedule</i> yang sudah lampau	3.c.2 Sistem menampilkan pesan <i>error</i> dan meminta <i>user</i> untuk mengganti waktu pengiriman.
Skenario Alternatif	3.d.1 <i>User</i> membatalkan proses untuk membuat <i>schedule</i> baru	3.d.2 Sistem menampilkan halaman yang sebelumnya dibuka.
Exception		3.e.1 Pembuatan <i>schedule</i> dibatalkan
3.e <i>Device</i> mati atau <i>hang</i> ketika <i>user</i> mengkonfirmasi pembuatan <i>schedule</i>		
Exception		9.a.1 Proses penyimpanan <i>schedule</i> ke <i>database</i> dibatalkan.
9.a <i>Device</i> mati atau <i>hang</i> ketika sistem akan menyimpan <i>schedule</i> ke <i>database</i>		

Tabel 11. Skenario case *update schedule*

Case ID	UC02	
Nama Case	<i>Update schedule</i>	
Aktor yang berperan	<i>User</i>	
Deskripsi	<i>Case</i> ini digunakan ketika <i>user</i> melakukan perubahan pada <i>schedule</i> yang sudah ada sebelumnya. <i>Case</i> ini akan menerima perubahan data yang diinput <i>user</i> dan melakukan <i>update</i> ke <i>database</i> sesuai dengan <i>schedule</i> yang akan diubah.	
Main Success Scenario	Aktor	Sistem
	1. <i>User</i> memilih menu untuk mengubah <i>schedule</i> yang sudah ada	2. Sistem menampilkan <i>form</i> yang sudah berisi informasi <i>schedule</i> pengiriman pesan yang dipilih <i>user</i> untuk dilakukan perubahan data.
	3. <i>User</i> mengisi <i>form</i> dan melakukan konfirmasi perubahan <i>schedule</i> baru.	4. Sistem menerima masukan dari <i>user</i> berupa data-data yang akan di- <i>update</i> ke <i>database</i> .
		5. Sistem melakukan validasi pada setiap data yang diinputkan <i>user</i> untuk memastikan bahwa tidak ada data yang masih kosong.

Tabel 12. Skenario *case update schedule* (lanjutan)

Case ID	UC02	
Nama Case	<i>Update schedule</i>	
Main Success Scenario	Aktor	Sistem
		6. Sistem melakukan validasi pada waktu pengiriman untuk memastikan waktu yang diinputkan bukan waktu yang lampau.
		7. Sistem mengirimkan form yang sudah terisi data-data yang valid ke database untuk selanjutnya dilakukan proses penyimpanan <i>schedule</i> .
		8. Apabila <i>form</i> sudah terisi semua dan data yang dimasukkan sesuai dengan aturan, maka sistem akan memasukkan data-data yang diinput <i>user</i> ke <i>database</i> .
Skenario Alternatif 3.a. <i>User</i> menggunakan <i>template</i> pada <i>content message</i>	3.a.1 <i>User</i> memilih <i>template</i> untuk mengisi <i>content message</i>	3.a.2 Sistem menampilkan <i>template</i> yang dipilih <i>user</i> di <i>content message</i> .
Skenario Alternatif 3.b <i>User</i> menginputkan <i>form</i> yang belum lengkap	3.b.1 <i>User</i> melakukan konfirmasi perubahan <i>schedule</i> dengan <i>form</i> yang belum lengkap	3.b.2 Sistem menampilkan pesan <i>error</i> dan meminta <i>user</i> untuk melengkapi data yang masih kosong.
Skenario Alternatif 3.c <i>User</i> menginputkan waktu pengiriman yang sudah lampau	3.c.1 <i>User</i> melakukan konfirmasi perubahan <i>schedule</i> dengan waktu pengiriman <i>schedule</i> yang sudah lampau	3.c.2 Sistem menampilkan pesan <i>error</i> dan meminta <i>user</i> untuk mengganti waktu pengiriman.
Skenario Alternatif 3.d <i>User</i> membatalkan proses untuk mengubah <i>schedule</i>	3.d.1 <i>User</i> membatalkan proses untuk mengubah <i>schedule</i> yang sudah ada sebelumnya	3.d.2 Sistem menampilkan halaman yang sebelumnya dibuka.
Exception 3.f <i>Device</i> mati atau <i>hang</i> ketika <i>user</i> mengkonfirmasi perubahan <i>schedule</i>		3.f.1 Sistem membatalkan proses perubahan <i>schedule</i> .
Exception 9.a <i>Device</i> mati atau <i>hang</i> ketika sistem akan menyimpan <i>schedule</i> ke <i>database</i>		9.a.1 Sistem membatalkan proses penyimpanan <i>schedule</i> ke <i>database</i> .

Tabel 13. Skenario *case delete schedule*

Case ID	UC03	
Nama Case	<i>Delete schedule</i>	
Aktor yang berperan	<i>User</i>	
Deskripsi	Case yang digunakan ketika <i>user</i> menghapus <i>schedule</i> pengiriman pesan yang sudah dibuat sebelumnya. Case ini akan menghapus <i>schedule</i> yang dipilih <i>user</i> .	
Main Success Scenario	Aktor	Sistem
	1. <i>User</i> memilih <i>schedule</i> yang akan dihapus	2. Sistem menerima konfirmasi penghapusan <i>schedule</i> dari <i>user</i> . Kemudian, sistem menghapus <i>schedule</i> dari kategori <i>list scheduled</i> .
Skenario Alternatif 1.a <i>User</i> membatalkan penghapusan <i>schedule</i>	1.a.1 <i>User</i> membatalkan proses untuk menghapus <i>schedule</i>	3. Pada <i>database</i> , dilakukan penghapusan data yang terkait dengan <i>schedule</i> yang dipilih untuk dihapus.
Exception 2.a <i>Device</i> mati atau hang ketika sistem akan menghapus <i>schedule</i> ke <i>database</i>		2.a.1 Proses penghapusan <i>schedule</i> dibatalkan.

Tabel 14. Skenario *case send message from schedule*

Case ID	UC04	
Nama Case	<i>Send message from schedule</i>	
Aktor yang berperan	<i>User</i> dan <i>SMSC</i>	
Deskripsi	Sistem melakukan pengecekan waktu pengiriman pada <i>schedule</i> yang sudah tersimpan di <i>database</i> untuk kemudian dilakukan pengiriman.	
Main Success Scenario	Aktor	Sistem
	1. <i>User</i> membuat jadwal baru atau mengubah jadwal yang telah ada	2. Sistem melakukan proses validasi dan penyimpanan jadwal ke tabel <i>schedule</i> di <i>database</i>
		3. Sistem melakukan pengecekan waktu pengiriman antara waktu yang ada pada <i>schedule</i> dengan waktu sistem (waktu <i>device</i>).
		4. Ketika waktu pada <i>schedule</i> sama dengan waktu sistem, sistem mengambil data-data yang terkait dengan <i>schedule</i> tersebut
		5. Sistem mengirimkan pesan sesuai dengan data-data yang terkait dengan <i>schedule</i> , dimana waktu <i>schedule</i> tersebut sudah sama dengan waktu sistem.
	6. <i>SMSC</i> menerima pesan yang dikirim aplikasi	

Tabel 15. Skenario *case send message from schedule* (lanjutan)

Case ID	UC04	
Nama Case	<i>Send message from schedule</i>	
Skenario Alternatif	Aktor	Sistem
3.a Belum ditemukan waktu <i>schedule</i> yang sama dengan waktu sistem		3.a.1 Jika saat pengecekan ternyata waktu pada <i>schedule</i> belum sama dengan waktu sistem, sistem kembali melakukan pengecekan pada <i>schedule</i> lainnya hingga menemukan waktu <i>schedule</i> yang sama dengan waktu sistem
Skenario Alternatif 3.b Terdapat beberapa <i>schedule</i> yang harus dilaksanakan di waktu yang sama		3.b.1 Jika pada saat pengecekan sistem menemukan beberapa <i>schedule</i> yang harus dilaksanakan pada waktu yang sama, sistem akan melakukan pengecekan berdasarkan waktu penyimpanannya.
Skenario Alternatif 5.a Ketika sedang melakukan pengiriman, waktu sistem sudah maju ke waktu berikutnya		5.a.1 Jika pengiriman suatu <i>schedule</i> masih belum selesai tetapi waktu sistem sudah maju ke waktu berikutnya, sistem menyelesaikan pengiriman <i>schedule</i> yang belum selesai tersebut terlebih dahulu.
Skenario Alternatif 5.b Terdapat beberapa <i>schedule</i> yang harus dilaksanakan di waktu yang sama		5.b.1 Jika pada saat pengecekan waktu pengiriman, sistem menemukan beberapa <i>schedule</i> yang harus dilaksanakan pada waktu yang sama, sistem akan terlebih dahulu melaksanakan pengiriman <i>schedule</i> yang waktu penyimpanannya lebih awal.
Exception 5.c Pulsa habis ketika akan melakukan pengiriman atau sedang melakukan pengiriman		5.c.1 Jika sistem akan melakukan pengiriman atau sedang melakukan pengiriman dan pulsa pada <i>device</i> habis, maka pesan tidak dikirimkan dan <i>schedule</i> dianggap gagal.
Exception 5.d Sinyal tiba-tiba tidak tersedia ketika melakukan pengiriman atau sedang melakukan pengiriman		5.d.1 Jika sistem akan melakukan pengiriman atau sedang melakukan pengiriman dan sinyal tidak tersedia, maka pesan tidak dikirimkan dan <i>schedule</i> dianggap gagal.
Exception 5.e <i>Device</i> mati atau <i>hang</i> ketika akan melakukan pengiriman atau sedang melakukan pengiriman		5.e.1 Jika sistem akan melakukan pengiriman atau sedang melakukan pengiriman dan <i>device</i> mati atau <i>hang</i> , maka pesan tidak dikirimkan dan <i>schedule</i> dianggap gagal.

Tabel 16. Skenario *case show notification*

Case ID	UC05	
Nama Case	<i>Show notification</i>	
Aktor yang berperan	<i>User</i>	
Deskripsi	Sistem menampilkan notifikasi ketika melakukan pengiriman pesan	
Main Success Scenario	Aktor	Sistem
	1. <i>User</i> membuat jadwal baru atau mengubah jadwal yang telah ada	2. Sistem menjalankan proses pengecekan dan mengirimkan pesan sesuai dengan jadwal yang telah disimpan pada <i>database</i> .
		3. Sistem menampilkan notifikasi pengiriman pesan.

Tabel 17. Skenario *case show history*

Case ID	UC06	
Nama Case	<i>Show history</i>	
Aktor yang berperan	<i>User</i>	
Deskripsi	Sistem menampilkan <i>history</i> pengiriman pesan yang sudah dilakukan.	
Main Success Scenario	Aktor	Sistem
	1. <i>User</i> membuka notifikasi pengiriman pesan atau memilih menu <i>history</i>	2. Sistem menampilkan <i>history</i> pengiriman pesan

3.2 Analisis Kebutuhan

Analisis kebutuhan aplikasi dilakukan untuk mendapatkan apa saja kebutuhan-kebutuhan yang diperlukan oleh *user* untuk aplikasi SMS *Scheduler*. Analisis kebutuhan perangkat lunak ini didapatkan dari hasil evaluasi analisis sistem yang ada saat ini.

3.2.1 Domain Kajian

Dari hasil evaluasi sistem yang sudah ada saat ini, didapatkan kebutuhan-kebutuhan yang diperlukan untuk pengembangan aplikasi ini. Kebutuhan-kebutuhan tersebut dijelaskan sebagai berikut.

1. Aplikasi dapat membuat jadwal baru untuk mengirimkan pesan.
2. Aplikasi dapat mengirimkan pesan berdasarkan jadwal yang sudah dibuat.
3. Aplikasi bersifat satu arah, maksudnya aplikasi hanya dapat mengirimkan pesan saja. tetapi tidak dapat menerima balasan pesan (balasan pesan tetap masuk ke kotak masuk *device*).
4. Penerima pesan dapat ditulis secara manual maupun memilih dari kontak.
5. Penerima pesan dapat dipilih lebih dari satu (*multiple*).
6. Aplikasi dapat tersinkronisasi dengan kontak-kontak yang ada pada *device*.
7. Aplikasi dapat mengakses grup yang ada pada kontak *device*.

8. Aplikasi dapat mengirimkan pesan kepada nomor penerima yang diawali dengan menggunakan kode negara (misalnya 62 atau +62 untuk kode negara Indonesia).
9. Aplikasi dapat melakukan pencarian nama atau nomor penerima pesan ketika sedang memilih dari kontak.
10. Aplikasi dapat menampilkan notifikasi ketika pesan dikirim atau jadwal dilaksanakan.
11. Aplikasi dapat menampilkan *history* pengiriman pesan.
12. Jadwal yang sudah dibuat dapat diedit atau dihapus.
13. Aplikasi dapat memberikan pilihan frekuensi pengiriman pesan (dikirim satu kali, setiap 5 menit, setiap hari, sampai dengan setiap tahun)
14. Pengaturan awal untuk tanggal dan waktu yang ditampilkan ketika akan membuat jadwal (tanggal dan waktu sebelum *diset* oleh *user*) disesuaikan dengan tanggal dan waktu sistem.
15. Aplikasi memiliki *user interface* yang *user friendly*, artinya *user interface* mudah dioperasikan, interaktif (*user* dapat dengan mudah berinteraksi dengan aplikasi), dan memenuhi kebutuhan *user*.
16. Aplikasi dapat mengelompokkan pesan sesuai dengan kategorinya, misalnya kategori *schedule*, *sent*, *paused*, dan *failed*.
17. Aplikasi dapat memberikan peringatan jika *user* mengatur tanggal atau waktu pengiriman pesan yang sudah lampau.
18. Aplikasi dapat menampilkan nama-nama kontak yang pesannya tidak terkirim.
19. Aplikasi dapat melakukan pengiriman ulang untuk nama-nama kontak yang pesannya tidak terkirim.
20. Aplikasi dapat memberikan informasi ketika pesan sudah diterima pada *device recipient*.
21. Aplikasi menyediakan fitur *template* pesan.
22. Aplikasi tersedia untuk beberapa *platform*.

Detail dari kebutuhan aplikasi tersebut dapat dilihat pada dokumen Lampiran A - *Software Requirement Specification* (SRS).

3.2.2 Teknologi

Pada bagian ini akan dijelaskan mengenai teknologi yang digunakan untuk membuat aplikasi SMS *Scheduler*. Teknologi tersebut terdiri dari teknik pembuatan aplikasi dan *tools* yang akan digunakan.

3.2.2.1 Teknik Pembuatan Aplikasi

Untuk memenuhi kebutuhan aplikasi agar dapat dinikmati oleh pengguna *smartphone* dan *non-smartphone*, maka dilakukan pengembangan dengan membuat aplikasi di beberapa *platform*. Dalam pembuatan aplikasi dikenal dengan 3 teknik, yaitu *native*, *mobile web*, dan *hybrid*. Karena aplikasi SMS *Scheduler* merupakan aplikasi berbasis *mobile*, maka kemungkinan teknik pembuatan aplikasi yang digunakan adalah *native* dan *hybrid application*. Teknik *native* merupakan teknik pembuatan aplikasi yang diinstal langsung pada *device* dengan menggunakan bahasa pemrograman yang digunakan untuk membuat aplikasi tersebut (*native programming language*). Dengan kata lain, teknik pembuatan aplikasi secara *native* menghasilkan aplikasi yang khusus untuk suatu sistem operasi tertentu dan biasanya sistem operasi menyediakan *tools* dan API khusus bagi *developer* untuk membuat aplikasi. Contohnya untuk membuat aplikasi android, menggunakan bahasa pemrograman Java dan menggunakan ADT (*Android Development Tools*), dan android SDK yang telah disediakan android. Kelebihan *native* diantaranya adalah aplikasi memiliki performa yang cepat, karena sudah terinstal langsung di *device*, dapat mengakses semua fitur yang dimiliki oleh sistem operasi yang bersangkutan, seperti *alarm*, *calendar*, dan *contact*. Sementara itu, kekurangan *native* adalah memerlukan biaya dan *effort* yang lebih besar karena pembuatan aplikasi untuk setiap *platform* dilakukan satu per satu.

Teknik *hybrid* merupakan penggabungan antara teknik *native* dan *web*. Dengan teknik *hybrid*, aplikasi yang dihasilkan sama seperti *native*, yaitu aplikasi yang diinstal langsung pada *device* namun dengan menggunakan bahasa pemrograman yang digabungkan antara bahasa pemrograman *web*, yang digunakan untuk membuat aplikasi inti dan *native programming language*-nya digunakan untuk mengakses fitur yang tidak dapat dijalankan dengan bahasa pemrograman *web*. Berikut ini merupakan kelebihan dari teknik *hybrid*.

1. Dapat digunakan untuk membuat aplikasi *multiplatform* (android, blacberry, Symbian, Windows Phone, Samsung Bada, dan WebOS).
2. Tidak perlu mempelajari bahasa pemrograman *native* dari aplikasi yang akan dibuat, misalnya bahasa pemrograman untuk aplikasi android (java), objective-c (iOS), dan C# (*windows phone*).
3. Bersifat *open source*, sehingga aplikasi masih dapat dikembangkan oleh para *developer*.

Namun disamping itu, ada pula kekurangan dari teknik *hybrid*, yaitu :

1. Terdapat beberapa fitur *device* yang tidak dapat diakses.
2. Biasanya fitur-fitur pada aplikasi yang mendukung teknik *hybrid* kurang *up-to-date* daripada aplikasi yang mendukung teknik *native*.
3. Tampilan untuk semua *platform* yang dibuat cenderung sama.
4. Terdapat kendala jika disinkronisasikan dengan iOS *platform*, misalnya masalah pada *user interface*.
5. Aplikasi yang dibuat tidak seoptimal aplikasi yang dibuat secara *native*, karena aplikasi yang mendukung teknik *hybrid* perlu menginterpretasikan *javascript* terlebih dahulu agar bisa diterima oleh *native* SDK dari *platform* yang dibuat. Hal ini juga membuat performa aplikasi menjadi turun, karena dipengaruhi oleh waktu yang cukup lama untuk proses interpretasi.

Karena aplikasi SMS *Scheduler* memerlukan fitur-fitur khusus dari *device* (seperti alarm, *calendar*, dan *contact*), tidak memerlukan koneksi internet, dan adanya keinginan untuk menyesuaikan *user interface* sesuai *platform* yang akan dibuat, maka disimpulkan bahwa teknik pembuatan aplikasi yang akan digunakan adalah teknik *native development*.

3.2.2.2 Tools

Informasi untuk teknologi yang digunakan dalam pembuatan aplikasi SMS *Scheduler* yang sudah ada saat ini sebetulnya sangat sulit didapatkan, karena masing-masing *developer* tidak memberikan keterangan *tools* apa yang digunakan untuk membuat aplikasi tersebut. Oleh karena itu, dilakukan analisis dengan bantuan *mapping tools*.

Mapping tools digunakan agar memudahkan *developer* untuk mengetahui *tools-tools* apa saja yang dapat digunakan untuk membuat suatu aplikasi, terutama jika pengembangan aplikasi tersebut akan dibuat pada beberapa *platform*, dimana setiap

platform pasti memiliki banyak *tools* yang dapat digunakan. Selain itu, *mapping tools* juga digunakan untuk mengetahui kesamaan yang ada diantara satu *tool* pengembangan dengan *tool* pengembangan lainnya. Hal tersebut dimaksudkan untuk meminimalisir penggunaan *tools*. Karena, jika telah diketahui *tools* untuk *platform* mana saja yang memiliki kesamaan, maka *tools* tersebutlah yang akan digunakan dan *developer* tidak harus melakukan instalasi semua *tools* untuk semua *platform*.

Untuk membuat aplikasi android, terdapat dua IDE yang biasa digunakan *developer*, yaitu Eclipse IDE dan Netbeans IDE. Kelebihan dari Eclipse IDE ini antara lain :

1. Merupakan *official tool* untuk men-*develop*e aplikasi android.
2. Lebih ringan, karena tidak perlu diinstall (*portable*).
3. Terdapat *shortcut keys* yang sangat menguntungkan dalam *developing*.
4. *Open source*.
5. *Plugin*-nya lebih banyak dan bervariasi.

Sementara itu, kekurangan dari Eclipse IDE ini antara lain :

1. *Coding* dilakukan dari awal, karena ketika membuat *project*, *worksheet*-nya *blank*. Sebutulnya ini sangat bagus untuk pemula, karena akan membantu untuk mengingat *code* apa saja yang harus ada ketika mulai melakukan *coding*. Namun, bagi beberapa *user*, hal ini justru tidak disukai karena dianggap tidak membantu.
2. Beberapa *user* mengatakan bahwa *user interface* dari Eclipse IDE kurang menarik.

Untuk Netbeans IDE, terdapat beberapa kelebihan, diantaranya :

1. *User Interface*-nya lebih menarik.
2. Tidak perlu *coding* dari awal (tinggal melakukan implementasi saja).

Dan kekurangan dari Netbeans IDE ini antara lain :

1. Sering hang ditengah-tengah *coding*.
2. Sering terjadi *error* ketika sedang menginstall ADT *plugin*.
3. Harus diinstall dan memerlukan RAM minimal 512 Mb

Oleh karena itu, untuk pengembangan aplikasi SMS *Scheduler* pada android akan menggunakan *tools* Eclipse IDE (bersama dengan Java 6, Android SDK dan ADT *Plugin*).

Untuk membuat aplikasi Blackberry OS, terdapat dua *tools* yang dapat digunakan, yaitu Eclipse dan Netbeans. Tetapi *tools* yang digunakan untuk pengembangan aplikasi SMS *Scheduler* ini adalah Eclipse. Alasannya hampir sama dengan penggunaan Eclipse untuk android, walaupun memang Eclipse bukanlah *tools* yang disarankan oleh para *developer* blackberry *apps*. Selain itu, penggunaan Eclipse ini dapat menghemat memori dan lebih praktis, karena hanya perlu menginstall Blackberry SDK dan Blackberry *plugin* pada Eclipse saja. Jadi, satu *software* Eclipse dapat digunakan untuk men-*develop* dua aplikasi yang berbeda *platform* sekaligus.

Untuk membuat aplikasi Java OS, terdapat dua *tools* yang dapat digunakan, yaitu Netbeans dan Nokia Eclipse for J2ME. Namun, *tools* yang akan digunakan untuk pengembangan aplikasi SMS *Scheduler* pada Java OS adalah Netbeans. Alasan utama dipilihnya *tools* tersebut adalah karena aplikasi yang dihasilkan dari *tools* tersebut tidak hanya untuk handphone Nokia saja, melainkan untuk *handphone-handphone* java lainnya. Sementara dengan menggunakan Nokia Eclipse for J2ME, aplikasi yang dihasilkan merupakan aplikasi untuk Nokia saja.

Untuk membuat aplikasi iOS, *tools* yang digunakan adalah Xcode. Xcode merupakan *official tools* yang digunakan untuk membuat aplikasi iOS. Sebetulnya masih ada beberapa *tools* yang dapat digunakan untuk *platform* iOS, namun *tools* tersebut digunakan untuk membuat *game* pada *platform* iOS, bukan aplikasi iOS. *Tools* lain yang digunakan adalah iOS SDK yang sudah termasuk di dalam Xcode.

Hasil dari *mapping tools* dapat dilihat pada Lampiran B - *Mapping Tools*.

3.3 Pengumpulan Data (Survei)

Untuk mengetahui apakah fitur-fitur tambahan yang akan dibuat pada aplikasi SMS Scheduler ini diperlukan oleh masyarakat, terutama pengguna mobile device dan pengguna SMS, maka diadakan survei dengan teknik pembuatan kuisioner. Dengan pembuatan kuisioner ini agar diperoleh tingkat keandalan (reliability) dan keabsahan (validity) setinggi mungkin.

Pada tahap pembuatan kuesioner ini, dipersiapkan informasi yang ingin didapatkan dari responden, setelah itu barulah dilakukan penyusunan pertanyaan. Jenis pertanyaan yang digunakan dalam kuesioner ini adalah pertanyaan tertutup (ya dan tidak).

Responden dari survei ini terdiri dari 200 orang yang merupakan perwakilan dari para pengguna *mobile device* dan pengguna SMS. Kuisioner disebar dengan media internet di forum-forum POLBAN, terutama di JTK, baik itu ditujukan bagi mahasiswa yang masih aktif kuliah maupun alumni. Selain itu, kuisioner juga disebar di beberapa forum umum di media sosial *twitter* dan *facebook*.

Bentuk form kuisioner yang digunakan dalam survei ini dapat dilihat pada lampiran D. Pada form kuisioner tersebut, terdapat 16 butir pertanyaan mengenai fitur-fitur yang akan dibuat pada aplikasi SMS *Scheduler*. Pada butir 1-2, merupakan pertanyaan mengenai fitur *template*. Pada butir 3-7, merupakan pertanyaan mengenai fitur *edit schedule*. Pada butir 8-9, merupakan pertanyaan mengenai fitur *pause* dan *resume*. Pada butir 10, merupakan pertanyaan mengenai fitur *delete schedule*. Pada butir 11, merupakan pertanyaan mengenai fitur *alert*. Pada butir 12-15, merupakan pertanyaan untuk fitur *notification*. Dan pada butir 16, merupakan pertanyaan untuk fitur kategori pesan.

3.3.1 Kesimpulan Hasil Survei

Berdasarkan hasil survei yang telah dilakukan, didapatkan hasil sebagai berikut.

Tabel 18. Hasil Survei

Butir Pertanyaan	Jawaban	
	Ya	Tidak
1	65%	35%
2	81%	19%
3	97,5%	2,5%
4	86,5%	13,5%
5	81,5%	18,5%
6	76,5%	23,5%
7	75%	25%
8	78%	22%
9	82%	18%
10	95%	5%
11	82%	18%
12	80,5%	19,5%
13	98%	2%
14	96%	4%
15	92,5%	7,5%
16	83%	17%

Dari hasil survei tersebut, dapat disimpulkan bahwa semua fitur yang akan dibuat pada aplikasi SMS *Scheduler* diperlukan oleh *user*. Oleh karena itu, diputuskan bahwa fitur-fitur sesuai dengan pertanyaan dari butir 1-16 akan dibuat pada aplikasi SMS

Scheduler disamping fitur utama aplikasi, yaitu fitur membuat *schedule* baru dan mengirimkan pesan sesuai *schedule*.

3.4 Kesimpulan Analisis dan Pembahasan

Dari analisis yang dilakukan, didapatkan hasil bahwa dari aplikasi yang ada saat ini (*current system*) masih terdapat beberapa hal yang belum ditangani, yaitu mengenai kurangnya informasi pengiriman pesan bagi *user* yang mengirimkan pesan (notifikasi), pesan dinamis (*typical message*), *alert* dan *setting* aplikasi. Selain itu, masalah lainnya terdapat pada aplikasi yang dapat dijalankan pada *platform* Java ME/ J2ME. Pada aplikasi tersebut, nomor penerima pesan yang digunakan harus ditulis secara manual (tidak dapat memilih dari kontak *device*). Sementara itu, pada aplikasi yang dapat dijalankan di *platform* Blackberry, mekanisme pengambilan nomor penerima pesan untuk *multiple recipient* tidak dapat dilakukan secara *grouping*, sehingga harus diulang sebanyak *recipient* yang akan digunakan. Dikarenakan masalah-masalah diatas belum dapat ditangani pada aplikasi yang ada saat ini, baik aplikasi pada *platform* Android, Blackberry, iOS, dan Java ME/ J2ME, ditambah dengan adanya masalah penerima pesan pada aplikasi dengan *platform* Blackberry dan Java ME/ J2ME, maka aplikasi SMS *Scheduler* ini pun dibuat pada beberapa *platform* agar dapat digunakan oleh para pengguna *handphone*, yaitu *platform* Android, Blackberry, iOS, dan Java ME/ J2ME.

Selain beberapa hal yang akan ditangani pada aplikasi yang akan dibuat, ada pula beberapa hal yang tidak akan ditangani pada sistem yang akan dibuat karena hal tersebut berkaitan dengan masalah yang terjadi pada *hardware*, yaitu *mobile device*. Hal-hal yang tidak akan ditangani tersebut, yaitu :

1. Aplikasi tidak menangani masalah kegagalan pengiriman yang diakibatkan oleh *device* dalam keadaan mati, *hang*, atau rusak, baik itu kerusakan sistem maupun kerusakan *hardware device*
2. Aplikasi tidak menangani masalah kegagalan pengiriman yang diakibatkan oleh tidak adanya pulsa yang cukup untuk melakukan pengiriman pesan, adanya gangguan sinyal telepon, adanya gangguan pada operator atau *provider*.
3. Aplikasi tidak menangani masalah keterlambatan pengiriman pesan atau pengiriman pesan terlalu cepat, yang diakibatkan oleh waktu *device* yang tidak dalam kondisi benar, artinya waktu *device* tidak sesuai dengan waktu yang sedang berlangsung.

Pengembangan aplikasi untuk beberapa *platform* ini dilakukan dengan teknik *native*, dikarenakan cara tersebut memiliki beberapa kelebihan yang akan memudahkan ketika proses pengembangan aplikasi dibandingkan dengan menggunakan teknik *hybrid*. Salah satu kelebihan teknik *native* yang mendukung dalam pembuatan aplikasi SMS *Scheduler* adalah dapat mengakses fungsi-fungsi *device*, seperti fungsi kalender, kontak, dll.

Selain itu, masih terdapat beberapa kebutuhan-kebutuhan lain yang diperlukan untuk membangun aplikasi SMS *Scheduler* ini, dimana penjelasan lebih *detail* dari kebutuhan tersebut dapat dilihat pada dokumen Lampiran A - *Software Requirement Specification* (SRS).

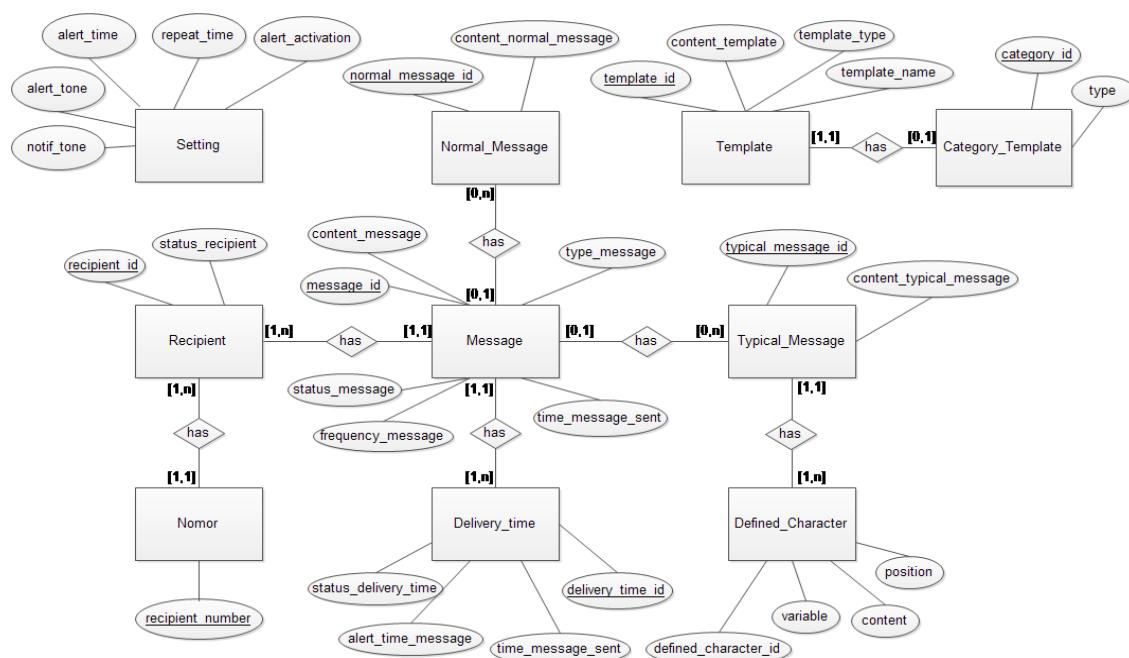
BAB IV

Perancangan

Pada tahap ini akan dilakukan perancangan untuk aplikasi yang akan dibuat berdasarkan hasil analisis yang telah dilakukan pada tahap sebelumnya. Diharapkan dari kegiatan tersebut didapatkan rancang bangun aplikasi yang dapat digunakan untuk melakukan pembuatan aplikasi (implementasi), baik dari segi rancangan data, rancangan arsitektur maupun rancangan proses untuk aplikasi yang akan dibuat. Untuk itu, dilakukan perancangan berdasarkan hasil analisis aplikasi yang sudah terdefinisi pada dokumen SRS di Lampiran A.

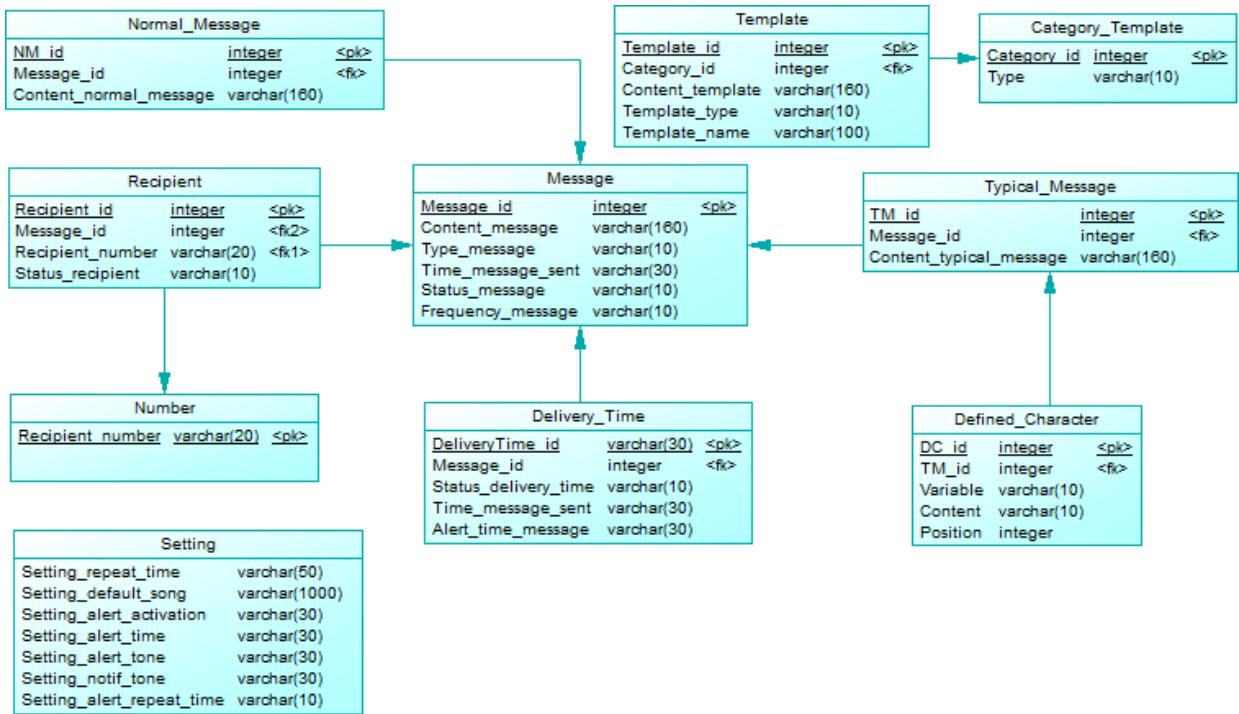
4.1 Perancangan Data

Perancangan *database* dari aplikasi SMS *Scheduler* ini bertujuan untuk memenuhi kebutuhan informasi dari pengguna aplikasi, menyediakan struktur informasi yang mudah dimengerti oleh pengguna serta mendukung kebutuhan pemrosesan dan beberapa objek kinerja dari suatu sistem *database*. Berdasarkan kebutuhan data yang telah dijelaskan pada dokumen SRS subbab Kebutuhan Data (*Logical Database Requirements*), didapatkan E-R Diagram pada gambar 11.



Gambar 11 . E-R Diagram SMS Scheduler

Berikut ini merupakan *Physical Data Model* (PDM) untuk aplikasi SMS *Scheduler* yang didapatkan dari E-R Diagram pada gambar 11.

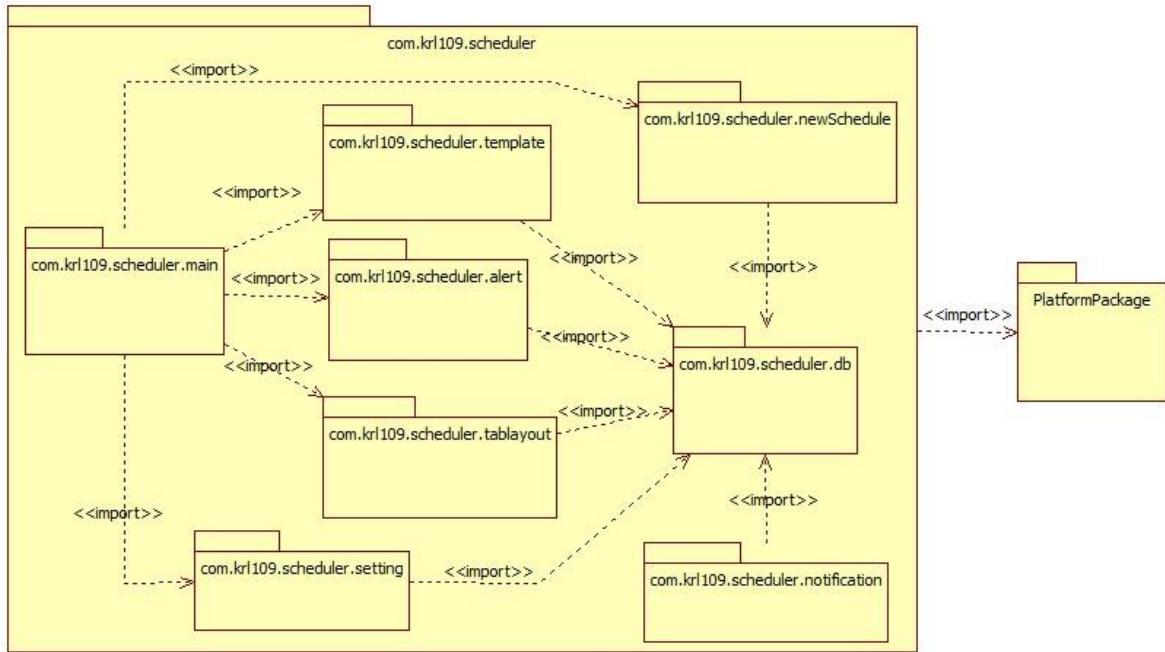


Gambar 12 . Physical Data Model (PDM) Database SMS Scheduler

4.2 Arsitektur Sistem

4.2.1 Perancangan Arsitektur Aplikasi SMS Scheduler

Perancangan arsitektur dirancang untuk memenuhi *requirement* yang telah didefinisikan pada bab sebelumnya. Gambar 13 menggambarkan pengelompokan kelas-kelas dan hubungan antar *package* menggunakan *tools package diagram*.

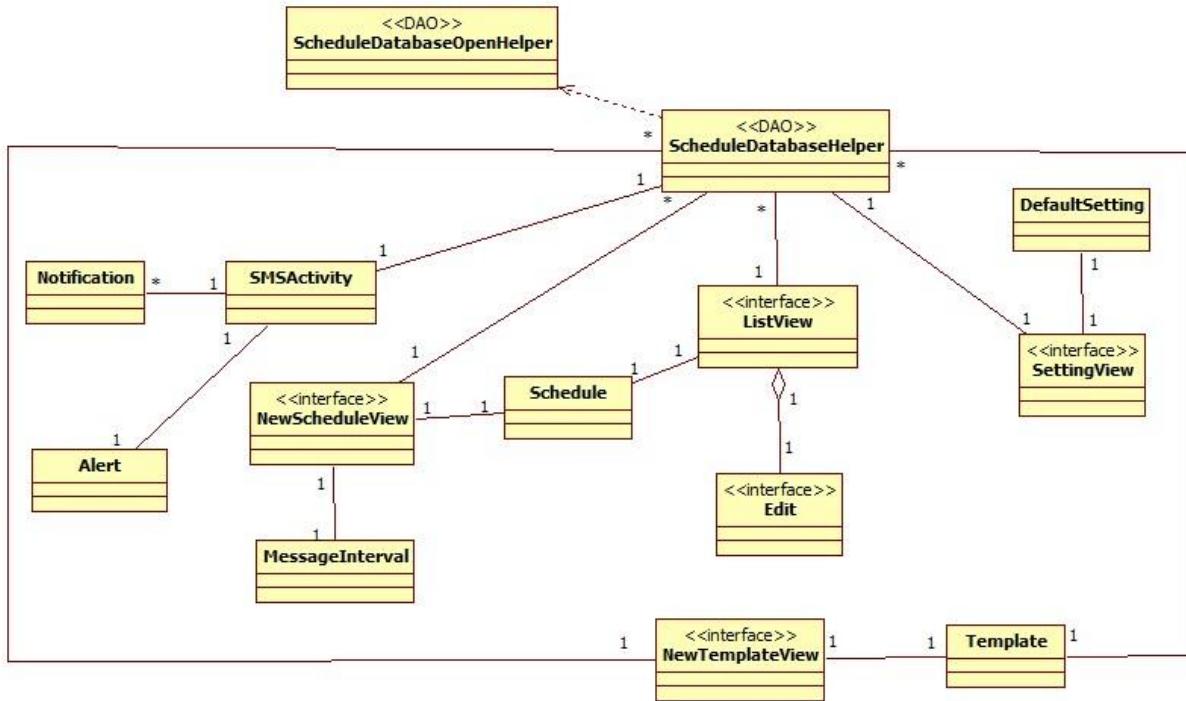
Gambar 13 . *Package diagram* aplikasi SMS *Scheduler*

Penjelasan dari *package diagram* diatas, dijelaskan pada tabel 19.

Tabel 19. Keterangan *Package Diagram*

Nama Package	Deskripsi
com.krl109.scheduler.tablayout	Package ini berisi <i>class-class</i> yang digunakan untuk menampilkan kategori pesan dan <i>editing schedule</i>
com.krl109.scheduler.newSchedule	Package ini berfungsi untuk menampung <i>class</i> yang berfungsi untuk membuat <i>schedule</i> baru, dan terdapat <i>class</i> yang berfungsi untuk mentrigger pesan yang sudah terjadwal untuk dikirim
com.krl109.scheduler.db	Package ini berfungsi untuk menangani data yang akan dimasukkan ke <i>database</i> , menangani data yang keluar dari <i>database</i> untuk di passing ke <i>class</i> lain dan untuk menangani data yang mengalami perubahan
com.krl109.scheduler.template	Package ini berisi <i>class-class</i> yang digunakan untuk menangani <i>template</i> pesan
com.krl109.scheduler.alert	Package ini berisi <i>class-class</i> yang digunakan untuk menangani <i>alert</i>
com.krl109.scheduler.setting	Package ini berisi <i>class-class</i> yang digunakan untuk menangani <i>setting</i> aplikasi
com.krl109.scheduler.main	Package ini berisi <i>class-class</i> yang digunakan untuk menangani menu utama aplikasi
com.krl109.scheduler.notification	Package ini berisi <i>class-class</i> yang digunakan untuk menangani <i>notifiacation</i>

Dibawah ini dijelaskan mengenai keterkaitan antar kelas-kelas yang terdapat pada aplikasi SMS *Scheduler* menggunakan *class diagram*.



Gambar 14 . *Class diagram SMS Scheduler*

Selain *class* yang ada pada *class diagram* diatas, ada pula *class* lain yang digunakan, yaitu *class* dari setiap *platform* yang dibuat. Diantaranya adalah *class* untuk mengakses komponen *user interface* yang diperlukan dan *class* untuk mengakses *alarm service*.

Untuk detail dari masing-masing *class* yang ada pada gambar *class diagram*, dijelaskan pada tabel 20 - 32.

Tabel 20. Keterangan *Class ScheduleDatabaseHelper*

<i>Class ScheduleDatabaseHelper</i>					
No	Nama Attribute	Tipe	Deskripsi	Nama Method	Deskripsi
1.	openHelper	ScheduleDatabase OpenHelper	variabel penampung instansiasi dari ScheduleDatabaseOpenHelper.	saveScheduleToMessage (long timesend, String[] data)	Digunakan untuk menyimpan <i>schedule</i> ke tabel message di <i>database</i>
				saveScheduleToTime (long timesend, String messageId)	Digunakan untuk menyimpan <i>schedule</i> ke tabel delivery time di <i>database</i>
2.	database Readable	SQLite Database	Inisiasi dari kelas SQLite Database untuk mengambil/membaca data dari <i>database</i>	saveScheduleToContact (String phoneNumbers)	Digunakan untuk menyimpan <i>schedule</i> ke tabel number di <i>database</i>
				saveScheduleToRecipient (Schedule schedule)	Digunakan untuk menyimpan <i>schedule</i> ke tabel recipient di <i>database</i>
3.	database WritableDatabase	SQLite Database	Inisiasi dari kelas SQLite Database untuk menginput data dari <i>database</i>	saveScheduleToType (Schedule schedule)	Digunakan untuk menyimpan tipe pesan dengan <i>id message</i> tertentu ke tabel message di <i>database</i>
				saveTemplateToTemplate (int categoryId, String[] data)	Digunakan untuk menyimpan <i>template</i> yang dibuat ke tabel template di <i>database</i>
				saveSettingToSetting (boolean alertActivation, long alertTime, Uri alertTone, Uri notifTone)	Digunakan untuk menyimpan setting yang diatur user ke tabel setting di <i>database</i>
				saveTemplateToCategory (String data)	Digunakan untuk menyimpan kategori <i>template</i> ke tabel category_template <i>database</i>
				getScheduleList()	Digunakan untuk mendapatkan semua <i>schedule</i> yang ada di <i>database</i>
				getMessageFromMessage (String messageId)	Digunakan untuk mengambil isi pesan dengan <i>id message</i> tertentu
				getRecipientFromRecipient (String messageId)	Digunakan untuk mengambil nomor <i>recipient</i> dengan <i>id message</i> tertentu

Class ScheduleDatabaseHelper	
	get messageIdFromTime (long timesend)
	getTime(String messageId)
	getRecipients(Cursor recipients)
	getMessage(String messageId)
	getMessageFrequency (String messageId)
	getMessageDatetime (String messageId)
	getMessageType(String messageId)
	getNotifTone()
	getAlertActivation()
	getAlertTime()
	getAlertTone()
	getCategoryId(String Category)
	getTemplateList()
	updateContent(String messageId, String data)

Class ScheduleDatabaseHelper					
				updateType(String type, String content, String messageId)	Digunakan untuk meng-update tipe <i>message</i> dengan tipe pesan sesuai dengan <i>content</i> pesan yang diubah oleh <i>user</i>
				updateTime(String messageId, String[] data)	Digunakan untuk meng-update <i>time</i> dengan waktu <i>schedule</i> dengan <i>id message</i> tertentu
				updateAlertActivation (Boolean alertActivation)	Digunakan untuk menyimpan perubahan aktivasi <i>alert</i> ke <i>database</i>
				updateAlertTime(Int alertTime)	Digunakan untuk menyimpan perubahan waktu untuk menampilkan <i>alert</i> ke <i>database</i>
				updateAlertTone(Uri alertTone)	Digunakan untuk menyimpan perubahan nada dering <i>alert</i> ke <i>database</i>
				updateNotifTone(Uri notifTone)	Digunakan untuk menyimpan perubahan nada dering <i>notification</i> ke <i>database</i>
				updateStatus(string messageId)	Digunakan untuk mengupdate status <i>schedule</i> di <i>database</i>
				deleteTime(String messageId)	Digunakan untuk menghapus waktu pengiriman dengan <i>message id</i> tertentu
				removeTemplate(int id)	Digunakan untuk menghapus <i>template</i> dengan <i>id template</i> tertentu dari <i>database</i>
				count(Cursor cursor)	Digunakan untuk menghitung banyaknya <i>record</i> dari <i>time</i> untuk <i>id message</i> tertentu

Tabel 21. Keterangan *Class ScheduleDatabaseOpenHelper*

Class ScheduleDatabaseOpenHelper					
<i>Class</i> ini berfungsi untuk membuat <i>database</i> bila ditemukan belum terdapat <i>database</i> aplikasi dalam <i>device</i> .					
No	Nama Attribute	Tipe	Deskripsi	Nama Method	Deskripsi
				create()	Digunakan untuk membuat <i>table</i> di <i>database</i>

Class ScheduleDatabaseOpenHelper				
			upgrade()	Digunakan untuk melakukan <i>upgrade table</i> di database

Tabel 22. Keterangan *Class Schedule*

Class Schedule					
Class schedule, sebagai penampung data jadwal.					
No	Nama Attribute	Tipe	Deskripsi	Nama Method	Deskripsi
1.	image	String	Untuk menampung data image/gambar.	getDateTime()	Digunakan untuk mengambil tanggal dan waktu <i>schedule</i>
				setDateTime(string dateTime)	Digunakan untuk <i>set</i> tanggal dan waktu <i>schedule</i>
2.	dateTime	String	Untuk menampung data waktu dan tanggal.	getRecipientNumbers()	Digunakan untuk mengambil nomor <i>recipient</i>
				setRecipientNumber(String recipientNumbers)	Digunakan untuk <i>set</i> nomor <i>recipient</i>
3.	Recipient Numbers	String	Untuk menampung data kontak penerima pesan dalam satu <i>schedule</i> .	getContentMessage()	Digunakan untuk mengambil isi pesan pada <i>schedule</i>
				setContentMessage(String contentMessage)	Digunakan untuk <i>set</i> isi pesan pada <i>schedule</i>
				getTimemillis()	Digunakan untuk mengambil tanggal dan waktu <i>schedule</i> yang sudah dikonversi ke <i>milisecond</i>
4.	Content Message	String	Untuk menampung data pesan.	setTimemillis(long timemillis)	Digunakan untuk <i>set</i> tanggal dan waktu <i>schedule</i> yang sudah dikonversi ke <i>milisecond</i>
				getFrequency()	Digunakan untuk mengambil frekuensi pengiriman <i>schedule</i>

Class Schedule					
5.	scheduleId	Int	Untuk menampung id schedule	setFrequency(String frequency)	Digunakan untuk <i>set</i> frekuensi pengiriman <i>schedule</i>
6.	timemillis	Long	untuk menampung data waktu dan tanggal yang telah diubah menjadi millisecond.	getStatus()	Digunakan untuk mengambil status <i>schedule</i>
				setStatus(String status)	Digunakan untuk <i>set</i> status <i>schedule</i>
7.	timesend	Long	Untuk menampung waktu pengiriman yang sudah diconvert ke dalam milisecond	getScheduleId()	Digunakan untuk mengambil <i>id schedule</i>
				setScheduleId(int scheduleId)	Digunakan untuk <i>set id schedule</i>
				getTimesend()	Digunakan untuk mengambil waktu pengiriman <i>schedule</i>
8.	frequency	String	Untuk menampung pengulangan pengiriman <i>schedule</i>	setTimesend(long timesend)	Digunakan untuk <i>set</i> waktu pengiriman <i>schedule</i>
				setMessageType(String message)	Digunakan untuk <i>set</i> tipe <i>message</i>
9.	status	String	Untuk menampung status <i>schedule</i>	getMessageType(String message)	Digunakan untuk mengambil tipe <i>message</i>
				pauseSchedule(string messageId)	digunakan untuk menghapus <i>schedule</i> dengan id <i>message</i> tertentu dari <i>alarm manager</i>
11.	freqtime	String	Untuk menampung banyaknya sisa <i>schedule</i> yang harus dikirimkan		
12.	type	String	Untuk menampung tipe <i>message</i> (<i>normal</i> atau <i>typical message</i>)		

Tabel 23. Keterangan *Class SMSActivity*

Class SMSActivity					
Class ini berfungsi untuk menangani pengiriman SMS					
No	Nama Attribute	Tipe	Deskripsi	Nama Method	Deskripsi
1.	phoneNumber	String	untuk menampung data kontak.	splitPhoneNumber(String phoneNumber, String message)	
2.	message	String	untuk menampung data pesan atau isi pesan.	sendSMS(String phoneNumber, String message)	Digunakan untuk melakukan pengiriman pesan sesuai dengan <i>schedule</i> yang sebelumnya telah dibuat, dengan nomor <i>recipient</i> dan isi pesan yang didapatkan dari <i>method</i> getMessageFromMessage dan <i>method</i> getRecipientFromRecipient

Tabel 24. Keterangan *Class MessageInterval*

Class MessageInterval					
Class ini digunakan untuk penanggulangan perhitungan repetition (pengulangan jadwal)					
No	Nama Attribute	Tipe	Deskripsi	Nama Method	Deskripsi
				repetition(PendingIntent pending, AlarmManager alarm, ScheduleDatabaseHelper databaseHelper, String[] data, long[] time)	digunakan untuk menangani pengulangan pengiriman <i>schedule</i>

Tabel 25. Keterangan Class NewScheduleView

Class NewScheduleView					
Class NewScheduleView berfungsi untuk pembuatan jadwal baru.					
No	Nama Attribute	Tipe	Deskripsi	Nama Method	Deskripsi
1.	timemillis	long	untuk menampung data waktu dan tanggal yang telah diubah menjadi millisecond.		
2.	recipient	EditText	untuk menampung masukkan kontak penerima pesan.		
3.	date	DatePicker	untuk menampung tanggal yang dipilih.		
4.	time	TimePicker	untuk menampung waktu yang dipilih.		
5.	contact_picker_result	final int	untuk menampung contact yang dipilih		
6.	Database Helper	ScheduleDatabase Helper	variable penampung instansiasi dari TimeListDatabase Helper.		
7.	btn_save	Button	untuk menampung aksi konfirmasi pengeditan.		
8.	btn_cancel	Button	untuk menampung aksi pembatalan proses pengeditan.		
9.	frequency	Spinner	untuk menampung data yang dipilih dari spinner.		
10.	data	String	untuk menampung pilihan menu edit.		

Tabel 26. Keterangan Class ListView

Class ListScheduleView					
Class ListScheduleView ini berfungsi untuk menangani <i>schedule</i> yang belum dilaksanakan, kemudian akan ditampilkan dalam bentuk <i>list</i> .					
No	Nama Attribute	Tipe	Deskripsi	Nama Method	Deskripsi
1.	date	DatePicker	untuk menampung tanggal yang dipilih.	optionEdit()	Digunakan untuk menampilkan opsi untuk melakukan <i>edit</i>
2.	time	TimePicker	untuk menampung waktu yang dipilih.		
3.	new_schedule	Button	menampung aksi untuk pembuatan jadwal baru.	datetimeDialog()	Digunakan untuk membuka <i>date and time picker</i>
4.	option_edit	String	array satu dimensi of string dengan panjang elemen yang dinamis		
5.	jawaban	String	untuk menampung pilihan menu.		
6.	listView	ListView	untuk menampilkan data secara list.		
7.	helper	ScheduleDatabase Helper	variable instansiasi dari ScheduleDatabaseHelper		
8.	schedule	List <Schedule>	untuk menampung data jadwal yang telah melalui proses perubahan ke dalam list.		

Tabel 27. Keterangan *Class Edit*

Class EditSchedule					
Class ini digunakan untuk menangani perubahan <i>schedule</i> , baik perubahan nomor penerima pesan, waktu pengiriman, maupun isi pesan.					
No	Nama Attribute	Tipe	Deskripsi	Nama Method	Deskripsi
				doLaunchContactPicker(View view)	Digunakan untuk membuka contact picker
				datetimeDialog()	Digunakan untuk membuka date and time picker

Tabel 28. Keterangan *Class Alert*

Class Alert					
Class ini digunakan untuk menampilkan <i>alert</i>					
No	Nama Attribute	Tipe	Deskripsi	Nama Method	Deskripsi
1.	alertActivation	Boolean	Untuk menampung aktivasi <i>alert</i>	showAlert(String recipient, String message)	Digunakan untuk menampilkan <i>alert</i>
2.	alertTone	Uri	Untuk menampung nada dering yang akan digunakan saat <i>alert</i> ditampilkan		
3.	timeAlert	Long	Menampung waktu untuk menampilkan <i>alert</i>		
4.	scheduleCount	Int	Untuk menampung jumlah <i>schedule</i> yang akan dikirim pada waktu yang sama		

Tabel 29. Keterangan *Class NewTemplateView*

<i>Class NewTemplateView</i>					
Class ini digunakan untuk menangani pembuatan <i>template</i> baru					
No	Nama Attribute	Tipe	Deskripsi	Nama Method	
1.	Database Helper	ScheduleDatabase Helper	Variabel penampung instansiasi dari ScheduleDatabaseOpenHelper.	openDialogCategory()	Digunakan untuk membuka pilihan kategori <i>template</i>
2.	def_character	String	Untuk menampung string khusus yang diinputkan oleh <i>user</i>		

Tabel 30. Keterangan *Class Template*

<i>Class Template</i>					
Class ini digunakan untuk menampilkan dan menghapus <i>template</i>					
No	Nama Attribute	Tipe	Deskripsi	Nama Method	Deskripsi
1.	helper	Schedule Database Helper	Variabel penampung instansiasi dari ScheduleDatabaseOpenHelper.		
2.	result	ArrayList	Untuk menampung <i>list template</i> dari <i>database</i>		

Tabel 31. Keterangan *Class SettingView*

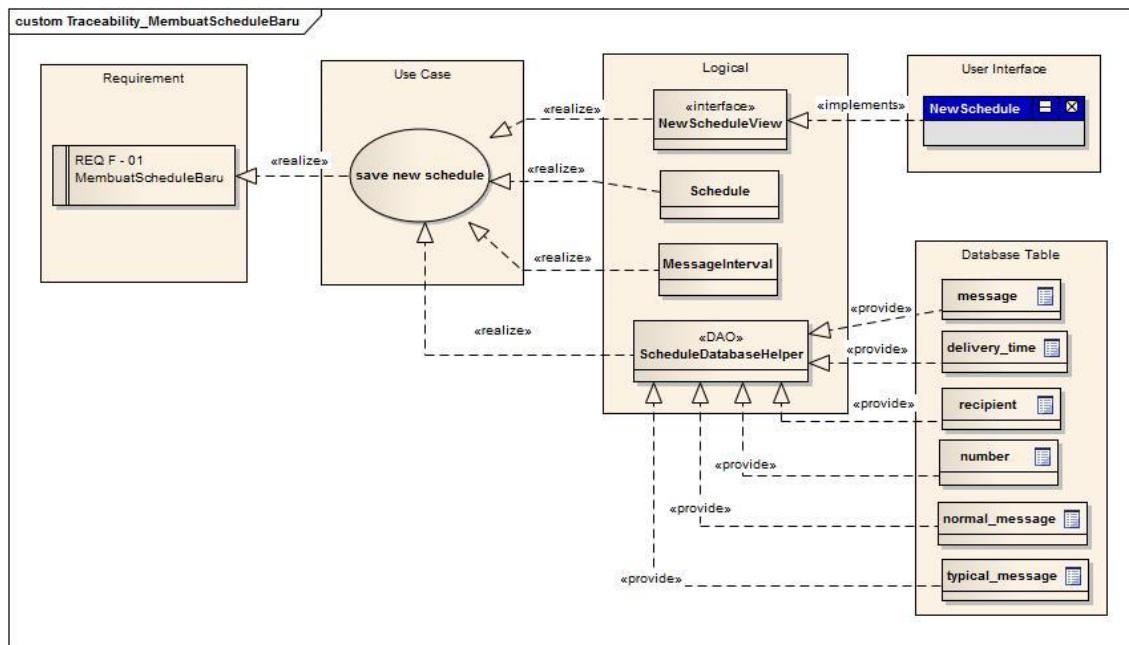
<i>Class SettingView</i>					
Class ini berfungsi untuk mengakses dan melakukan pengaturan aplikasi					
No	Nama Attribute	Tipe	Deskripsi	Nama Method	Deskripsi
1.	alert Activation	Boolean	Untuk menampung aktivasi <i>alert</i>		
2.	alertTime	String	Untuk menampung waktu ketika menampilkan <i>alert</i>		
3.	alertTone	Uri	Untuk menampung nada dering yang akan digunakan saat <i>alert</i> ditampilkan		
4.	notifTone	Uri	Untuk menampung nada dering yang akan digunakan saat <i>notification</i> ditampilkan		
5	defaultAlert Activation	Boolean	Untuk menampung nilai default untuk aktivasi <i>alert</i>		
6	defaultAlert Time	String	Untuk menampung nilai default untuk waktu ketika menampilkan <i>alert</i>		
7	defaultAlert Tone	Uri	Untuk menampung nilai default untuk nada dering yang akan digunakan saat <i>alert</i> ditampilkan		
8	defaultNotif Tone	Uri	Untuk menampung nilai default untuk nada dering yang akan digunakan saat <i>notification</i> ditampilkan		

Tabel 32. Keterangan *Class DefaultSetting*

<i>Class DefaultSetting</i>					
<i>Class ini berfungsi untuk nilai awal setting (default setting)</i>					
No	Nama Attribute	Tipe	Deskripsi	Nama Method	Deskripsi
1.	alert Activation	Boolean	Untuk menampung aktivasi <i>alert</i>	setDefaultAlertActivation (boolean alertActivation)	Digunakan untuk melakukan <i>set default</i> aktivasi <i>alert</i>
				setDefaultAlertTime (string alertTime)	Digunakan untuk melakukan <i>set default</i> waktu <i>alert</i>
2.	alertTime	String	Untuk menampung waktu ketika menampilkan <i>alert</i>	setDefaultAlertTone(Uri alertTone)	Digunakan untuk melakukan <i>set default</i> nada dering <i>alert</i>
				setDefaultNotifTone(Uri notifTone)	Digunakan untuk melakukan <i>set default</i> nada dering notifikasi
3.	alertTone	Uri	Untuk menampung nada dering yang akan digunakan saat <i>alert</i> ditampilkan	getDefaultAlertActivation()	Digunakan untuk mengambil <i>default</i> aktivasi <i>alert</i> yang sudah di <i>set</i>
				getDefaultAlertTime()	Digunakan untuk mengambil <i>default</i> waktu <i>alert</i> yang sudah di <i>set</i>
4.	notifTone	Uri	Untuk menampung nada dering yang akan digunakan saat <i>notification</i> ditampilkan	getDefaultAlertTone()	Digunakan untuk mengambil <i>default</i> nada dering <i>alert</i> yang sudah di <i>set</i>
				getDefaultNotifTone()	Digunakan untuk mengambil <i>default</i> nada dering notifikasi yang sudah di <i>set</i>

4.2.2 Perancangan Proses Aplikasi SMS Scheduler

Dari tahap analisis dan perancangan arsitektur aplikasi, telah didapatkan requirement, use case, kebutuhan data dan class yang diperlukan dalam pembuatan aplikasi SMS *Scheduler* yang kemudian digunakan sebagai acuan untuk membuat perancangan proses aplikasi.



Gambar 14. Keterkaitan antara *requirement*, *use case*, *class*, *database table* dan *user interface* aplikasi SMS *Scheduler* (example : fitur Membuat *Schedule* Baru)

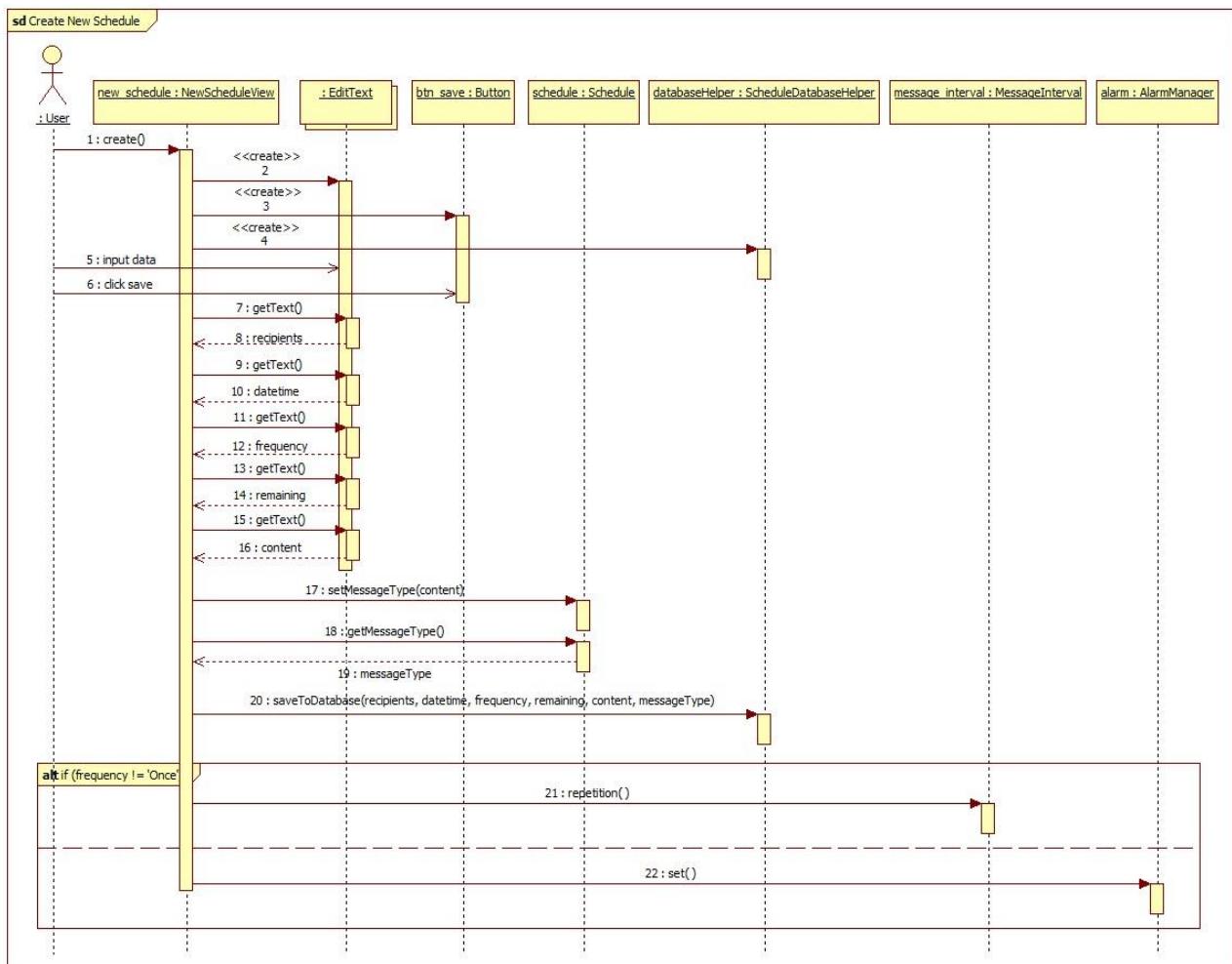
Pada gambar 14, merupakan contoh keterkaitan *requirement*, *use case*, kebutuhan data dan *class* untuk fitur atau fungsi membuat *schedule* baru.

1. Dari *requirement* yang sudah terdefinisi, yaitu REQ F - 01 : Membuat *Schedule* Baru, direalisasikan pada *use case* “save new schedule”.
Use case “save new schedule” menceritakan bagaimana sistem menangani proses pembuatan *schedule* baru (menerima input dari *user* untuk membuat *schedule* baru) sampai dengan melakukan penyimpanan *schedule* ke *database*.
2. Untuk merealisasikan *use case* tersebut, dibuatlah beberapa *class*, yaitu :
 - *NewScheduleView* : untuk menangani *interface* ketika membuat *schedule* baru, termasuk menerima *input* dari *user* dan memanggil *method* dari *class* *ScheduleDatabaseHelper* untuk melakukan penyimpanan *schedule* ke *database*.
 - *Schedule* : untuk menampung sementara data *schedule*

- `MessageInterval` : untuk penanggulangan perhitungan *repetition* (pengulangan pengiriman *schedule*)
- `ScheduleDatabaseHelper` : untuk mengakses *database*, baik melakukan *input*, menghapus maupun mengambil data yang sudah tersimpan. Tabel yang dipergunakan untuk proses membuat *schedule* baru ini adalah *table message*, *delivery_time*, *recipient*, *number*, *normal_message*, dan *typical message*.

Class-class tersebutlah yang menjadi acuan untuk menentukan objek-objek yang diperlukan untuk membuat rancangan proses. Tujuan perancangan proses adalah untuk menggambarkan interaksi atau komunikasi yang terjadi diantara objek-objek dalam suatu proses pada aplikasi SMS *Scheduler*. Perancangan proses digambarkan dengan *tools sequence diagram*. Proses yang digambarkan pada *sequence diagram* SMS *Scheduler* merupakan proses yang mendekati implementasi aplikasi.

4.2.2.1 Sequence Diagram Membuat Schedule



Gambar 15 . Sequence diagram New Schedule

Keterangan dari *sequence diagram New Schedule* pada gambar 15, dijelaskan pada tabel 33 - 43.

Tabel 33. Keterangan *sequence diagram New Schedule*

No.	SD - 1
Requirement	[REQ F - 01], [REQ CRT - 01], [REQ CRT - 02], [REQ CRT - 04], [REQ CRT - 05], [REQ CRT - 06], [REQ CRT - 07], [REQ CRT - 08], [REQ CRT - 09], [REQ CRT - 10]
Deskripsi	Sequence diagram ini menjelaskan mengenai proses dalam pembuatan <i>schedule</i> baru sampai dengan penyimpanan ke <i>database</i> .
Class yang terkait	NewScheduleView, EditText, Button, ScheduleDatabaseHelper, Schedule, MessageInterval, AlarmManager
Method yang terkait	getText(), setMessageType(), getMessageType(), saveToDatabase(), repetition(), set()
Logika Proses	<pre> Var recipients : String, //penampung data recipient datetime : String, //penampung data datetime frequency : String, //penampung data frequency remaining : String, //penampung data remaining content : String, //penampung data content databaseHelper : ScheduleDatabaseHelper, //inisiasi dari class ScheduleDatabaseHelper text : EditText, //inisiasi dari class EditText messageInterval : MessageInterval, //inisiasi dari class MessageInterval begin if(NewScheduleView.clicked) then if(btn_save.clicked) then //mendapatkan data yang sudah diinput user recipients <- text.getText(); datetime <- text.getText(); frequency <- text.getText(); remaining <- text.getText(); content <- text.getText(); schedule <- new Schedule(); schedule.set MessageType(content); messageType <- schedule.getMessageType(); //penyimpanan ke dalam database databaseHelper.saveToDatabase(recipients, datetime, frequency, remaining, content, messageType); if(frequency != 'Once') then repetition(); else alarm.set(); endif endif end </pre>

Tabel 34. Spesifikasi proses saveToDatabase

No.	SP - 1
Nama Method	saveToDatabase
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ CRT - 09]
Deskripsi	Method ini digunakan untuk melakukan penyimpanan <i>schedule</i> ke <i>database</i> .
Return	-
Parameter	-
Method yang terkait	convertTimeToMillisecond() : class Calendar saveScheduleToMessage(), getMessageIdFromMessage(), saveScheduleToType(), saveScheduleToTime(), saveScheduleToContact(), saveScheduleToRecipient() : class scheduleDatabaseHelper

No.	SP - 1
Nama Method	saveToDatabase
<pre> Logika Proses Procedure saveToDatabase(String recipients, String datetime, String frequency, String remaining, String content) var databaseHelper : ScheduleDatabaseHelper, //inisiasi dari class ScheduleDatabaseHelper schedule : Schedule, //inisiasi dari class Schedule type : String, //penampung type dari message timesend : long, //penampung waktu pengiriman pesan yang sudah dikonversi calendar : Calendar, //inisiasi dari class Calendar data[5] : Array of String dengan besar 5 messageId : String, //sebagai penampung messageId service : AlarmManager, begin //konversi waktu ke dalam millisecond timesend <- calendar.convertTimeToMillisecond(datetime); //memasukkan recipients, datetime, frequency, remaining, content, ke dalam variable data data[0] <- recipients; data[1] <- datetime; data[2] <- frequency; data[3] <- remaining; data[4] <- content; //apakah pesan terdapat string khusus atau tidak schedule.setMessageType(content); //mendapatkan tipe dari pesan type <- schedule.getMessageType(); //penyimpanan ke dalam entiti Message databaseHelper.saveScheduleToMessage(timesend, data[]); //mendapatkan messageId dari record terbaru di entiti Message messageId <- database.getMessageIdFromMessage(); //penyimpanan ke dalam entiti Normal_Message atau Typical_Message databaseHelper.saveScheduleToType(type, content, messageId); //penyimpanan ke dalam entiti Delivery_Time databaseHelper.saveScheduleToTime(timesend, messageId); //penyimpanan ke dalam entiti Number databaseHelper.saveScheduleToContact(recipients); //penyimpanan ke dalam entiti Recipient databaseHelper.saveScheduleToRecipient(recipients, messageId); end endprocedure </pre>	

Tabel 35. Spesifikasi proses getMessageIdFromMessage

No.	SP - 2
Nama Method	getMessageIdFromMessage
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ CRT - 09]
Deskripsi	Method ini digunakan untuk mengambil id message tertentu dari database.
Return	messageId : string
Parameter	-
Method yang terkait	moveToNextRecord(), getString() : class Cursor
Logika Proses	<pre> function getMessageIdFromMessage() </pre>

No.	SP - 2
Nama Method	get messageIdFromMessage
	<pre> var messageId : String, //menampung message_id dari database cursor : Cursor, //sebagai penampung dari data yang berhasil diambil dari database begin cursor <- SELECT all FROM Message WHERE message_id = (SELECT MAX(message_id) FROM Message); if(cursor.moveToFirst() = true) then messageId <- cursor.getString(cursor.getColumnIndex("message_id")); endif return messageId; end endfunction </pre>

Tabel 36. Spesifikasi proses setMessageType

No.	SP - 3
Nama Method	setMessageType
Nama Class	Schedule
Requirement	[REQ CRT - 10]
Deskripsi	Method ini digunakan untuk mengeset tipe pesan
Return	-
Parameter	message : String
Method yang terkait	contain()
Logika Proses	<pre> procedur setMessageType(message : String) var type : String //sebagai penampung type dari message untuk melakukan query read begin if(message.contains("%")) then type <- "typical"; else type <- "normal"; endif end endprocedure </pre>

Tabel 37. Spesifikasi proses getMessageType

No.	SP - 4
Nama Method	getMessageType
Nama Class	Schedule
Requirement	[REQ CRT - 10]
Deskripsi	Method ini digunakan untuk mengambil tipe pesan
Return	messageType : String
Parameter	-
Method yang terkait	-
Logika Proses	<pre> function getMessageType() var messageType : String, //sebagai penampung type dari message begin return messageType; end endfunction </pre>

Tabel 38. Spesifikasi proses saveScheduletoMessage

No.	SP - 5
Nama Method	saveScheduletoMessage
Nama Class	ScheduleDatabaseHelper

No.	SP - 5
Requirement	[REQ CRT - 09]
Deskripsi	<i>Method</i> ini digunakan untuk menyimpan <i>schedule</i> ke tabel message di database
Return	-
Parameter	timesend : long, data[5] : Array of String
Method yang terkait	
Logika Proses	<pre>procedure saveScheduleToMessage(timesend : long, data[5] : Array of String) begin INSERT INTO Message (time_message_sent, type_message, content_message, status_message, frequency_message) VALUES (timesend, data[7], data[2], data[5], data[3]); end endprocedure</pre>

Tabel 39. Spesifikasi proses saveScheduleToType

No.	SP - 6
Nama Method	saveScheduleToType
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ CRT - 09]
Deskripsi	<i>Method</i> ini digunakan untuk menyimpan tipe pesan dengan <i>id message</i> tertentu ke tabel message di database
Return	-
Parameter	type, content, messageId : string
Method yang terkait	
Logika Proses	<pre>procedure saveScheduleToType(type, content, messageId : String) begin if(type = "normal") then INSERT INTO Normal_Message (content_normal_message, message_id) VALUES (content, messageId); else INSERT INTO Typical_Message (content_typical_message, message_id) VALUES (content, messageId); endif end endprocedure</pre>

Tabel 40. Spesifikasi proses saveScheduleToTime

No.	SP - 7
Nama Method	saveScheduleToTime
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ CRT - 09]
Deskripsi	<i>Method</i> ini digunakan untuk menyimpan waktu <i>schedule</i> dengan <i>id message</i> tertentu ke tabel time di database
Return	-
Parameter	timesend : long, messageId : String
Method yang terkait	-
Logika Proses	<pre>procedure saveScheduleToTime(timesend : long, messageId : String) begin INSERT INTO Delivery_Time (message_id, status_delivery_time, time_message_sent, alert_time_message) VALUES (messageId, scheduled, timesend, alert); end endprocedure</pre>

Tabel 41. Spesifikasi proses saveScheduleToContact

No.	SP - 8
Nama Method	saveScheduleToContact
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ CRT - 09]
Deskripsi	Method ini digunakan untuk menyimpan <i>recipient schedule</i> ke tabel nomor di database.
Return	-
Parameter	phoneNumbers : String
Method yang terkait	hasMoreElements(), nextElement, moveToNextRecord() : class cursor
Logika Proses	<pre> procedure saveScheduleToContact(phoneNumbers : String) var st : StringTokenizer //sebagai penampung inisiasi terhadap class StringTokenizer begin st <- new StringTokenizer(phoneNumbers, ";"); while(st.hasMoreElements()) do tempPhoneNumber <- st.nextElement(); cursor <- SELECT all FROM contact WHERE contact_number = temptPhoneNumber; if(cursor.moveToNextRecord() = false) then INSERT INTO Number (recipient_number) VALUES (tempPhoneNumber); endif endwhile end endprocedure </pre>

Tabel 42. Spesifikasi proses saveScheduleToRecipient

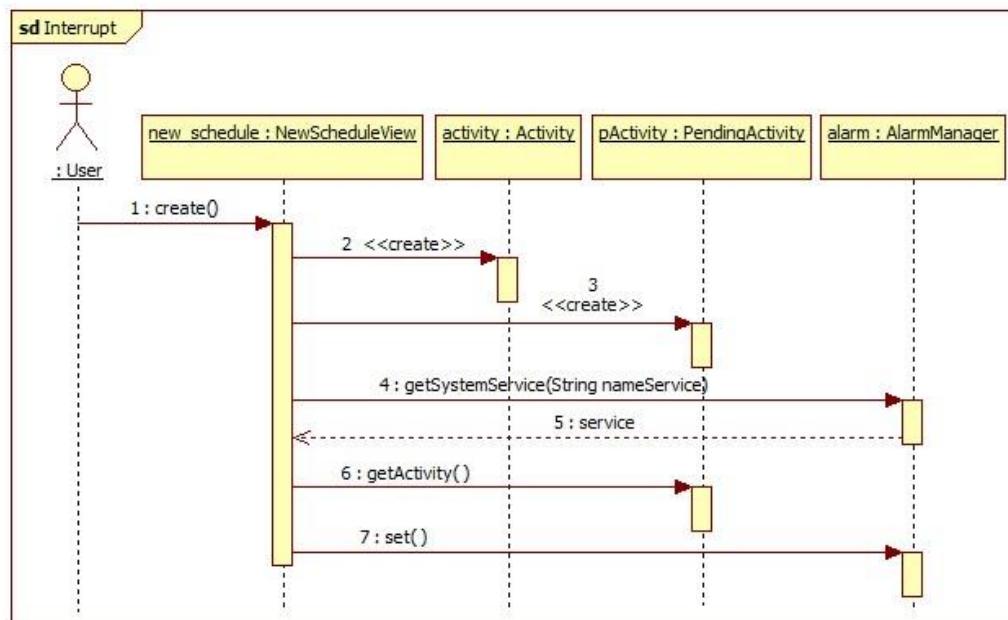
No.	SP - 9
Nama Method	saveScheduleToRecipient
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ CRT - 10]
Deskripsi	Method ini digunakan untuk menyimpan <i>recipient schedule</i> dengan <i>id message</i> tertentu ke tabel recipient di database
Return	-
Parameter	recipients, messageId : string
Method yang terkait	hasMoreElements(), nextElement()
Logika Proses	<pre> procedure saveScheduleToRecipient(recipients, messageId : String) var st : StringTokenizer //sebagai penampung inisiasi terhadap class StringTokenizer begin while (st.hasMoreElements()) do tempPhoneNumber <- st.nextElement(); INSERT INTO Recipient (message_id, recipient_number, status_recipient) VALUES (messageId, tempPhoneNumber, "pending"); endwhile end endprocedure </pre>

Tabel 43. Spesifikasi proses repetition

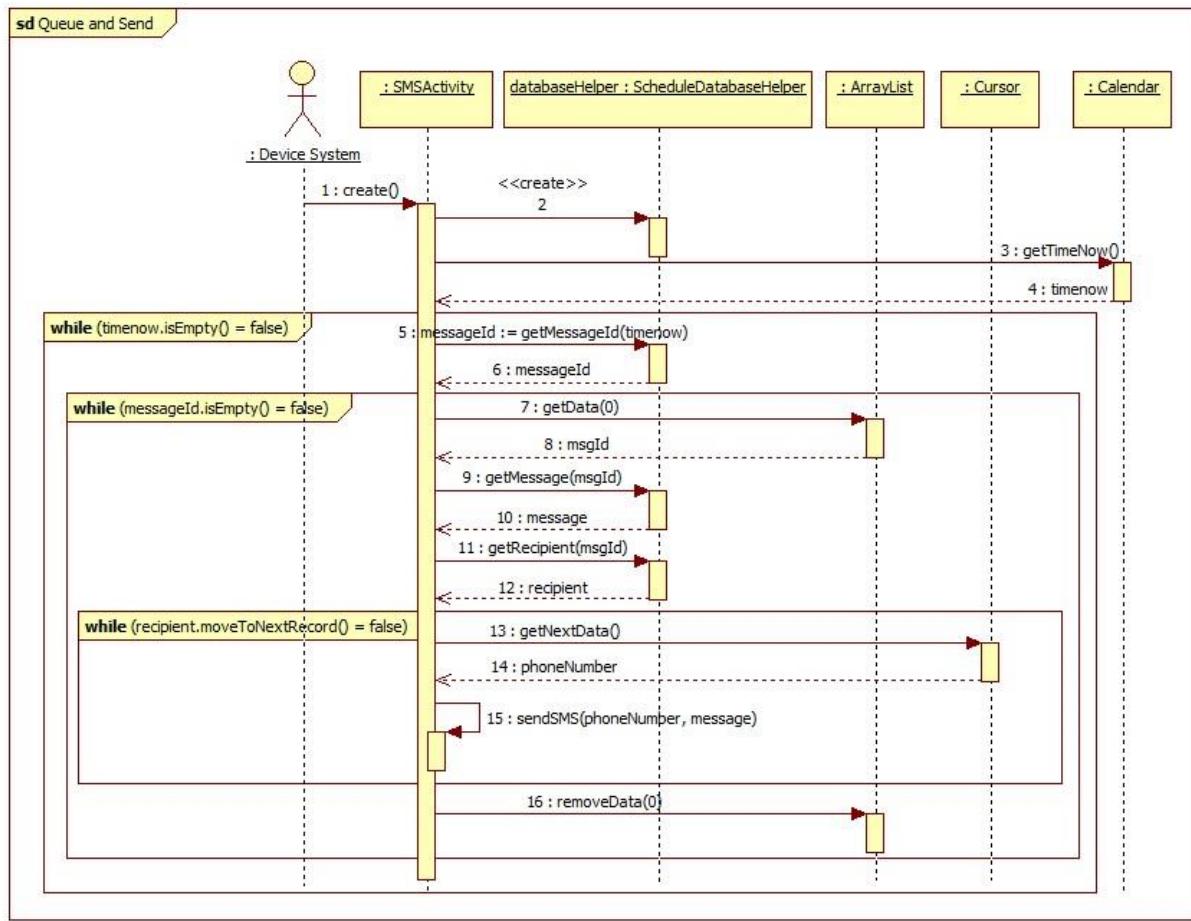
No.	SP - 10
Nama Method	repetition
Nama Class	MessageInterval
Requirement	[REQ CRT - 07]
Deskripsi	Method ini digunakan untuk menangani pengulangan pengiriman <i>schedule</i>
Return	-
Parameter	pending : PendingActivity, alarm : AlarmManager, databaseHelper :

No.	SP - 10
Nama Method	repetition
	ScheduleDatabaseHelper, data : Array of String. time : Array of Long
Method yang terkait	
Logika Proses	
<pre> procedure repetition(pending : PendingActivity, alarm : AlarmManager, databaseHelper : ScheduleDatabaseHelper, data : Array of String, time : Array of Long) var timesend : long //sebagai penampung data timesend frequency : String //sebagai penampung data frequency messageId : String //sebagai penampung data messageId interval : long //sebagai penampung data interval begin timesend <- time[0]; frequency <- data[0]; messageId <- data[1]; service <- AlarmManager.getSystemService(); //memanggil method interval untuk menghitung interval waktu sesuai dengan frequencynya interval <- interval(frequency); //set ke dalam AlarmManager sebagai interrupt service.setRepeating(AlarmManager.TYPE_ALARM_SERVICE, timesend, interval, pending); //insert ke dalam database sesuai dengan remaining yang diinputkan for(int remaining <- Integer.parseInt(data[2]) to remaining > 1) do //melakukan insert ke database ke table time databaseHelper.saveScheduleToTime(timesend, messageId); endfor end endprocedure </pre>	

4.2.2.1.1 Sequence Diagram Send Message



Gambar 16 . Sequence diagram interrupt



Gambar 17 . Sequence diagram queue dan send message

Keterangan dari *sequence diagram queue* pada gambar 16 dan 17, dijelaskan pada tabel 44 - 49.

Tabel 44. Keterangan *sequence diagram interrupt*

No.	SD - 2
Requirement	[REQ F - 05], [REQ SND - 01]
Deskripsi	Sequence ini digunakan untuk menggambarkan proses <i>interrupt</i>
Class yang terkait	NewScheduleView, Activity, PendingActivity, AlarmManager
Method yang terkait	getSystemService(), set() : class AlarmManager getActivity() : class PendingActivity
<pre> var activity : Activity //sebagai penampung inisiasi dari class Activity pending Activity : Pending Activity //sebagai penampung inisiasi dari class PendingActivity timesend : long //sebagai penampung waktu pengiriman yang sudah berbentuk millisecond begin activity <- new Activity(); //sebagai inisiasi dari class Activity pActivity <- new PendingActivity(); //sebagai inisiasi dari class Pending Activity alarm <- new AlarmManager(); //sebagai inisiasi dari class Alarm Manager service <- alarm.getSystemService(); //mendapatkan izin untuk dapat melakukan activity terhadap class tertentu activity.getActivity(); //melakukan set terhadap class AlarmManager yang berfungsi sebagai interrupt dengan waktu interrupt adalah timesend service.set(); end </pre>	

Tabel 45. Keterangan *sequence diagram queue* dan *send message*

No.	SD - 3
Requirement	[REQ F - 05], [REQ SND - 01], [REQ SND - 02], [REQ SND - 03], [REQ SND - 04]
Deskripsi	Sequence ini digunakan untuk menggambarkan pembuatan <i>queue</i> dan pelaksanaan <i>queue</i> tersebut
Class yang terkait	SMSActivity, ScheduleDatabaseHelper, ArrayList, Cursor, Calendar
Method yang terkait	set(), add(), isEmpty(), getMessageIdFromTime(), get(), getMessageFromMessage(), getRecipientFromRecipient(), moveToNext, sendSMS(), remove()
<pre> var calendar : Calendar //sebagai penampung inisiasi terhadap class Calendar timesend : Array of long //sebagai penampung waktu dalam bentuk millisecond messageId : Array of String //sebagai penampung messageId -> plural databaseHelper : ScheduleDatabaseHelper //sebagai penampung inisiasi terhadap class ScheduleDatabaseHelper msgId : String //sebagai penampung messageId -> singular message : String //sebagai penampung data message hasil fetch dari database cursorRecipient : Cursor //sebagai penampung data recipient begin //mendapatkan instance dari class Calendar calendar <- getInstance(); //mengeset nilai dari second calendar.set(calendarSecond, 0); //mengeset nilai dari millisecond calendar.set(calendarMilisecond, 0); //mengisi variabel timesend dengan waktu yang sudah berbentuk millisecond timesend.add(calendar.getTimeInMillis()); //loop sampai isi array timesend tidak ada lagi while(timesend.isEmpty() = false) do //mengambil messageId dari table time di database messageId <- databaseHelper.getMessageIdFromTime(timesend); while(messageId.isEmpty() = false) do //mengambil messageId dalam array dengan index 0 (awal) msgId <- messageId.get(0); //mengambil content message dari database dengan messageId tertentu message <- databaseHelper.getMessageFromMessage(msgId); //mendapatkan recipient dari database dengan messageId tertentu cursorRecipient <- databaseHelper.getRecipientFromRecipient(msgId); //loop sampai cursorRecipient habis while(cursorRecipient.moveToNext()) do //proses pengiriman sms sendSMS(cursorRecipient.getString(cursorRecipient.getColumnIndex("recipient_number")), message); endwhile //menghapus data array messageId dengan index 0 messageId.remove(0); endwhile //menghapus data array timesend dengan index 0 timesend.remove(0); endwhile end </pre>	

Tabel 46. Spesifikasi proses getMessageIdFromTime

No.	SP - 11
Nama Method	getMessageIdFromTime
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ F - 05], [REQ SND - 01]
Deskripsi	Method ini digunakan untuk mengambil <i>id message</i> berdasarkan waktu <i>schedule</i> yang sudah tersimpan di <i>database</i> .
Return	messageId : ArrayList of String
Parameter	timeMessageSent: long
Method yang terkait	moveToNextRecord(), add(), getColumnIndex()

No.	SP - 11
Nama Method	get messageIdFromTime
Logika Proses	
<pre> function getMessageIdFromTime(timeMessageSent : long) var cursor : Cursor, //penampung data yang diambil dari database messageId : ArrayList of String //penampung data id message yang didapat dari database begin //mendapatkan messageId dari table time dengan timesend tertentu cursor <- SELECT message_id from Delivery_Time WHERE time_message_timesend='timemillis'; while(cursor.moveToFirst()) do //menambahkan data ke dalam array list of string messageId.add(cursor.getString(cursor.getColumnIndex("message_id"))); endwhile return messageId; end end function </pre>	

Tabel 47. Spesifikasi proses getMessagFromMessage

No.	SP - 12
Nama Method	getMessageFromMessage
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ F - 05], [REQ SND - 01]
Deskripsi	Method ini digunakan untuk mengambil isi pesan dengan <i>id message</i> tertentu.
Return	message : String
Parameter	msgId : String
Method yang terkait	moveToNextRecord(), getString(), getColumnIndex()
Logika Proses	
<pre> function getMessageFromMessage(msgId : String) var message : String //penampung data message atau content yang didapat dari database cursor : Cursor //penampung data yang diambil dari database begin message <- null; //mendapatkan message dari table message dengan messageId tertentu cursor <- SELECT content_message from Message WHERE message_id = msgId; if(cursor.moveToFirst() = true) then message <- cursor.getString(cursor.getColumnIndex("content_message")); endif return message; end endfunction </pre>	

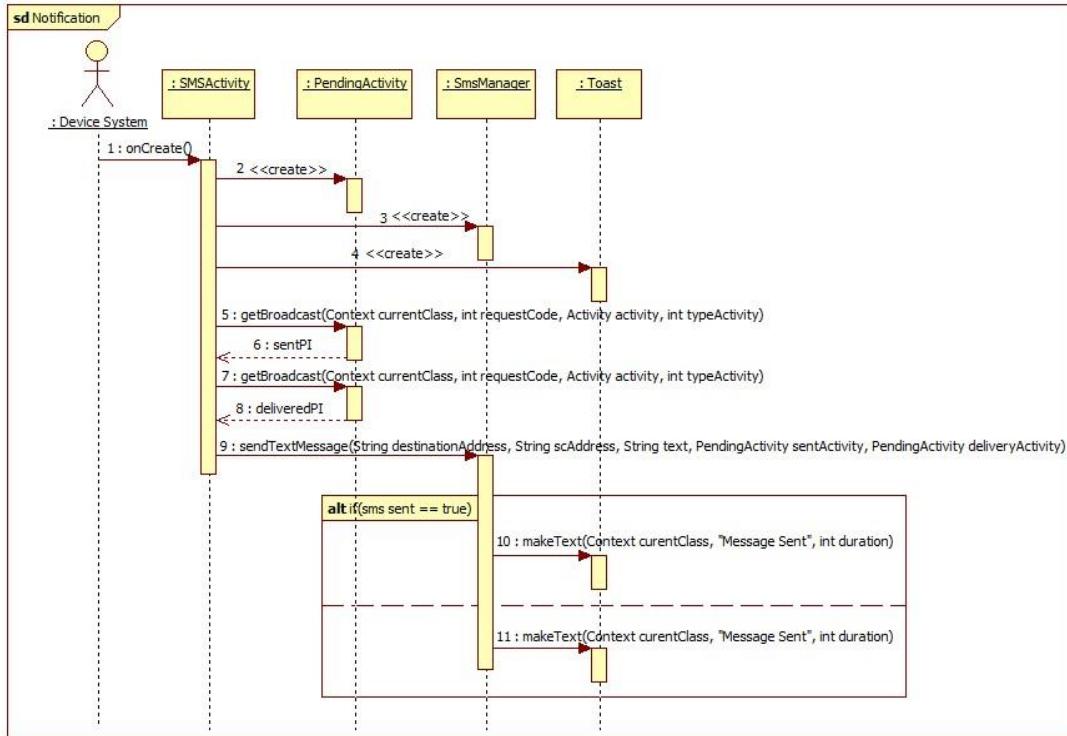
Tabel 48. Spesifikasi proses getRecipientFromRecipient

No.	SP - 13
Nama Method	getRecipientFromRecipient
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ F - 05], [REQ SND - 01]
Deskripsi	Method ini digunakan untuk mengambil nomor <i>recipient</i> dengan <i>id message</i> tertentu
Return	cursorRecipient : Cursor
Parameter	msgId : String
Method yang terkait	
Logika Proses	
<pre> function getRecipientFromRecipient (msgId : String) begin //mendapatkan semua record recipient yang memiliki messageId tertentu return SELECT all FROM Recipient where message_id = msgId; end endfunction </pre>	

Tabel 49. Spesifikasi proses sendSMS

No.	SP - 14
Nama Method	sendSMS
Nama Class	SMSActivity
Requirement	[REQ F - 05], [REQ SND - 01], [REQ SND - 02], [REQ SND - 03], [REQ SND - 04]
Deskripsi	<i>Method</i> ini digunakan untuk melakukan pengiriman pesan sesuai dengan <i>schedule</i> yang sebelumnya telah dibuat, dengan nomor <i>recipient</i> dan isi pesan yang didapatkan dari <i>method getMessageFromMessage</i> dan <i>method getRecipientFromRecipient</i>
Return	-
Parameter	phoneNumber, message : String
Method yang terkait	getDefault(), sendTextMessage(), contains(), get(), getInstance(), replace()
Logika Proses	<pre> procedure sendSMS(phoneNumber, message : String) var sms : SmsManager //inisiasi class SmsManager untuk dapat menggunakan method dari class ini sentPI : PendingActivity //penampung bila sms berhasil dikirimkan deliveredPI : PendingActivity //penampung bila sms gagal dikirimkan begin if (type.equals("typical"))then temp <- message.split("-"); integer y <- Integer.parseInt(temp[2]); device <- Calendar.getInstance(); Integer year <- device.get(calendarYear); If (message.contains("%%AGE%%"))then Integer age <- year - y; message.replace("%%AGE%%", age); endif if (message.contains("%%DATE%%"))then message.replace("%%DATE%%", temp[0]); endif if (message.contains("%%MONTH%%"))then message.replace("%%MONTH%%", temp[1]); endif if (message.contains("%%YEAR%%"))then message.replace("%%YEAR%%", year); endif endif //izin untuk dapat menggunakan fitur sms yang telah tersedia SmsManager sms <- SmsManager.getDefault(); //proses pengiriman sms oleh class yang telah disediakan sms.sendTextMessage(phoneNumber, null, message, sentPI, deliveredPI); end endprocedure </pre>

4.2.2.1.2 Sequence Diagram Notification



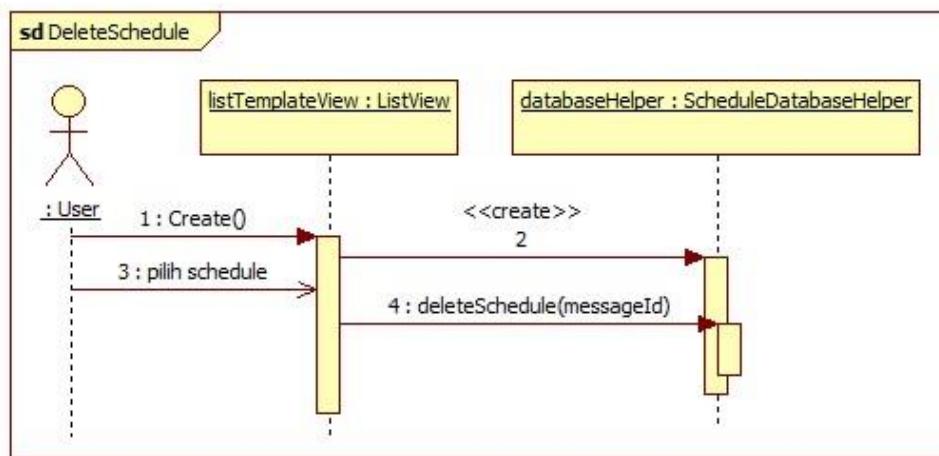
Gambar 18 . Sequence diagram notification

Keterangan dari *sequence diagram notification* pada gambar 18, dijelaskan pada tabel 50.

Tabel 50. Keterangan *sequence diagram notification*

No.	SD - 4
Requirement	[REQ F - 08] , [REQ NTF - 01], [REQ NTF - 02], [REQ NTF - 03]
Deskripsi	Sequence ini menggambarkan proses untuk menampilkan notifikasi pengiriman pesan, baik untuk pesan yang berhasil dikirim, maupun pesan yang gagal dikirim
Class yang terkait	SMSActivity, PendingActivity, SmsManager, Toast
Method yang terkait	getBroadcast(), sendTextMessage(), makeText()
<pre> var sentPI : PendingActivity //sebagai penampung inisiasi dari class PendingActivity deliveredPI : PendingActivity //sebagai penampung inisiasi dari class PendingActivity toast : Toast //sebagai penampung inisiasi class Toast destinationAddress : String //sebagai penampung nomor yang dituju scAddress : String // sebagai penampung isi pesan yang akan dikirim sms : SmsMnager //sebagai penampung inisiasi dari class SmsManager begin sentPI <- PendingActivity.getBroadcast(notification.class, 0, new Activity("SENT")); //sebagai inisiasi dari class PendingActivity deliveredPI <- PendingActivity.getBroadcast(notification.class, 0, new Activity("DELIVERED"), 0); //sebagai inisiasi dari class PendingActivity sms <- SmsMnager.getDefault(); //sebagai inisiasi dari class SmsManager //pengiriman sms sms.sendTextMessage(destinationAddress, scAddress, sentPI, deliveredPI); //bila sms berhasil dikirim if(sms.sent = true) then toast.makeText(this, "SMS Sent"); else toast.makeText(this, "SMS Failed"); endif end </pre>	

4.2.2.1.3 Sequence Diagram Menghapus Schedule



Gambar 19 . Sequence diagram menghapus schedule

Keterangan dari sequence diagram menghapus schedule pada gambar 19, dijelaskan pada tabel 51 - 58.

Tabel 51. Keterangan sequence diagram menghapus schedule

No.	SD - 5
Requirement	[REQ F - 03], [REQ DEL - 01], [REQ DEL - 02], [REQ DEL - 03]
Deskripsi	Sequence ini digunakan untuk menggambarkan proses untuk menghapus schedule dengan id tertentu dari database
Class yang terkait	ListView, ScheduleDatabaseHelper
Method yang terkait	deleteSchedule()
Logika Proses	<pre> var listScheduleView : ListView, //sebagai inisiasi dari class ListView messageId : String, //sebagai penampung schedule id yang dipilih user schedule : Schedule, //inisiasi dari class Schedule begin if(user.pilihschedule) then messageId <- listScheduleView.OnItemClick(user.pilihschedule); schedule.deleteSchedule(messageId); endif end </pre>

Tabel 52. Spesifikasi proses deleteRecipient

No.	SP - 15
Nama Method	deleteRecipient
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ F - 03], [REQ DEL - 01], [REQ DEL - 02], [REQ DEL - 03]
Deskripsi	Digunakan untuk menghapus recipient dengan id message tertentu
Return	-
Parameter	messageId : string
Method yang terkait	deleteRecipient(), getMessageType(), deleteMessageType(), getTime(), cancel(), deleteTime(), deleteMessage()
Logika Proses	<pre> Procedure deleteRecipient(messageId : String) messageId : String, //sebagai penampung messageId tableName : String, //sebagai penampung nama tabel </pre>

No.	SP - 15
	<pre> timesend : Cursor, //sebagai penampung waktu pengiriman databaseHelper : ScheduleDatabaseHelper, //sebagai inisisasi terhadap class ScheduleDatabaseHelper alarm : AlarmManager.getSystemService(), //sebagai penampung service yang didapatkan dari class Alarm begin //menghapus recipient databaseHelper.deleteRecipient(messageId); //mendapatkan nama table database sesuai dengan tipe message nya tableName <- database.getMessageType(messageId); //menghapus tipe dari message sesuai dengan tipenya databaseHelper.deleteMessageType(messageId, tableName); //mendapatkan waktu pengiriman dengan messageId tertentu timesend <- getTime(messageId); while(timesend.moveToNextRecord()) do alarm.cancel(timesend, messageId); endwhile //menghapus waktu pengiriman dengan messageId tertentu databaseHelper.deleteTime(messageId); //menghapus pesan dengan messageId tertentu databaseHelper.deleteMessage(messageId); end endprocedure </pre>

Tabel 53. Spesifikasi proses getMessageType

No.	SP - 16
Nama Method	getMessageType
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ F - 03], [REQ DEL - 01], [REQ DEL - 02], [REQ DEL - 03]
Deskripsi	Method ini digunakan untuk mengambil tipe message dengan id tertentu
Return	tableName : String
Parameter	messageId : string
Method yang terkait	
Logika Proses	<pre> function getMessageType(messageId : String) var tableName : String, //variabel untuk menampung nama tabel database messageType : String, //variable untuk menampung tipe message begin //mendapatkan tipe dari message dengan messageId tertentu messageType <- SELECT type_message FROM Message WHERE message_id = messageId; if(messageType = 'normal') then tableName <- 'Normal_Message'; else tableName <- 'Typical_Message'; endif return tableName; end endfunction </pre>

Tabel 54. Spesifikasi proses deleteRecipient

No.	SP - 17
Nama Method	deleteRecipient
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ F - 03], [REQ DEL - 01], [REQ DEL - 02], [REQ DEL - 03]
Deskripsi	Method ini digunakan untuk menghapus recipient dengan message id tertentu
Return	-
Parameter	messageId : string
Method yang terkait	

No.	SP - 17
Nama Method	deleteRecipient
Logika Proses	
<pre>Procedure deleteMessage(messageId : String) begin //menghapus recipient dengan messageId tertentu DELETE FROM Recipient WHERE message_id = messageId; end endprocedure</pre>	

Tabel 55. Spesifikasi proses deleteMessageType

No.	SP - 18
Nama Method	deleteMessageType
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ F - 03], [REQ DEL - 01], [REQ DEL - 02], [REQ DEL - 03]
Deskripsi	Method ini digunakan untuk menghapus tipe message dengan id tertentu
Return	-
Parameter	messageId, tableName : string
Method yang terkait	
Logika Proses	
<pre>Procedure deleteMessageType(messageId, tableName : String) begin //menghapus tipe message dari table Typical_Message atau Normal_Message dengan //messageId tertentu DELETE FROM tableName WHERE message_id = messageId; end endprocedure</pre>	

Tabel 56. Spesifikasi proses getTime

No.	SP - 19
Nama Method	getTime
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ F - 03], [REQ DEL - 01], [REQ DEL - 02], [REQ DEL - 03]
Deskripsi	Method ini digunakan untuk mengambil semua waktu pengiriman dengan message id tertentu
Return	Cursor : cursor
Parameter	MessageId : string
Method yang terkait	
Logika Proses	
<pre>function getTime(messageId : String) var cursor : Cursor, //sebagai penampung dari data yang berhasil diambil dari database begin //mendapatkan waktu pengiriman dengan messageId tertentu cursor <- SELECT all FROM Delivery_Time WHERE message_id = messageId; return cursor; end endfunction</pre>	

Tabel 57. Spesifikasi proses deleteTime

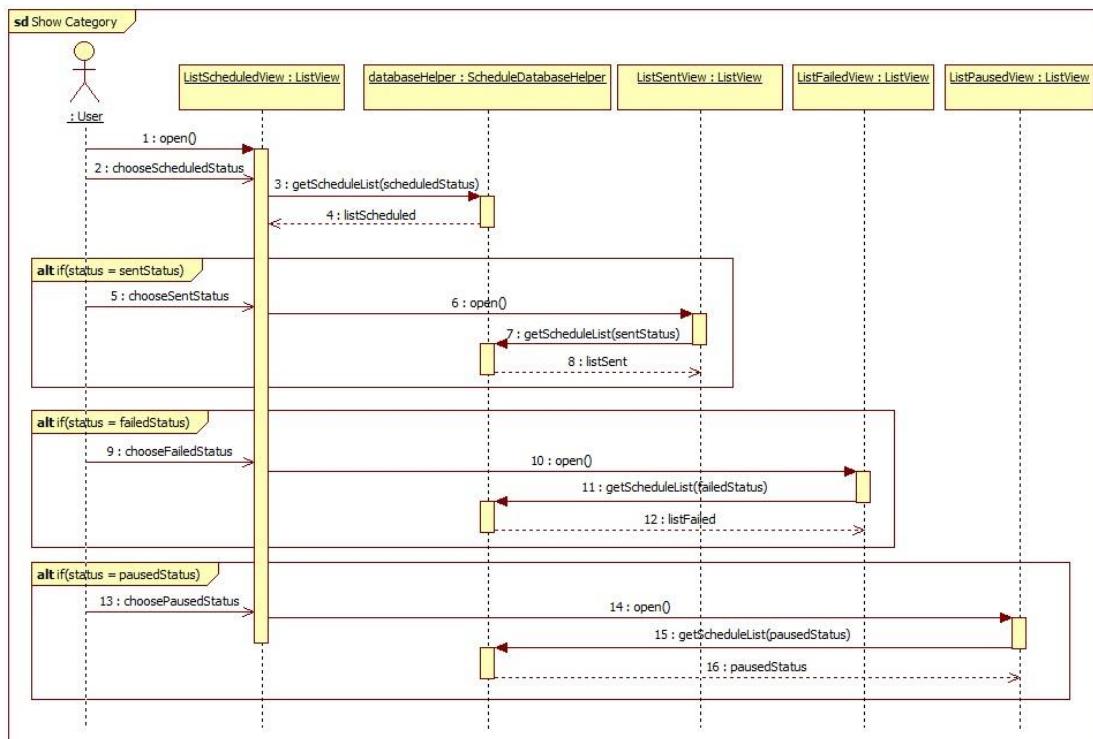
No.	SP - 20
Nama Method	deleteTime
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ F - 03], [REQ DEL - 01], [REQ DEL - 02], [REQ DEL - 03]
Deskripsi	Method ini digunakan untuk menghapus waktu pengiriman dengan message id tertentu
Return	-
Parameter	messageId : string

No.	SP - 20
Nama Method	deleteTime
Method yang terkait	
Logika Proses	
<pre>Procedure deleteTime(messageId : String) begin //menghapus waktu pengiriman dengan messageId tertentu DELETE FROM Delivery_Time WHERE message_id = messageId; end endprocedure</pre>	

Tabel 58. Spesifikasi proses deleteMessage

No.	SP - 21
Nama Method	deleteMessage
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ F - 03], [REQ DEL - 01], [REQ DEL - 02], [REQ DEL - 03]
Deskripsi	Method ini digunakan untuk menghapus message dengan message id tertentu
Return	-
Parameter	messageId : string
Method yang terkait	
Logika Proses	
<pre>Procedure deleteMessage(String messageId) begin //menghapus pesan dari table Message dengan messageId tertentu DELETE FROM Message WHERE message_id = messageId; end endprocedure</pre>	

4.2.2.1.4 Sequence Diagram Menampilkan Kategori Pesan



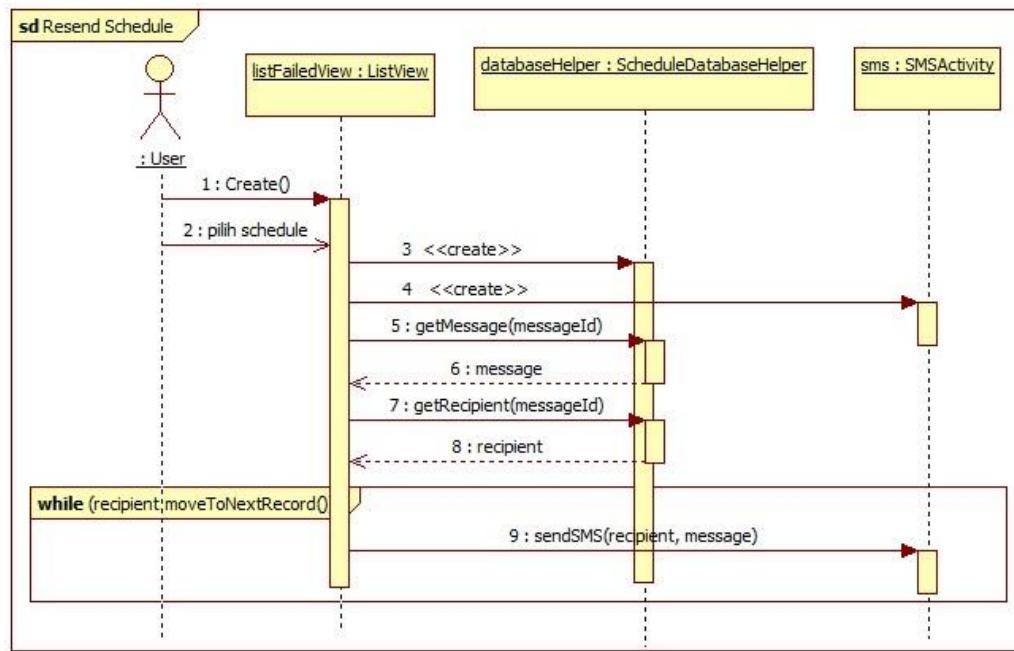
Gambar 20 . Sequence diagram Menampilkan Kategori Pesan

Keterangan dari *sequence diagram* Menampilkan Kategori Pesan pada gambar 20, dijelaskan pada tabel 59.

Tabel 59. Keterangan *sequence diagram* Menampilkan Kategori Pesan

No.	SD - 6
Requirement	[REQ F - 06] , [REQ FOL - 01], [REQ FOL - 02], [REQ FOL - 03], [REQ FOL - 04], [REQ FOL - 05], [REQ FOL - 06]
Deskripsi	Sequence ini digunakan untuk menggambarkan proses untuk menampilkan kelompok pesan sesuai dengan kategorinya
Class yang terkait	ListView, ScheduleDatabaseHelper
Method yang terkait	getScheduleList()
Var	<pre> cursor : Cursor //sebagai penampung inisiasi class Cursor databaseHelper : ScheduleDatabaseHelper //sebagai penampung inisiasi class ScheduleDatabaseHelper status: String //sebagai penampung status schedule listActivity : Activity begin if(user.chooseScheduledStatus) then status <- user.chooseScheduledStatus; listScheduled <- databaseHelper.getScheduleList(scheduledStatus); endif if(status = 'sentStatus') status <- user.chooseSentStatus; listSent <- databaseHelper.getScheduleList(sentStatus); endif if(status = 'failedStatus') status <- user.chooseFailedStatus; listFailed <- databaseHelper.getScheduleList(failedStatus); endif if(status = 'pausedStatus') status <- user.choosePausedStatus; pausedSent <- databaseHelper.getScheduleList(pausedStatus); endif end </pre>

4.2.2.1.5 Sequence Diagram Resend Schedule



Gambar 21 . Sequence diagram resend schedule

Keterangan dari *sequence diagram edit all schedule* pada gambar 21, dijelaskan pada tabel 60 - 63.

Tabel 60. Keterangan *sequence diagram resend schedule*

No.	SD - 7
Requirement	[REQ F - 01] , [REQ F - 06], [REQ F - 05]
Deskripsi	Sequence ini menggambarkan proses untuk melakukan <i>resend schedule</i> untuk <i>schedule</i> dengan status <i>failed</i>
Class yang terkait	ListView, ScheduleDatabaseHelper, SMSActivity
Method yang terkait	getRecipientFromRecipient(), getMessage(), moveToNextRecord(), getColumnIndex(), sendSMS()
Logika Proses	<pre> var recipient : Cursor, //sebagai penampung data semua recipient message : String, //sebagai penampung message databaseHelper : ScheduleDatabaseHelper, //sebagai inisisasi terhadap class ScheduleDatabaseHelper sms : SMSActivity, //sebagai inisiasi terhadap class SMSActivity messageId : String, //sebagai penampung schedule id yang dipilih user listFailedView : ListView, //sebagai inisiasi dari class ListView begin if(user.pilihschedule) then messageId <- listFailedView.OnItemClick(user.pilihschedule); //mendapat recipient dengan messageId tertentu recipient <- databaseHelper.getRecipient(messageId); //mendapat message dengan messageId tertentu message <- databaseHelper.getMessage(messageId); while(recipients.moveToNextRecord()) do sms.sendSMS(recipient, message); endwhile endif end </pre>

Tabel 61. Spesifikasi proses getRecipient

No.	SP - 22
Nama Method	getRecipient
Nama Class	ScheduleDatabasehelper
Requirement	[REQ F - 01] , [REQ F - 06], [REQ F - 05]
Deskripsi	Method ini digunakan untuk mengambil nomor recipient dengan id message tertentu
Return	cursor
Parameter	msgId : string
Method yang terkait	
Logika Proses	<pre>function getRecipientFromRecipient(msgId : string) begin //mendapatkan semua record recipient yang memiliki messageId tertentu return SELECT all FROM Recipient where message_id = messageId; end endfunction</pre>

Tabel 62. Spesifikasi proses getMessage

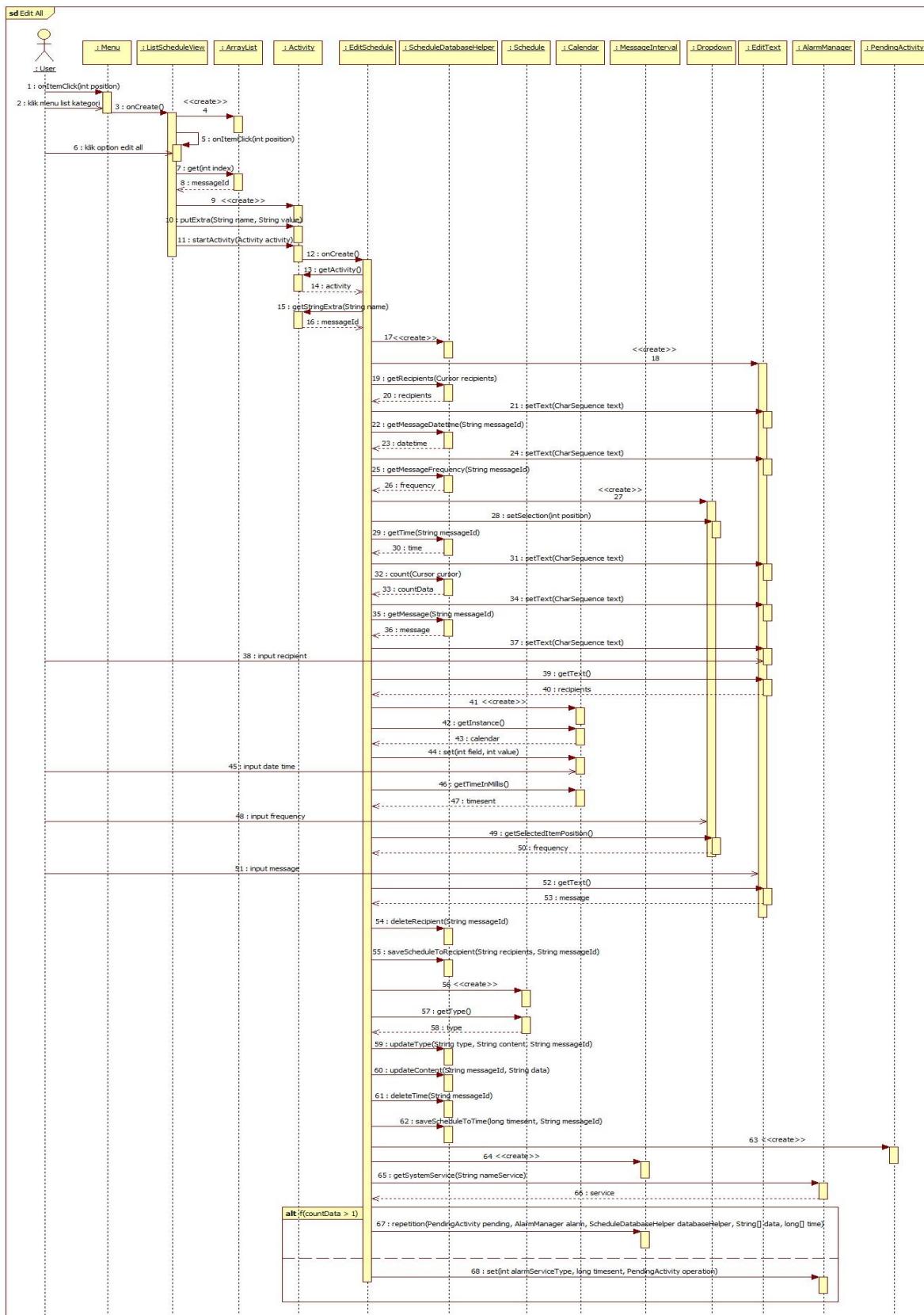
No.	SP - 23
Nama Method	getMessage
Nama Class	ScheduleDatabasehelper
Requirement	[REQ F - 01] , [REQ F - 06], [REQ F - 05]
Deskripsi	Method ini digunakan untuk mendapatkan isi pesan dengan id message tertentu
Return	message : string
Parameter	messageId : string
Method yang terkait	moveToNextRecord(), getColumnIndex(), getString()
Logika Proses	<pre>function getMessage(String messageId) var message : String, //sebagai penampung data message hasil fetch dari database cursor : Cursor, //sebagai penampung dari data yang berhasil diambil dari database begin message <- SELECT content_message FROM Message where message_id = messageId; return message; end endfunction</pre>

Tabel 63. Spesifikasi proses sendSMS

No.	SP - 24
Nama Method	sendSMS
Nama Class	SMSActivity
Requirement	[REQ F - 01] , [REQ F - 06], [REQ F - 05]
Deskripsi	Method ini digunakan untuk melakukan pengiriman pesan sesuai dengan schedule yang sebelumnya telah dibuat, dengan nomor recipient dan isi pesan yang didapatkan dari method getMessageFromMessage dan method getRecipientFromRecipient
Return	-
Parameter	phoneNumber, message : string
Method yang terkait	getDefault(), sendTextMessage()
Logika Proses	<pre>procedure sendSMS(phoneNumber, message : string) var sms : SmsManager, //inisiasi class SmsManager untuk dapat menggunakan method dari class ini sentPI : PendingActivity, //penampung bila sms berhasil dikirimkan deliveredPI : PendingActivity, //penampung bila sms gagal dikirimkan</pre>

```
begin
    //izin untuk dapat menggunakan fitur sms yang telah tersedia
    SmsManager sms <- SmsManager.getDefault();
    //proses pengiriman sms oleh class yang telah disediakan
    sms.sendTextMessage(phoneNumber, null, message, sentPI, deliveredPI);
end
endfunction
```

4.2.2.1.6 *Sequence Diagram Edit All Schedule*



Gambar 22 . Sequence diagram edit all

Keterangan dari *sequence diagram edit all schedule* pada gambar 22, dijelaskan pada tabel 64 - 78.

Tabel 64. Keterangan *sequence diagram edit all schedule*

No.	SD - 8
Requirement	[REQ F - 02], [REQ UPD - 01], [REQ UPD - 02], [REQ UPD - 03], [REQ UPD - 04], [REQ UPD - 05], [REQ UPD - 06], [REQ UPD - 07], [REQ UPD - 08], [REQ UPD - 09], [REQ UPD - 11], [REQ UPD - 12]
Deskripsi	<i>Sequence</i> ini menggambarkan proses untuk melakukan perubahan <i>schedule</i> secara keseluruhan (<i>edit all content schedule</i>)
Class yang terkait	Menu, ListScheduleView, ArrayList, Activity, EditSchedule, ScheduleDatabaseHelper, Cursor, EditText, Dropdown, Calendar, Schedule, MessageInterval, AlarmManager, PendingActivity
Method yang terkait	get(), putExtra(), startActivity(), getActivity(), getStringExtra, getRecipients(), setText(), getMessageDatetime(), setText(), getMessageFrequency(), setSelection(), getTime(), count(), getMessage(), getInstance(), getTimeInMillis(), getSelectedItemPosition(), deleteRecipient(), saveScheduleToRecipient(), getType(), updateType(), updateContent(), deleteTime(), saveScheduleToTime(), repetition(), set()
Var	<pre> databaseHelper : ScheduleDatabaseHelper freqtimes : MessageInterval schedule : ArrayList messageId, type, content : String activity : Activity pending : PendingActivity lview : ListScheduleView calendar : Calendar alarm : AlarmManager recipients, datetime, message, countData, frequency, time : String timesend : Long rcp, dt, msg, freq, cntdata : EditText Widget freq_dropdown : Dropdown Widget position : Integer data : Array of String //penampung untuk data-data yang akan disimpan di database schedule time : Array of Long </pre>
begin	<pre> schedule <- new ArrayList(); //mengambil nilai id dari schedule sesuai dari schedule yang dipilih messageId <- schedule.get(index); activity <- new Activity(); //menyimpan nilai terhadap activity baru activity.putExtra(messageId,messageId); startActivity(activity); activity <- activity.getActivity(); //mengambil nilai simpanan yang ada pada activity messageId <- activity.getStringExtra(); database <- new ScheduleDatabaseHelper(); //melakukan pengecekan apakah recipient sama dengan yang di database recipients <- databaseHelper.getRecipients(messageId); rcp.setText(recipients); //melakukan pengecekan apakah waktu dan tanggal sama dengan yang di database datetime <- databaseHelper.getMessageDatetime(messageId); dt.setText(datetime); //melakukan pengecekan apakah content message sama dengan yang di database message <- databaseHelper.getMessage(messageId); msg.setText(datetime); //melakukan pengecekan apakah waktu sama dengan yang di database time <- databaseHelper.getTime(messageId); //melakukan pengecekan apakah jumlah schedule yang akan dikirim sama dengan yang di database countData <- databaseHelper.getCount(cursor); </pre>

No.	SD - 8
	<pre> cntdata.setText(datetime); //melakukan pengecekan apakah nilai frequency sama dengan yang di database frequency <- databaseHelper.getMessageFrequency(messageId); freq_dropdown.setSelection(position); rcp.getText(); //mengambil nilai baru pada calendar calendar <- Calendar.getInstance(); Calendar.set(field : Integer , value : Integer); timemillis <- getTimeInMillis(); timesend <- timemillis; //mengambil nilai baru pada frequency frequency <- freq_dropdown.getSelectedItemPosition(position); //mengambil nilai baru pada content message message <- msg.getText(); //mengganti data recipient dengan data yang baru databaseHelper.deleteRecipient(messageId); databaseHelper.saveScheduleToRecipient(recipients, messageId); schedule <- new Schedule(); //mengganti jenis tipe dari schedule type <- schedule.getType(); //memperbaharui data pada database databaseHelper.updateType(type, content, messageId); databaseHelper.updateContent(messageId, data); databaseHelper.deleteTime(messageId); //menyimpan ulang waktu pada schedule databaseHelper.saveScheduleToTime(timemillis, timesend, messageId); activity <- new Activity(); activity.putExtra("timemillis", timesend); pending <- pending.getActivity(); //melakukan penyimpanan terhadap schedule yang siap dikirim alarm <- getSystemService(ALARM_SERVICE); frequities <- new MessageInterval(); alarm <- new AlarmManager(); //pengecekan terhadap jumlah schedule yang sama yang akan dikirim if(countData > 1) then frequities <- repetition(pending, alarm, databaseHelper, data, time); else alarm.set(); endif en </pre>

Tabel 65. Spesifikasi proses getRecipient

No.	SP - 25
Nama Method	getRecipient
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 02]
Deskripsi	Method ini digunakan untuk mengambil nomor-nomor <i>recipient</i> dengan <i>id message</i> tertentu
Return	recipient : String
Parameter	messageId : String
Method yang terkait	moveToNextRecord(), add(), getString(), getColumnIndex(), append(), get(), toString()
Logika Proses	<pre> function getRecipients(messageId : String) var recipients : String //penampung data recipients yang diambil dari database contact : ArrayList of String //penampung data contact sb : StringBuilder //sebagai inisiasi dari class StringBuilder begin while(recipient.moveToNext()) do contact.add(recipient.getString(recipient.getColumnIndex("recipient_number"))); endwhile sb.append(contact.get(0)); </pre>

No.	SP - 25
	<pre> for(int i <- 1 to i < contact.size() do sb.append(";"); sb.append(contact.get(i)); endfor recipients = sb.toString(); return recipients; end endfunction </pre>

Tabel 66. Spesifikasi proses getMessageDatetime

No.	SP - 26
Nama Method	getMessageDatetime
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 02]
Deskripsi	Method ini digunakan untuk mengambil waktu pengiriman dengan <i>id message</i> tertentu
Return	datetime : String
Parameter	messageId : String
Method yang terkait	moveToNextRecord(), getString(), getColumnIndex()
Logika Proses	<pre> function getMessageDatetime(messageId : String) var datetime : String //sebagai penampung data datetime hasil fetch dari database cursor : Cursor //sebagai penampung dari data yang berhasil diambil dari database begin cursor <- SELECT time_message_sent FROM Message WHERE message_id='messageId'; if(cursor.moveToFirst() = true) then datetime <- cursor.getString(cursor.getColumnIndex("time_message_sent")); endif return datetime; end endfunction </pre>

Tabel 67. Spesifikasi proses getMessageFrequency

No.	SP - 27
Nama Method	getMessageFrequency
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 02]
Deskripsi	Method ini digunakan untuk mengambil frekuensi pengiriman dengan <i>id message</i> tertentu
Return	frequency : String
Parameter	messageId : String
Method yang terkait	moveToNextRecord(), getString(), getColumnIndex()
Logika Proses	<pre> function getMessageFrequency(messageId : String) var frequency : String //sebagai penampung data frequency hasil fetch dari database cursor : Cursor //sebagai penampung dari data yang berhasil diambil dari database begin cursor <- SELECT frequency_message FROM Message WHERE message_id='messageId'; if(cursor.moveToFirst() = true) then frequency <- cursor.getString(cursor.getColumnIndex("frequency_message")); endif return frequency; end endfunction </pre>

Tabel 68. Spesifikasi proses getTime

No.	SP - 28
Nama Method	getTime
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 02]
Deskripsi	Method ini digunakan untuk mendapatkan semua <i>time</i> dengan <i>id message</i> tertentu
Return	cursor : Cursor
Parameter	messageId : String
Method yang terkait	
Logika Proses	<pre>function getTime(messageId : String) var cursor : Cursor //sebagai penampung dari data yang berhasil diambil dari database begin cursor <- SELECT all FROM Delivery_Time WHERE message_id='messageId'; return cursor; end endfunction</pre>

Tabel 69. Spesifikasi proses count

No.	SP - 29
Nama Method	count
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 02]
Deskripsi	Method ini digunakan untuk menghitung banyaknya <i>record</i> yang tersimpan
Return	count : integer
Parameter	cursor : cursor
Method yang terkait	getCount()
Logika Proses	<pre>function count(Cursor cursor) : int var count : int, //sebagai penampung berapa banyak row record yang ada pada cursor begin count <- cursor.getCount(); return count; end endfunction</pre>

Tabel 70. Spesifikasi proses getMessage

No.	SP - 30
Nama Method	getMessage
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 02]
Deskripsi	Method ini digunakan untuk mendapatkan isi pesan dengan <i>id message</i> tertentu
Return	message : String
Parameter	messageId : String
Method yang terkait	moveToNextRecord(), getString(), getColumnIndex()
Logika Proses	<pre>function getMessage(messageId : String) var message : String, //sebagai penampung data message hasil fetch dari database cursor : Cursor, //sebagai penampung dari data yang berhasil diambil dari database begin cursor <- SELECT content_message FROM Message where message_id='messageId'; if(cursor.moveToFirst() = true) then message <- cursor.getString(cursor.getColumnIndex("content message")); end endfunction</pre>

No.	SP - 30
	<pre> endif return message; end endfunction </pre>

Tabel 71. Spesifikasi proses deleteRecipient

No.	SP - 31
Nama Method	deleteRecipient
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 11]
Deskripsi	Method ini digunakan untuk menghapus nomor recipient dengan id message tertentu dari database
Return	-
Parameter	messageId : String
Method yang terkait	
Logika Proses	<pre> procedure deleteRecipient(messageId : String) begin DELETE FROM Recipient WHERE message_id = messageId; end endprocedure </pre>

Tabel 72. Spesifikasi proses saveScheduleToRecipient

No.	SP - 32
Nama Method	saveScheduleToRecipient
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 11]
Deskripsi	Method ini digunakan untuk menyimpan schedule yang diubah ke tabel recipient di database
Return	-
Parameter	recipients, messageId : String
Method yang terkait	hasMoreElements(), nextElement()
Logika Proses	<pre> procedure saveScheduleToRecipient(recipients, messageId : String) var st : StringTokenizer //sebagai inisiasi terhadap class StringTokenizer tempPhoneNumber : String //sebagai penampung data nomor recipient begin StringTokenizer st <- new StringTokenizer(recipients, ";"); while (st.hasMoreelements()) do tempPhoneNumber <- st.nextElement(); INSERT INTO Recipient (message_id, recipient_number, status_recipient) VALUES (messageId, tempPhoneNumber, 'pending'); endwhile end endprocedure </pre>

Tabel 73. Spesifikasi proses getMessageType

No.	SP - 33
Nama Method	getMessageType
Nama Class	Schedule
Requirement	[REQ UPD - 01], [REQ UPD - 12]
Deskripsi	Method ini digunakan untuk mengambil tipe pesan
Return	messageType : String
Parameter	-

No.	SP - 33
Method yang terkait	-
Logika Proses	
<pre>function getMessageType() begin return messageType; end endfunction</pre>	

Tabel 74. Spesifikasi proses updateType

No.	SP - 34
Nama Method	updateType
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 12]
Deskripsi	Method ini digunakan untuk meng-update tipe message dengan tipe pesan sesuai dengan content pesan yang diubah oleh user
Return	-
Parameter	type, content, messageId : String
Method yang terkait	
Logika Proses	
<pre>procedure updateType(type, content, messageId : String) begin if(type = "normal") then UPDATE Normal_Message SET content_normal_message='content' WHERE message_id='messageId'; else UPDATE Typical_Message SET content_typical_message='content' WHERE message_id='messageId'; endif end endprocedure</pre>	

Tabel 75. Spesifikasi proses updateContent

No.	SP - 35
Nama Method	updateContent
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 12]
Deskripsi	Method ini digunakan untuk meng-update content pesan dengan content pesan yang diubah oleh user
Return	-
Parameter	messageId, data : String
Method yang terkait	
Logika Proses	
<pre>procedure updateContent(data, messageId : String) begin UPDATE Message SET content_message='data' where message_id = messageId; end endprocedure</pre>	

Tabel 76. Spesifikasi proses deleteTime

No.	SP - 36
Nama Method	deleteTime
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 11]
Deskripsi	Method ini digunakan untuk menghapus time dengan id message tertentu
Return	-

No.	SP - 36
Nama Method	deleteTime
Parameter	messageId : String
Method yang terkait	
Logika Proses	<pre>procedure deleteTime(messageId : String) begin DELETE FROM Delivery_Time WHERE message_id = messageId; end endprocedure</pre>

Tabel 77. Spesifikasi proses saveScheduleToTime

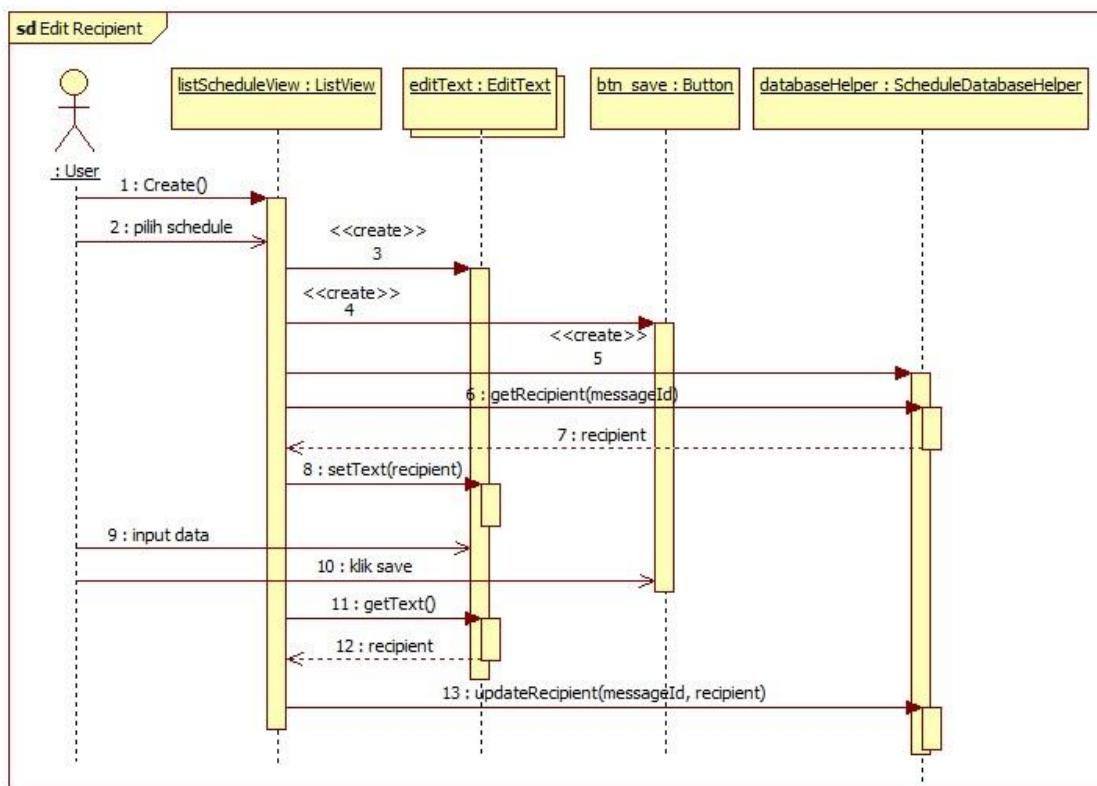
No.	SP - 37
Nama Method	saveScheduleToTime
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 11]
Deskripsi	Method ini digunakan untuk menyimpan waktu schedule ke tabel time di database
Return	-
Parameter	timesend : long messageId : String
Method yang terkait	
Logika Proses	<pre>procedure saveScheduleToTime(timesend : long, messageId : String) begin INSERT INTO Delivery_Time (message_id, status_delivery_time, time_message_sent, alert_time_message) VALUES (messageId, 'scheduled', timesend, alert); end endprocedure</pre>

Tabel 78. Spesifikasi proses repetition

No.	SP - 38
Nama Method	repetition
Nama Class	MessageInterval
Requirement	[REQ UPD - 01], [REQ UPD - 09]
Deskripsi	Method ini digunakan untuk menangani pengulangan pengiriman schedule
Return	-
Parameter	pending : PendingActivity, alarm : AlarmManager, databaseHelper : ScheduleDatabaseHelper, data : Array of String. time : Array of Long
Method yang terkait	setRepeating(), parseInt(), add(), saveScheduleToTime()
Logika Proses	<pre>procedure repetition(pending : PendingActivity, alarm : AlarmManager, databaseHelper : ScheduleDatabaseHelper, data : Array of String, time : Array of Long) var timesend : long //sebagai penampung data timesend frequency : String //sebagai penampung data frequency messageId : String //sebagai penampung data messageId interval : long //sebagai penampung data interval begin timesend <- time[0]; frequency <- data[0]; messageId <- data[1]; service <- AlarmManager.getServiceSystem(); //memanggil method interval untuk menghitung interval waktu sesuai dengan frequencynya interval <- interval(frequency); //set ke dalam AlarmManager sebagai interrupt service.setRepeating(AlarmManager.typeAlarmServices, timesend, interval, pending);</pre>

No.	SP - 38
Nama Method	repetition
	//insert ke dalam database sesuai dengan remaining yang diinputkan for(int remaining <- Integer.parseInt(data[2]) to remaining > 1) do //melakukan insert ke database ke table time databaseHelper.saveScheduleToTime(timesend, messageId); endfor end endprocedure

4.2.2.1.7 Sequence Diagram Edit Recipient Schedule



Gambar 23 . Sequence diagram edit recipient

Keterangan dari *sequence diagram edit recipient* pada gambar 23, dijelaskan pada tabel 79 - 85

Tabel 79. Keterangan *sequence diagram edit recipient*

No.	SD - 9
Requirement	[REQ F - 02], [REQ UPD - 01], [REQ UPD - 02], [REQ UPD - 03], [REQ UPD - 04], [REQ UPD - 05], [REQ UPD - 10]
Deskripsi	Sequence ini menggambarkan proses untuk melakukan perubahan penerima pesan dari <i>schedule</i> yang sebelumnya sudah dibuat (<i>edit recipient</i>)
Class yang terkait	ListView, ScheduleDatabaseHelper, EditText, Button
Method yang terkait	getRecipient(), setText(), getText(), updateRecipient()
	<pre> var listScheduleView : ListView, //sebagai inisiasi dari class ListView editText : EditText, //sebagai inisiasi dari class EditText btn_save : Button, //sebagai inisiasi dari class Button, yang memungkinkan untuk trigger </pre>

No.	SD - 9
	<pre> proses selanjutnya databaseHelper : ScheduleDatabaseHelper, //sebagai inisiasi terhadap class ScheduleDatabaseHelper recipient : String, //sebagai penampung data recipient messageId : String, //sebagai penampung schedule id yang dipilih user begin //user memilih schedule yang akan diedit if(user.pilihschedule) then messageId <- listScheduleView.OnItemClick(user.pilihschedule); recipient <- databasehelper.getRecipient(messageId); editText.setText(recipient); //user memasukkan data recipient baru if(user.inputdata) then //user klik button save if(btn_save.clicked) then recipient <- editText.getText(); databaseHelper.updateRecipient(messageId, recipient); endif endif endif end </pre>

Tabel 80. Spesifikasi proses getRecipient

No.	SP - 39
Nama Method	getRecipient
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 02]
Deskripsi	Method ini digunakan untuk mengambil nomor-nomor <i>recipient</i> dengan <i>id message</i> tertentu
Return	recipient : String
Parameter	messageId : String
Method yang terkait	getStringRecipientFromRecipient(), getStringRecipients()
Logika Proses	<pre> function getRecipients(messageId : String) var cursorRecipient : Cursor, //sebagai penampung data recipient hasil fetch dari database recipient : String, //sebagai penampung recipient hasil begin //mendapatkan record recipient dari table recipient dengan messageId tertentu cursorRecipients <- databaseHelper.getRecipientFromRecipient(messageId); //menjadikan data recipients menjadi satu string recipient <- getStringRecipients(cursorRecipients); end endfunction </pre>

Tabel 81. Spesifikasi proses updateRecipient

No.	SP - 40
Nama Method	updateRecipient
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 02]
Deskripsi	Method ini digunakan untuk meng-update data <i>recipient</i> untuk <i>id message</i> tertentu
Return	-
Parameter	messageId, recipient : String
Method yang terkait	
Logika Proses	<pre> Procedure updateRecipient(messageId, recipient : String) var messageId : String, //sebagai penampung messageId recipientsText : EditText, //sebagai inisiasi dari class EditText databaseHelper : ScheduleDatabaseHelper, //sebagai inisiasi terhadap class </pre>

No.	SP - 40
Nama Method	updateRecipient
Logika Proses	<pre> ScheduleDatabaseHelper begin //menghapus recipient dengan messageId tertentu databaseHelper.deleteRecipient(messageId); //insert data recipient ke dalam database databaseHelper.saveScheduleToRecipients(recipient, messageId); end endprocedure </pre>

Tabel 82. Spesifikasi proses getRecipientFromRecipient

No.	SP - 41
Nama Method	getRecipientFromRecipient
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 02]
Deskripsi	Method ini digunakan untuk mendapatkan semua record recipient dengan id message tertentu
Return	cursor
Parameter	messageId : String
Method yang terkait	
Logika Proses	<pre> function getRecipientFromRecipient(String messageId) begin //mendapatkan semua record recipient yang memiliki messageId tertentu return SELECT all FROM Recipient WHERE message_id = messageId; end endfunction </pre>

Tabel 83. Spesifikasi proses getStringRecipients

No.	SP - 42
Nama Method	getStringRecipients
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 02]
Deskripsi	Method ini digunakan untuk mendapatkan semua record recipient dengan id message tertentu
Return	recipients : string
Parameter	recipient : cursor
Method yang terkait	moveToNext(), add(), get(), size(), toString()
Logika Proses	<pre> function getStringRecipients(Cursor recipient) var recipients : String, //penampung data recipients yang diambil dari database contact : Array of String, //penampung data contact sb : StringBuilder, //sebagai inisiasi dari class StringBuilder begin //add recipient ke dalam array while(recipient.moveToNext()) do contact.add(recipient.getString(recipient.getColumnIndex("recipient_number"))); endwhile //menyatukan setiap data dalam array menjadi satu string sb.append(contact.get(0)); for(int i <- 1 to i < contact.size()) do sb.append(","); sb.append(contact.get(i)); endfor recipients = sb.toString(); end </pre>

No.	SP - 42
Nama Method	getStringRecipients
	<pre> return recipients; end endprocedure</pre>

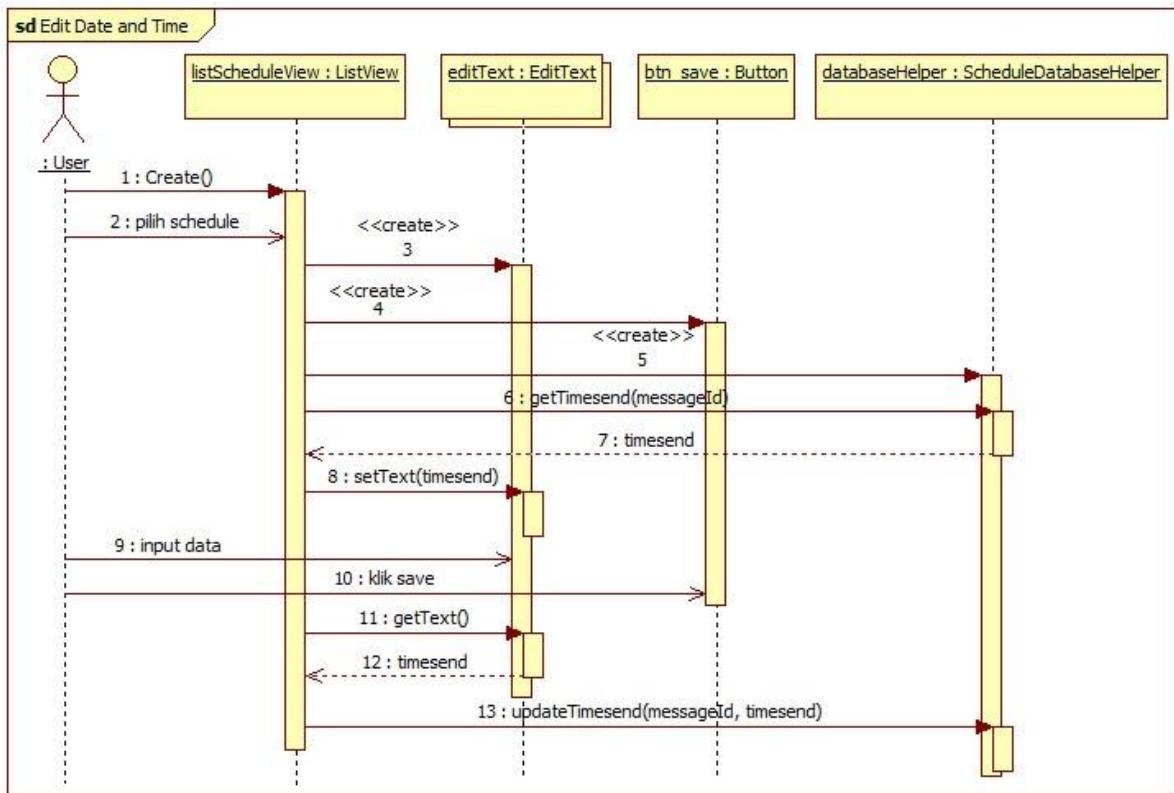
Tabel 84. Spesifikasi proses deleteRecipient

No.	SP - 43
Nama Method	deleteRecipient
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 11]
Deskripsi	Method ini digunakan untuk menghapus nomor recipient dengan id message tertentu dari database
Return	-
Parameter	messageId : String
Method yang terkait	
Logika Proses	<pre> procedure deleteRecipient(messageId : String) begin DELETE FROM Recipient WHERE message_id = messageId; end endprocedure</pre>

Tabel 85. Spesifikasi proses saveScheduleToRecipient

No.	SP - 44
Nama Method	saveScheduleToRecipient
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 11]
Deskripsi	Method ini digunakan untuk menyimpan schedule yang diubah ke tabel recipient di database
Return	-
Parameter	recipients, messageId : String
Method yang terkait	hasMoreElements(). nextElement()
Logika Proses	<pre> procedure saveScheduleToRecipient(recipients, messageId : String) var st : StringTokenizer //sebagai inisiasi terhadap class StringTokenizer tempPhoneNumber : String //sebagai penampung data nomor recipient begin StringTokenizer st <- new StringTokenizer(recipients, ";"); while (st.hasMoreElements()) do tempPhoneNumber <- st.nextElement(); INSERT INTO Recipient (message_id, recipient_number, status_recipient) VALUES (messageId, tempPhoneNumber, 'pending'); endwhile end endprocedure</pre>

4.2.2.1.8 Sequence Diagram Edit Date and Time Schedule



Gambar 24 . Sequence diagram edit date and time schedule

Keterangan dari *sequence diagram edit date and time schedule* pada gambar 24, dijelaskan pada tabel 86 - 93.

Tabel 86. Keterangan *sequence diagram edit date and time schedule*

No.	SD - 10
Requirement	[REQ F - 02], [REQ UPD - 05], [REQ UPD - 06], [REQ UPD - 07], [REQ UPD - 10]
Deskripsi	Sequence ini menggambarkan proses untuk melakukan perubahan waktu pengiriman dari <i>schedule</i> yang sebelumnya sudah dibuat (<i>edit date and time</i>)
Class yang terkait	ListView, ScheduleDatabaseHelper, EditText(), Button()
Method yang terkait	getTimeSend(), setText(), getText(), updateTimeSend()
Logika Proses	<pre> Var listScheduleView : ListView, //sebagai inisiasi dari class ListView editText : EditText, //sebagai inisiasi dari class EditText btn_save : Button, //sebagai inisiasi dari class Button, yang memungkinkan untuk trigger proses selanjutnya databaseHelper : ScheduleDatabaseHelper, //sebagai inisiasi terhadap class ScheduleDatabaseHelper timesend : String, //sebagai penampung data timesend messageId : String, //sebagai penampung schedule id yang dipilih user begin //user memilih schedule yang akan diedit if(user.pilihschedule) then messageId <- listScheduleView.OnItemClick(user.pilihschedule); timesend <- databasehelper.getTimeSend(messageId); editText.setText(timesend); </pre>

No.	SD - 10
	<pre>//user memasukkan data timesend baru if(user.inputdata) then //user klik button save if(btn_save.clicked) then timesend <- editText.getText(); databaseHelper.updateTimesend(messageId, timesend); endif endif endif endif</pre>

Tabel 87. Spesifikasi proses updateTime

No.	SP - 45
Nama Method	updateTime
Nama Class	EditTime
Requirement	[REQ UPD - 01], [REQ UPD - 11]
Deskripsi	Method ini digunakan untuk meng-update time dengan waktu <i>schedule</i> dengan id message tertentu
Return	-
Parameter	messageId : String, timesend : Long
Method yang terkait	
Logika Proses	<pre>procedure updateTime(messageId : String, timesend, timemilis : long) var messageId : String, //sebagai penampung messageId messageId : String, //sebagai penampung messageId timesend : long, //sebagai penampung waktu dalam bentuk millisecond cursor : Cursor, //sebagai penampung hasil fetch dari database databaseHelper : ScheduleDatabaseHelper, //sebagai inisisasi terhadap class ScheduleDatabaseHelper countData : int, //sebagai penampung banyaknya record suatu data yang di fetch dari database frequency : String, //sebagai penampung data frequency yang didapat dari database intent : new Intent(EditTime.this, SMSActivity.class), //sebagai inisisasi terhadap class Intent pending : PendingIntent.getActivity(EditTime.this, (int), intent, PendingIntent.FLAG_UPDATE_CURRENT), //sebagai inisisasi class PendingIntent dataRepetition : Array of String, //sebagai penampung data time : Array of long, //sebagai penampung time freqtime : Frequency, //sebagai insiasi terhadap class Frequency alarm : AlarmManager, //sebagai inisisasi terhadap class AlarmManager begin //mendapatkan record time dari database dengan messageId tertentu cursor <- databaseHelper.getTime(String messageId); //mendapatkan banyaknya record time dari database countData <- databaseHelper.count(cursor); //melakukan pengapusan record time dengan messageId tertentu databaseHelper.deleteTime(messageId); //mendapatkan frequency dari database terhadap table message dengan messageId tertentu frequency <- databaseHelper.getMessageFrequency(messageId); //melakukan save atau penyimpanan data ke database terhadap table time database.saveScheduleToTime(timesend, messageId); //bila countData > 1 berarti frequency schedule message tersebut bukanlah "Once" if(countData > 1) then //menampung data dataRepetition[0] <- frequency; dataRepetition[1] <- messageId; dataRepetition[2] <- countData; //menampung waktu time[0] <- timesend; //memanggil method terhadap class Frequency freqtimes.repetition(pending, alarm, databaseHelper, dataRepetition, time);</pre>

No.	SP - 45
Nama Method	updateTime
	<pre> else //melakukan set terhadap class AlarmManager yang berfungsi untuk interrupt alarm.set(); endif end endprocedure </pre>

Tabel 88. Spesifikasi proses getTime

No.	SP - 46
Nama Method	getTime
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 02]
Deskripsi	Method ini digunakan untuk mendapatkan semua <i>time</i> dengan <i>id message</i> tertentu
Return	cursor : Cursor
Parameter	messageId : String
Method yang terkait	
Logika Proses	<pre> function getTime(messageId : String) var cursor : Cursor //sebagai penampung dari data yang berhasil diambil dari database begin cursor <- SELECT all FROM Delivery_Time WHERE message_id = messageId; return cursor; end endfunction </pre>

Tabel 89. Spesifikasi proses count

No.	SP - 47
Nama Method	count
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ UPD - 01]
Deskripsi	Method ini digunakan untuk menghitung banyaknya <i>record</i> dari <i>time</i> untuk <i>id message</i> tertentu
Return	count : Integer
Parameter	cursor : Cursor
Method yang terkait	getCount()
Logika Proses	<pre> function count(Cursor cursor) var count : integer //sebagai penampung berapa banyak row record yang ada pada cursor begin //mendapatkan jumlah dari banyaknya record count <- cursor.getCount(); return count; end endfunction </pre>

Tabel 90. Spesifikasi proses deleteTime

No.	SP - 48
Nama Method	deleteTime
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 11]
Deskripsi	Method ini digunakan untuk menghapus <i>time</i> dengan <i>id message</i> tertentu
Return	-

No.	SP - 48
Nama Method	deleteTime
Parameter	messageId : String
Method yang terkait	
Logika Proses	<pre>procedure deleteTime(messageId : String) begin DELETE FROM Delivery_Time WHERE message_id = messageId; end endprocedure</pre>

Tabel 91. Spesifikasi proses getMessageFrequency

No.	SP - 49
Nama Method	getMessageFrequency
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 02]
Deskripsi	Method ini digunakan untuk mengambil frekuensi pengiriman dengan <i>id message</i> tertentu
Return	frequency : String
Parameter	messageId : String
Method yang terkait	
Logika Proses	<pre>function getMessageFrequency(messageId : String) var frequency : String //sebagai penampung data frequency hasil fetch dari database begin frequency <- SELECT frequency_message FROM Message WHERE message_id = messageId; return frequency; end endfunction</pre>

Tabel 92. Spesifikasi proses saveScheduleToTime

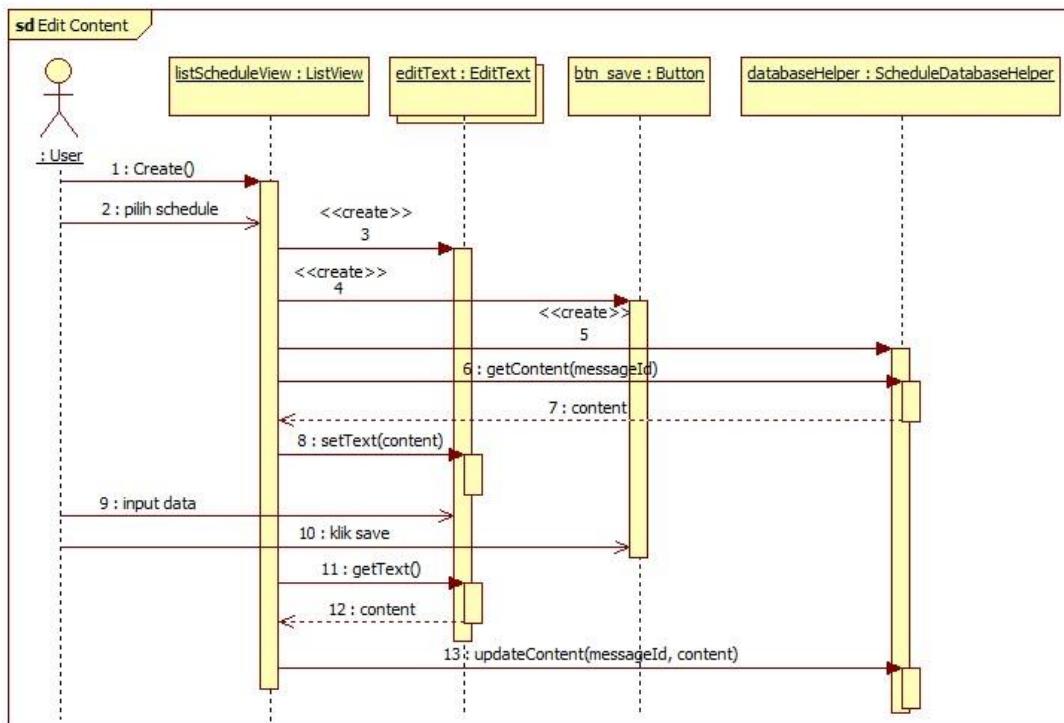
No.	SP - 50
Nama Method	saveScheduleToTime
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 11]
Deskripsi	Method ini digunakan untuk menyimpan waktu <i>schedule</i> ke tabel time di <i>database</i>
Return	-
Parameter	timesend : long, messageId : String
Method yang terkait	
Logika Proses	<pre>procedure saveScheduleToTime(timesend : long, messageId : String) begin INSERT INTO Delivery_Time (message_id, status_delivery_time, time_message_sent, alert_time_message) VALUES (messageId, 'scheduled', timesend, alert); end endprocedure</pre>

Tabel 93. Spesifikasi proses repetition

No.	SP - 51
Nama Method	repetition
Nama Class	MessageInterval
Requirement	[REQ UPD - 01], [REQ UPD - 09]
Deskripsi	Method ini digunakan untuk menangani pengulangan pengiriman <i>schedule</i>
Return	-

No.	SP - 51
Nama Method	repetition
Parameter	pending : PendingActivity, alarm : AlarmManager, databaseHelper : ScheduleDatabaseHelper, data : Array of String, time : Array of Long
Method yang terkait	setRepeating(), parseInt(), add(), saveScheduleToTime()
Logika Proses	
<pre> procedure repetition(pending : PendingActivity, alarm : AlarmManager, databaseHelper : ScheduleDatabaseHelper, data : Array of String, time : Array of Long) var timesend : long //sebagai penampung data timesend frequency : String //sebagai penampung data frequency messageId : String //sebagai penampung data messageId interval : long //sebagai penampung data interval begin timesend <- time[0]; frequency <- data[0]; messageId <- data[1]; service <- AlarmManager.getSystemServiceSystem(); //memanggil method interval untuk menghitung interval waktu sesuai dengan frequencynya interval <- interval(frequency); //set ke dalam AlarmManager sebagai interrupt service.setRepeating(typeServiceAlarm, timesend, interval, pending); //insert ke dalam database sesuai dengan remaining yang diinputkan for(int remaining <- Integer.parseInt(data[2]) to remaining > 1) do //melakukan insert ke database ke table time databaseHelper.saveScheduleToTime(timesend, messageId); endfor end endprocedure </pre>	

4.2.2.1.9 Sequence Diagram Edit Content Schedule



Gambar 25 . Sequence diagram edit content schedule

Keterangan dari *sequence diagram edit content schedule* pada gambar 25, dijelaskan pada tabel 94 - 97

Tabel 94. Keterangan *sequence diagram edit content schedule*

No.	SD - 11
Requirement	[REQ F - 02], [REQ UPD - 08], [REQ UPD - 10], [REQ UPD - 11]
Deskripsi	Sequence ini menggambarkan proses untuk melakukan perubahan waktu pengiriman dari <i>schedule</i> yang sebelumnya sudah dibuat (<i>edit date and time</i>)
Class yang terkait	ListView, ScheduleDatabaseHelper, EditText, Button
Method yang terkait	getContent(), setText(), getText(), updateContent()
Logika Proses	<pre> listScheduleView : ListView, //sebagai inisiasi dari class ListView editText : EditText, //sebagai inisiasi dari class EditText btn_save : Button, //sebagai inisiasi dari class Button, yang memungkinkan untuk trigger proses selanjutnya databaseHelper : ScheduleDatabaseHelper, //sebagai inisiasi terhadap class ScheduleDatabaseHelper content : String, //sebagai penampung data content messageId : String, //sebagai penampung schedule id yang dipilih user begin //user memilih schedule yang akan diedit if(user.pilihschedule) then messageId <- listScheduleView.OnItemClick(user.pilihschedule); content <- databasehelper.getMessage(messageId); editText.setText(content); //user memasukkan data content baru if(user.inputdata) then //user klik button save if(btn_save.clicked) then content <- editText.getText(); databaseHelper.updateRecipient(messageId, content); endif endif endif end </pre>

Tabel 95. Spesifikasi proses getMessage

No.	SP - 52
Nama Method	getMessage
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 02]
Deskripsi	Method ini digunakan untuk mendapatkan isi pesan dengan <i>id message</i> tertentu
Return	message : String
Parameter	messageId : String
Method yang terkait	
Logika Proses	<pre> function getMessage(String messageId) var message : String, //sebagai penampung data message hasil fetch dari database begin message <- SELECT content_message FROM Message where message_id = messageId; return message; end endfunction </pre>

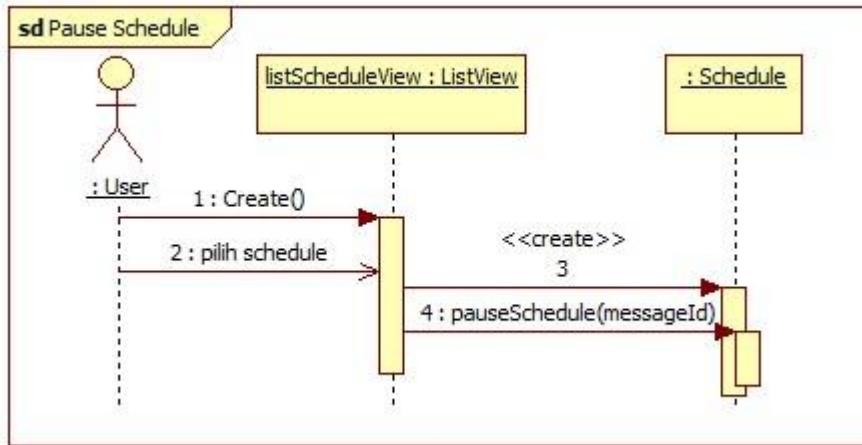
Tabel 96. Spesifikasi proses updateType

No.	SP - 53
Nama Method	updateType
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 12]
Deskripsi	Method ini digunakan untuk meng-update tipe message dengan tipe pesan sesuai dengan content pesan yang diubah oleh user
Return	-
Parameter	type, content, messageId : String
Method yang terkait	
Logika Proses	<pre> procedure updateType(type, content, messageId : String) begin if(type = "normal") then UPDATE Normal_Message SET content_normal_message = content WHERE message_id = messageId; else UPDATE Typical_Message SET content_typical_message = content WHERE message_id = messageId; endif end endprocedure </pre>

Tabel 97. Spesifikasi proses updateContent

No.	SP - 54
Nama Method	updateContent
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 12]
Deskripsi	Method ini digunakan untuk meng-update content pesan dengan content pesan yang diubah oleh user
Return	-
Parameter	messageId, Content : String
Method yang terkait	
Logika Proses	<pre> procedure updateContent(Content, messageId : String) var messageId : String, //sebagai penampung messageId content : String, //sebagai penampung pesan atau content yang didapatkan dari database databaseHelper : ScheduleDatabaseHelper, //sebagai inisisasi terhadap class ScheduleDatabaseHelper type : String, //sebagai penampung tipe dari pesan begin //mendapatkan tipe dari pesan type <- schedule.getType(); //melakukan update terhadap tipe dari content dengan messageId tertentu databaseHelper.updateType(type, content, messageId); //melakukan update terhadap content dengan messageId tertentu databaseHelper.updateContent(messageId, content); end endprocedure </pre>

4.2.2.1.10 Sequence Diagram Pause Schedule



Gambar 26 . Sequence diagram pause schedule

Keterangan dari *sequence diagram pause schedule* pada gambar 26, dijelaskan pada tabel 98 - 100.

Tabel 98. Keterangan *sequence diagram pause schedule*

No.	SD - 12
Requirement	[REQ F - 04], [REQ PSD - 01]
Deskripsi	Sequence ini menggambarkan proses untuk menghentikan sementara pengiriman suatu schedule yang sebelumnya telah dibuat (pause schedule)
Class yang terkait	Menu, ListScheduleView, ArrayList, AlarmManager, ScheduleDatabaseHelper
Method yang terkait	get(), pauseSchedule(), updateStatus()
Logika Proses	<pre> Var listScheduleView : ListView, //sebagai inisiasi dari class ListView messageId : String, //sebagai penampung schedule id yang dipilih user schedule : Schedule, //inisiasi dari class Schedule begin if(user.pilihSchedule) then messageId <- listScheduleView.OnItemClick(user.pilihSchedule); schedule.pauseSchedule(messageId); endif end </pre>

Tabel 99. Spesifikasi proses pauseSchedule

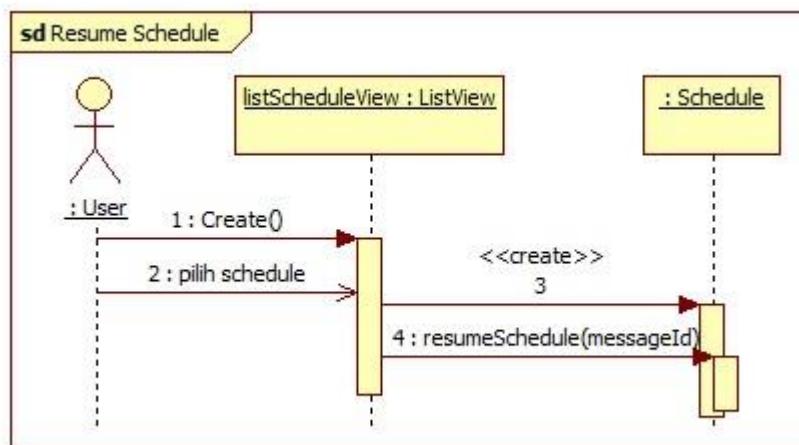
No.	SP - 55
Nama Method	pauseSchedule
Nama Class	AlarmManager
Requirement	[REQ PSD - 01]
Deskripsi	Method ini digunakan untuk menghapus schedule dengan id message tertentu dari alarm manager
Return	Boolean
Parameter	messageId : String
Method yang terkait	getTime(), moveToNextRecord(), getLong(), getColumnIndex()

No.	SP - 55
Nama Method	pauseSchedule
Logika Proses	
<pre>Procedure pauseSchedule(messageId : String) alarm : AlarmManager, //sebagai inisiasi terhadap class AlarmManager databaseHelper : ScheduleDatabaseHelper, //sebagai inisiasi terhadap class ScheduleDatabaseHelper cursor : Cursor, //penampung data hasil fetch dari database timesend : long, //penampung waktu pengiriman begin cursor <- databaseHelper.getTime(messageId); while(cursor.moveToNext()) do timesend <- cursor.getNextData(cursor.getColumnIndex("time_message_send")); alarm.cancel(timesend, messageId); endwhile end endprocedure</pre>	

Tabel 100. Spesifikasi proses getTime

No.	SP - 56
Nama Method	getTime
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ F - 03], [REQ DEL - 01], [REQ DEL - 02], [REQ DEL - 03]
Deskripsi	Method ini digunakan untuk mengambil semua waktu pengiriman dengan <i>message id</i> tertentu
Return	Cursor : cursor
Parameter	MessageId : string
Method yang terkait	
Logika Proses	
<pre>function getTime(messageId : String) var cursor : Cursor, //sebagai penampung dari data yang berhasil diambil dari database begin //mendapatkan waktu pengiriman dengan messageId tertentu cursor <- SELECT all FROM Delivery_Time WHERE message_id = messageId; return cursor; end endfunction</pre>	

4.2.2.1.11 Sequence Diagram Resume Schedule



Gambar 27 . Sequence diagram resume schedule

Keterangan dari *sequence diagram resume schedule* pada gambar 27, dijelaskan pada tabel 101 - 103.

Tabel 101. Keterangan *sequence diagram resume schedule*

No.	SD - 13
Requirement	[REQ F - 04], [REQ PSD - 02], [REQ PSD - 03], [REQ PSD - 04]
Deskripsi	Sequence ini menggambarkan proses untuk melanjutkan kembali pengiriman <i>schedule</i> yang sebelumnya dipause (<i>resume schedule</i>)
Class yang terkait	Menu, ListScheduleView, ArrayList, ScheduleDatabaseHelper, Activity, PendingActivity, AlarmManager
Method yang terkait	get(), getTime(), set()
Logika Proses	<pre> Var listScheduleView : ListView, //sebagai inisiasi dari class ListView schedule : Schedule, //sebagai inisiasi dari class Schedule messageId : String, //sebagai penampung schedule id yang dipilih user begin //user memilih schedule yang akan di resume if(user.pilihschedule) then messageId <- listScheduleView.OnItemClick(user.pilihschedule); schedule.resumeSchedule(messageId); endif end </pre>

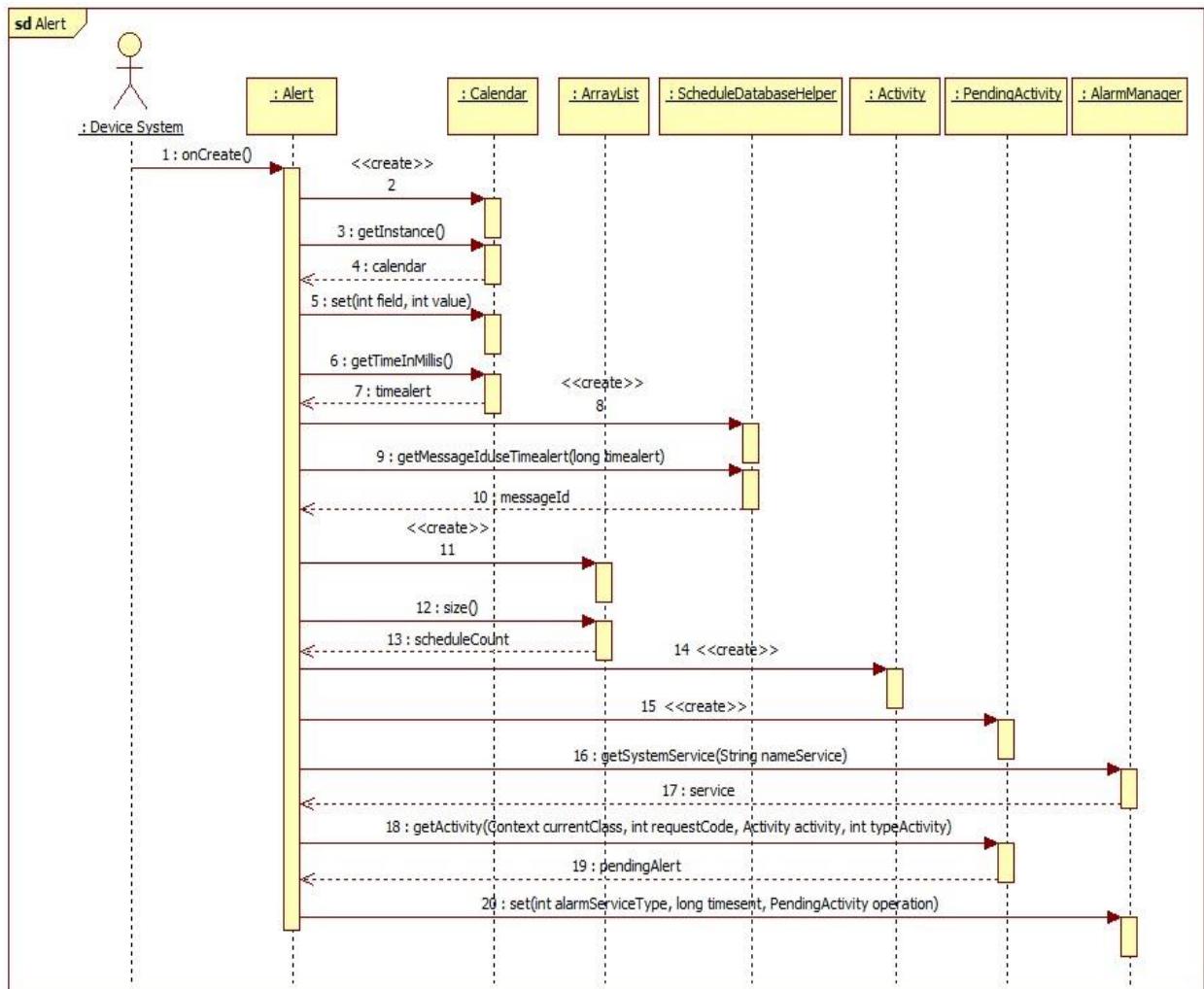
Tabel 102. Spesifikasi proses *resumeSchedule*

No.	SP - 57
Nama Method	resumeSchedule
Nama Class	Schedule
Requirement	[REQ PSD - 01]
Deskripsi	Method ini digunakan untuk melakukan <i>resume</i> pada <i>schedule</i> dengan <i>id</i> tertentu
Return	-
Parameter	messageId : String
Method yang terkait	getTime(), getSystemService(),moveToNextRecord(),
Logika Proses	<pre> Procedure resumeSchedule(messageId : String) var messageId : String, //sebagai penampung schedule id yang dipilih user timesend : Cursor, //sebagai inisiasi dari class Cursor databaseHelper : ScheduleDatabaseHelper, //sebagai inisiasi terhadap class ScheduleDatabaseHelper alarm : AlarmManager, //inisiasi terhadap class AlarmManager begin //mendapatkan waktu-waktu pengiriman yang memiliki messageId tertentu timesend <- databaseHelper.getTime(messageId); //bila waktu pengiriman belum melebihi dari waktu sistem if(timesend < system.timenow) then //mendapatkan layanan alarm <- AlarmManager.getSystemService(); while(timesend.moveToNext()) do alarm.set(); endwhile endif end endprocedure </pre>

Tabel 103. Spesifikasi proses getTime

No.	SP - 58
Nama Method	getTime
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ PSD - 01]
Deskripsi	Method ini digunakan untuk mendapatkan waktu <i>schedule</i> dengan <i>id</i> tertentu
Return	cursor : Cursor
Parameter	messageId : String
Method yang terkait	
Logika Proses	<pre>function getTime(messageId : String) var cursor : Cursor, //sebagai penampung dari data yang berhasil diambil dari database begin cursor <- SELECT all FROM Delivery_Time WHERE message_id = messageId; return cursor; end endprocedure</pre>

4.2.2.1.12 Sequence Diagram Menampilkan Alert



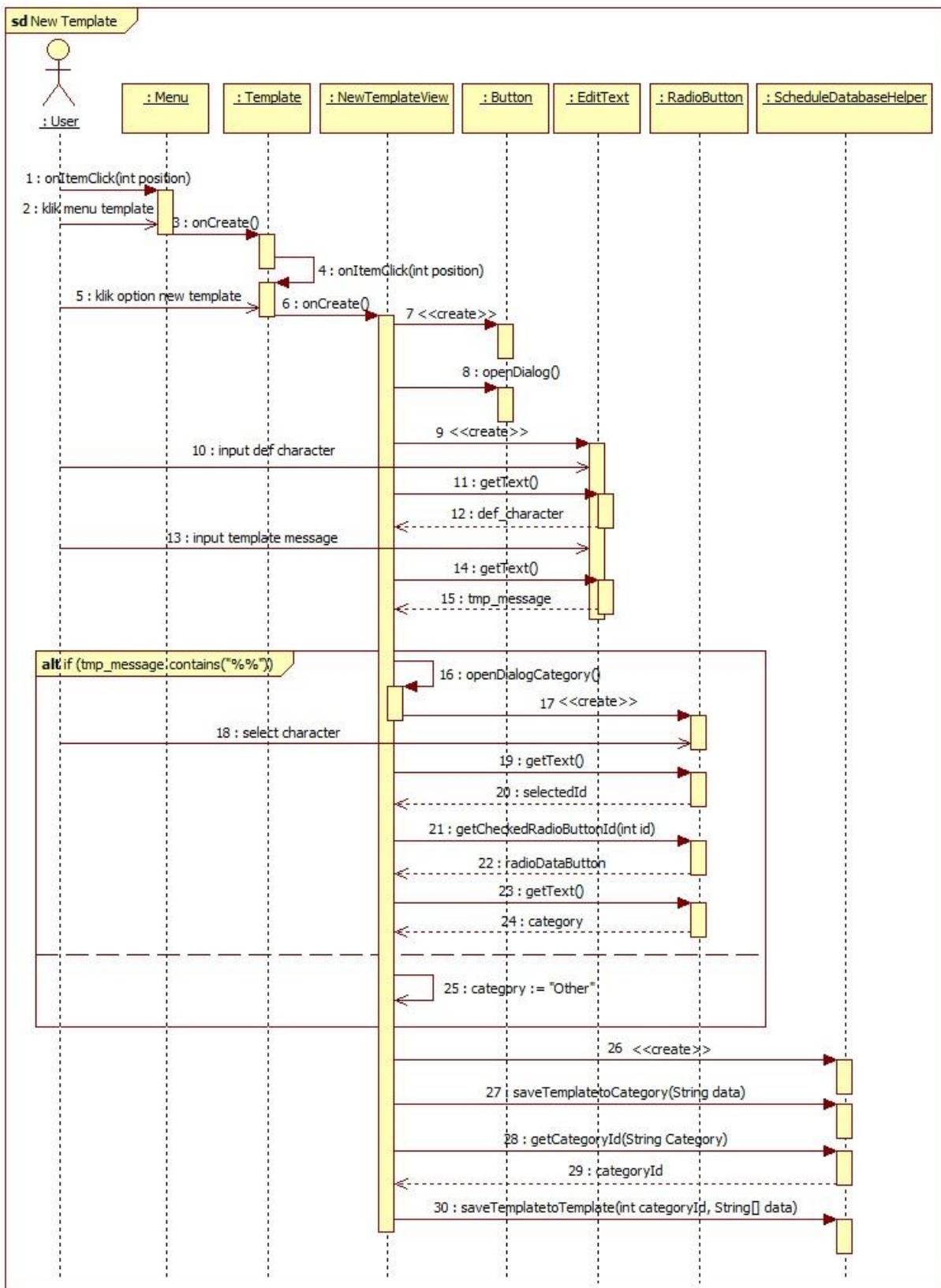
Gambar 28 . Sequence diagram menampilkan alert

Keterangan dari *sequence diagram* menampilkan *alert* pada gambar 28, dijelaskan pada tabel 104.

Tabel 104. Keterangan *sequence diagram* menampilkan *alert*

No.	SD - 14
Requirement	[REQ F - 07], [REQ ALR - 01], [REQ ALR - 02], [REQ ALR - 03], [REQ ALR - 04], [REQ ALR - 05]
Class yang terkait	Alert, Calendar, ArrayList, ScheduleDatabaseHelper, Activity, PendingActivity, AlarmManager
Method yang terkait	getInstance(), getTimeInMillis(), getMessageIduseTimeAlert(), size(), getSystemService(), getActivity, set()
Deskripsi	<i>Sequence</i> ini digunakan untuk menjalankan fungsi alert sesuai dengan pengaturan yang telah disimpan pada database.
Logika Proses	<pre> Var calendar : Calendar //sebagai inisiasi dari class Calendar timealert : Long //sebagai penampung waktu untuk menampilkan alert messageId : String //sebagai penampung id message scheduleCount : Integer //sebagai penampung banyaknya schedule yang akan dilaksanakan pendingAlert : PendingActivity //sebagai inisiasi dari class PendingActivity alertActivation : Boolean //sebagai penampung status aktivasi alert alertTone : Uri //sebagai penampung nada dering alert begin cal <- new Calendar(); calendar <- cal.getInstance(); cal.set(field, value); timealert <- cal.getTimeInMillis(); database <- new ScheduleDatabaseHelper(); messageId <- database.getMessageIduseTimealer(timealert); array_list <- new ArrayList(); scheduleCount <- array_list.size(); activityAlert <- new Activity(); activity.putExtra("ScheduleCount", scheduleCount); p_activity <- new PendingActivity(); pendingAlert <- p_activity.getActivity(); service <- AlarmManager.getSystemService(); service.set(AlarmManager.typeAlarmServices, timealert, pendingAlert); end </pre>

4.2.2.1.13 Sequence Diagram Membuat Template



Gambar 29 . Sequence diagram membuat template

Keterangan dari *sequence diagram* membuat *template* pada gambar 29, dijelaskan pada tabel 105 - 108.

Tabel 105. Keterangan *sequence diagram* membuat *template*

No.	SD - 15
Requirement	[REQ TMP - 09], [REQ TMP - 01], [REQ TMP - 02], [REQ TMP - 03], [REQ TMP - 04], [REQ TMP - 05]
Deskripsi	Sequence ini menggambarkan proses untuk membuat <i>template</i> baru.
Class yang terkait	Menu, Template, NewTemplate, Button, EditText, RadioButton, TemplateDatabaseHelper
Method yang terkait	registerForContextMenu(), getText(), openDialogCategory(), getCheckedRadioButtonId(), saveTemplatetoCategory(), getCategoryID(), saveTemplatetoTemplate()
Logika Proses	<pre> databaseHelper : ScheduleDatabaseHelper //sebagai penampung inisiasi class ScheduleDatabaseHelper data : Array of String //sebagai penampung data-data template yang akan disimpan data[1] : message of template //sebagai penampung isi template data[2] : type of template from type of category //sebagai penampung tipe template data[3] : name of template //sebagai penampung nama template name, message : String categoryID : integer //sebagai penampung id kategori template tmp_message, tmp_name : EditText //sebagai penampung inisiasi class EditText data_count : button // sebagai penampung inisiasi class Button def_ch : Array of String {"%AGE%", "%DATE%", "%MONTH%", "%YEAR%"} //sebagai penampung string khusus yang dapat disisipkan pada template begin tmp_message <- new EditText(); tmp_name <- new EditText(); data_count <- new button(); tmp_message <- getText(); tmp_name <- getText(); //after fields(all editText widget) fill up if (tmp_message.contains("%")) then openDialogCategory(); //select between Birthday and Anniversary else category <- "Other"; endif //menyimpan template ke tabel kategori di database saveTemplatetoCategory(category); //mengambil id kategori template categoryID <- getCategoryID(category); //menyimpan template ke tabel template di database saveTemplatetoTemplate(categoryID, data[]); end </pre>

Tabel 106. Spesifikasi proses saveTemplatetoCategory

No.	SP - 59
Nama Method	saveTemplatetoCategory
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ TMP - 04]
Deskripsi	Method ini digunakan untuk menyimpan kategori <i>template</i> ke tabel category_template database
Return	-
Parameter	data: String
Method yang terkait	

No.	SP - 59
Logika Proses	
<pre>procedure saveTemplatetoCategory(data : String) begin INSERT INTO Category_template (type) VALUES (data); end endprocedure</pre>	

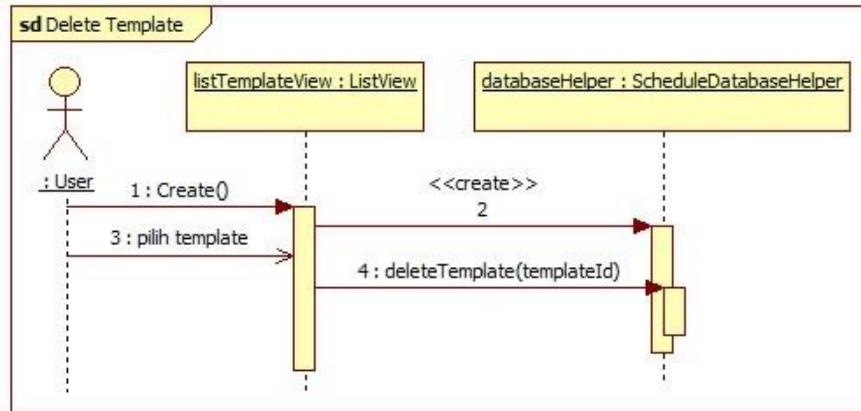
Tabel 107. Spesifikasi proses getCategoryId

No.	SP - 60
Nama Method	getCategoryId
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ TMP - 04]
Deskripsi	Method ini digunakan untuk mengambil <i>id</i> kategori <i>template</i>
Return	categoryId : Integer
Parameter	category : string
Method yang terkait	moveToNextRecord(), getInteger(), getColumnIndex()
Logika Proses	
<pre>function getCategoryId(category : String) cursor : Cursor categoryId : Integer begin cursor <- SELECT * FROM Category_template WHERE type = 'category'; if(cursor != null)then while(cursor.moveToFirst()) do categoryId = cursor.getInteger(cursor.getColumnIndex("category_id")); endwhile endif return categoryId; end endfunction</pre>	

Tabel 108. Spesifikasi proses saveTemplatetoTemplate

No.	SP - 61
Nama Method	saveTemplatetoTemplate
Nama Class	TemplateDatabaseHelper
Requirement	[REQ TMP - 04]
Deskripsi	Method ini digunakan untuk <i>template</i> yang dibuat ke tabel <i>Template</i> di <i>database</i>
Return	-
Parameter	categoryId : Integer, data : Array of String
Method yang terkait	getWritableDatabase(), put(), insert()
Logika Proses	
<pre>procedure saveTemplatetoTemplate(categoryID : Integer, data : Array of String) begin INSERT INTO Template (content_template, template_type, template_name) VALUES (data[1], data[2], data[3]); end endprocedure</pre>	

4.2.2.1.14 Sequence Diagram Menghapus Template



Gambar 30 . Sequence diagram menghapus template

Keterangan dari sequence diagram menghapus template pada gambar 30, dijelaskan pada tabel 109 - 110.

Tabel 109. Keterangan sequence diagram menghapus template

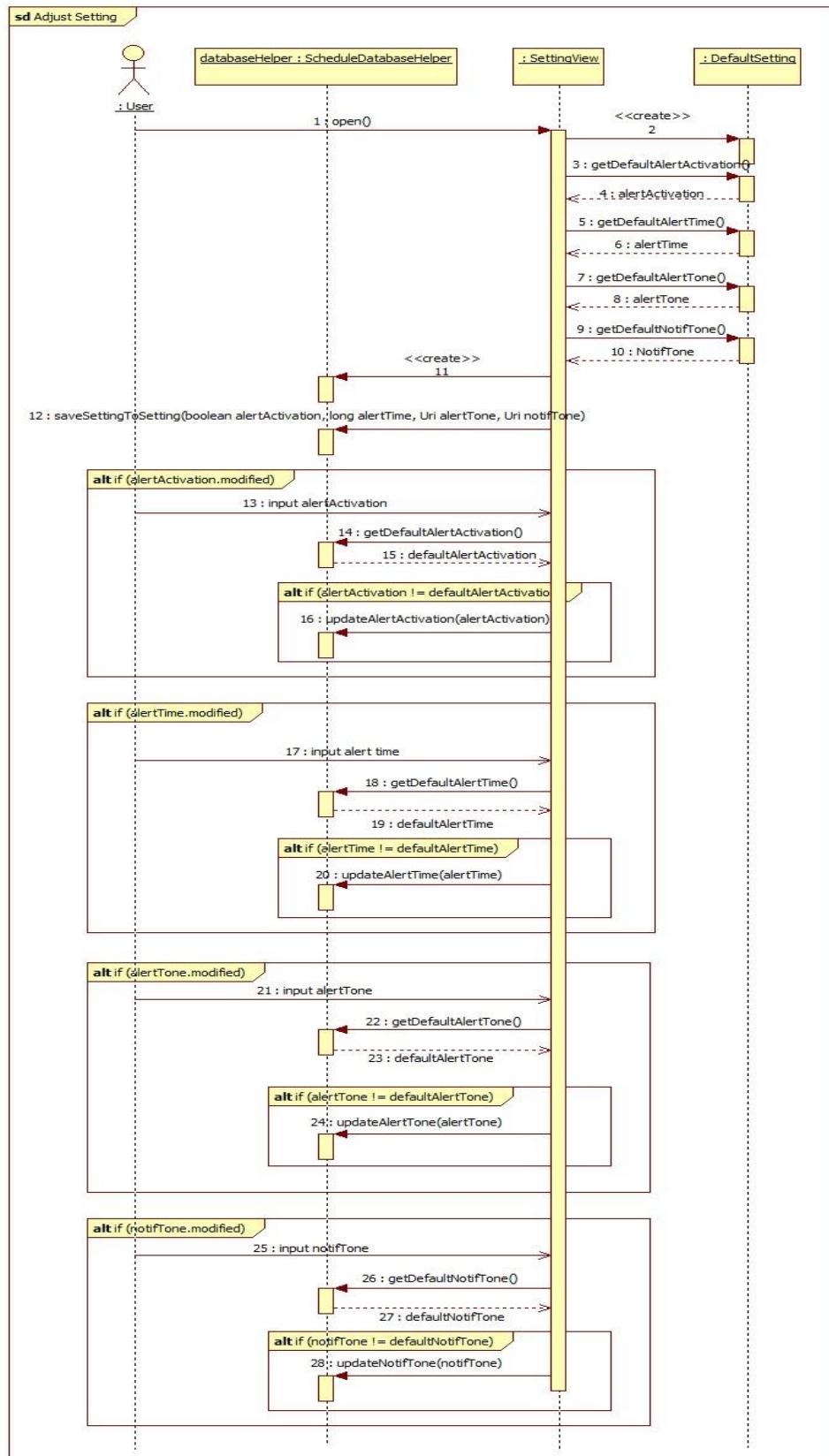
No.	SD - 16
Requirement	[REQ TMP - 09], [REQ TMP - 07]
Deskripsi	Sequence ini menggambarkan proses untuk menghapus template yang sebelumnya telah dibuat
Class yang terkait	ListView, ScheduleDatabaseHelper
Method yang terkait	deleteTemplate()
Logika Proses	<pre> listTemplateView : ListView, //sebagai inisiasi terhadap class ListView databaseHelper : ScheduleDatabaseHelper, //sebagai inisiasi terhadap class ScheduleDatabaseHelper templateId : String, //sebagai penampung template id yang dipilih user begin //user memilih tempa yang akan di hapus if(user.pilihtemplate) then templateId <- listTemplateView.OnItemClick(user.pilihtemplate); databaseHelper.deleteTemplate(templateId); endif end </pre>

Tabel 110. Spesifikasi proses deleteTemplate

No.	SP - 62
Nama Method	deleteTemplate
Nama Class	ScheduleDatabaseHelper
Requirement	[REQ TMP - 07]
Deskripsi	Method ini digunakan untuk menghapus template dengan id template tertentu dari database
Return	-
Parameter	templateId : String
Method yang terkait	

No.	SP - 62
Nama Method	deleteTemplate
Logika Proses	
<pre>Procedure deleteTemplate(templateId : String) var begin DELETE FROM Template WHERE template_id = templateId; end endfunction</pre>	

4.2.2.1.15 Sequence Diagram Setting



Gambar 31 . Sequence diagram setting

Keterangan dari *sequence diagram setting* pada gambar 31, dijelaskan pada tabel 111 – 124.

Tabel 111. Keterangan *sequence diagram setting*

No.	SD - 17
Requirement	[REQ F - 10], [REQ F - 10.1], [REQ F - 10.2], [REQ SA - 01], [REQ SA - 02], [REQ SA - 03], [REQ SA - 04], [REQ SA - 05], [REQ SN - 01], [REQ SN - 02], [REQ SN - 03]
Deskripsi	<i>Sequence</i> ini digunakan untuk menjalankan fungsi <i>setting</i> . Ketika aplikasi diinstall, <i>setting</i> aplikasi sudah diatur dengan <i>default setting</i> . Namun, jika <i>user</i> menginput perubahan <i>setting</i> , maka aplikasi akan menyimpan dan menjalankan <i>setting</i> sesuai dengan yang telah diatur oleh <i>user</i> .
Class yang terkait	DefaultSetting, ScheduleDatabaseHelper, SettingView
Method yang terkait	getDefaultUri(), saveToSetting(), getAlertActivation(), updateAlertActivation(), getAlertTime(), updateAlertTime(), getRingtoneUri(), getAlertTone, updateAlertTone, getNotifTone, updateNotifTone
Logika Proses	<pre> Var alertActivation, defaultAlertActivation : Boolean //sebagai penampung status aktivasi alert alertTime, defaultAlertTime : Long //sebagai penampung waktu untuk menampilkan alert alertTone, defaultAlertTone, notifTone, defaultNotifTone : Uri //sebagai penampung nada dering alert default_setting : Default Setting //sebagai penampung inisiasi class DefaultSetting helper : ScheduleDatabaseHelper //sebagai penampung inisiasi class ScheduleDatabaseHelper begin default_setting <- new DefaultSetting(); //mengambil nilai untuk default setting dari class DefaultSetting alertActivation <- default_setting.getDefaultAlert(); alertTime <- default_setting.getDefaultAlertTime(); alertTone <- default_setting.getDefaultAlertTone(); notifTone <- default_setting.getDefaultNotifTone(); //menyimpan default setting ke tabel setting di database helper <- new ScheduleDatabaseHelper(); helper.saveSettingToSetting(alertActivation, alertTime, alertTone, notifTone); //mengambil aktivasi alert yang diinputkan user alertActivation <- getText(); //mengambil aktivasi alert yang sebelumnya sudah tersimpan di database defaultAlertActivation <- helper.getDefaultAlertActivation(); //membandingkan aktivasi alert if(defaultAlertActivation != alertActivation) then helper.updateAlertActivation(alertActivation, defaultAlertActivation); end //mengambil waktu alert yang diinputkan user alertTime <- getText(); //mengambil waktu alert yang sebelumnya sudah tersimpan di database defaultAlertTime <- helper.getDefaultAlertTime(); //membandingkan waktu alert if(defaultAlertTime != alertTime) then helper.updateAlertTime(alertTime, defaultAlertTime); end //mengambil nada dering alert yang diinputkan user alertTone <- getRingtoneUri(position); //mengambil nada dering alert yang sebelumnya sudah tersimpan di database defaultAlertTone <- helper.getDefaultAlertTone(); //membandingkan nada dering alert if(defaultAlertTone != alertTone) then helper.updateAlertTone(alertTone, defaultAlertTone); end //mengambil nada dering notification yang diinputkan user notifTone <- getRingtone(); //mengambil nada dering notification yang sebelumnya sudah tersimpan di database defaultNotifTone <- helper.getDefaultNotifTone(); //membandingkan nada dering notification </pre>

No.	SD - 17
	<pre>if(defaultNotifTone != notifTone) then helper.updateNotifTone(notifTone, defaultNotifTone); end end</pre>

Tabel 112. Spesifikasi proses getDefaultAlertActivation

No.	SP - 63
Nama Method	getDefaultAlertActivation
Nama Class	DefaultSetting
Requirement	Method ini digunakan untuk mengambil <i>default setting</i> untuk aktivasi <i>alert</i> dari class DefaultSetting
Deskripsi	[REQ SA - 01]
Return	alertActivation : Boolean
Method yang terkait	-
Logika Proses	<pre>function getDefaultAlertActivation() begin return alertActivation; end endfunction</pre>

Tabel 113. Spesifikasi proses getDefaultAlertTime

No.	SP - 64
Nama Method	getDefaultAlertTime
Nama Class	DefaultSetting
Requirement	Method ini digunakan untuk mengambil <i>default setting</i> untuk waktu ketika menampilkan <i>alert</i> dari class DefaultSetting
Deskripsi	[REQ SA - 02]
Return	alertTime : long
Method yang terkait	-
Logika Proses	<pre>function getDefaultAlertTime() begin return alertTime; end endfunction</pre>

Tabel 114. Spesifikasi proses getDefaultAlertTone

No.	SP - 65
Nama Method	getDefaultAlertTone
Nama Class	DefaultSetting
Requirement	Method ini digunakan untuk mengambil <i>default setting</i> untuk nada dering <i>alert</i> dari class DefaultSetting
Deskripsi	[REQ SA - 03]
Return	alertTone : Uri
Parameter	-
Method yang terkait	-
Logika Proses	<pre>function getDefaultAlertTone() begin return alertTone; end endfunction</pre>

Tabel 115. Spesifikasi proses getDefaultNotifTone

No.	SP - 66
Nama Method	getDefaultNotifTone
Nama Class	DefaultSetting
Requirement	Method ini digunakan untuk mengambil <i>default setting</i> untuk aktivasi <i>notification</i> dari class DefaultSetting
Deskripsi	[REQ SN - 01]
Return	notifTone : Uri
Parameter	-
Method yang terkait	-
Logika Proses	<pre>function getDefaultNotifTone() begin return notifTone; end endfunction</pre>

Tabel 116. Spesifikasi proses saveSettingToSetting

No.	SP - 67
Nama Method	saveSettingToSetting
Nama Class	ScheduleDatabaseHelper
Requirement	Method ini digunakan untuk menyimpan <i>default setting</i> yang didapatkan dari class DefaultSetting. Tabel yang terkait dengan method ini adalah tabel Setting.
Deskripsi	[REQ SA - 05]
Return	-
Parameter	alertActivation : Boolean, alertTime : long, alertTone, notifTone : Uri
Method yang terkait	
Logika Proses	<pre>procedure saveSettingToSetting(alertActivation : Boolean, alertTime : Long, alertTone : Uri, notifTone : Uri) begin INSERT INTO Setting (alert_activation, alert_time, alert_tone, notif_tone) VALUES (alertActivation, alertTime, alertTone, notifTone); end endprocedure</pre>

Tabel 117. Spesifikasi proses getAlertActivation

No.	SP - 68
Nama Method	getAlertActivation
Nama Class	ScheduleDatabaseHelper
Requirement	Method ini digunakan untuk mengambil aktivasi <i>alert</i> yang sudah tersimpan di database. Tabel yang terkait dengan method ini adalah tabel Setting.
Deskripsi	[REQ SA - 05]
Return	defaultAlertActivation : boolean
Parameter	-
Method yang terkait	-
Logika Proses	<pre>function getAlertActivation() begin return defaultAlertActivation; end endfunction</pre>

Tabel 118. Spesifikasi proses updateAlertActivation

No.	SP - 69
Nama Method	updateAlertActivation
Nama Class	ScheduleDatabaseHelper
Requirement	Method ini digunakan untuk menyimpan perubahan aktivasi <i>alert</i> ke database. Tabel yang terkait dengan <i>method</i> ini adalah tabel Setting.
Deskripsi	[REQ SA - 05]
Return	-
Parameter	alertActivation : boolean
Method yang terkait	
Logika Proses	<pre>procedure updateAlertActivation(alertActivation : Boolean) begin UPDATE Setting SET alert_activation = alertActivation; end endprocedure</pre>

Tabel 119. Spesifikasi proses getAlertTime

No.	SP - 70
Nama Method	getAlertTime
Nama Class	ScheduleDatabaseHelper
Requirement	Method ini digunakan untuk mengambil waktu untuk menampilkan <i>alert</i> yang sudah tersimpan di database. Tabel yang terkait dengan <i>method</i> ini adalah tabel Setting.
Deskripsi	[REQ SA - 05]
Return	defaultAlertTime : long
Parameter	-
Method yang terkait	-
Logika Proses	<pre>function getAlertTime() begin return defaultAlertTime; end endfunction</pre>

Tabel 120. Spesifikasi proses updateAlertTime

No.	SP - 71
Nama Method	updateAlertTime
Nama Class	ScheduleDatabaseHelper
Requirement	Method ini digunakan untuk menyimpan perubahan waktu untuk menampilkan <i>alert</i> ke database. Tabel yang terkait dengan <i>method</i> ini adalah tabel Setting.
Deskripsi	[REQ SA - 05]
Return	-
Parameter	alertTime : long
Method yang terkait	-
Logika Proses	<pre>procedure updateAlertTime(alertTime : long) begin UPDATE Setting SET alert_time = alertTime; end endprocedure</pre>

Tabel 121. Spesifikasi proses getAlertTone

No.	SP - 72
Nama Method	getAlertTone
Nama Class	ScheduleDatabaseHelper
Requirement	Method ini digunakan untuk mengambil nada dering <i>alert</i> yang sudah tersimpan di database. Tabel yang terkait dengan method ini adalah tabel Setting.
Deskripsi	[REQ SA - 05]
Return	defaultAlertTone : uri
Parameter	-
Method yang terkait	-
Logika Proses	<pre>function getAlertTone() begin return defaultAlertTone; end endfunction</pre>

Tabel 122. Spesifikasi proses updateAlertTone

No.	SP - 73
Nama Method	updateAlertTone
Nama Class	ScheduleDatabaseHelper
Requirement	Method ini digunakan untuk menyimpan perubahan nada dering <i>alert</i> ke database. Tabel yang terkait dengan method ini adalah tabel Setting.
Deskripsi	[REQ SA - 05]
Return	-
Parameter	alertTone : Uri
Method yang terkait	-
Logika Proses	<pre>procedure updateAlertTone(alertTone : Uri) begin UPDATE Setting SET alert_tone = alertTone; end endprocedure</pre>

Tabel 123. Spesifikasi proses getNotifTone

No.	SP - 74
Nama Method	getNotifTone
Nama Class	ScheduleDatabaseHelper
Requirement	Method ini digunakan untuk mengambil nada dering <i>notification</i> yang sudah tersimpan di database. Tabel yang terkait dengan method ini adalah tabel Setting.
Deskripsi	[REQ SA - 05]
Return	defaultNotifTone : uri
Parameter	-
Method yang terkait	-
Logika Proses	<pre>function getNotifTone() begin return defaultNotifTone; end endfunction</pre>

Tabel 124. Spesifikasi proses updateNotifTone

No.	SP - 75
Nama Method	updateNotifTone
Nama Class	ScheduleDatabaseHelper
Deskripsi	Method ini digunakan untuk menyimpan perubahan nada dering <i>notification</i> ke database. Tabel yang terkait dengan <i>method</i> ini adalah tabel Setting.
Requirement	[REQ SA - 05]
Return	-
Parameter	notifTone : Uri
Method yang terkait	-
Logika Proses	<pre>procedure updateNotifTone(notifTone : Uri) begin UPDATE Setting SET notif_tone = notifTone; end endprocedure</pre>

4.3 Perancangan User Interface

Perancangan *user interface* dilakukan berdasarkan hasil analisis kebutuhan terhadap proses masukan dan keluaran yang terjadi pada aplikasi SMS *Scheduler* yang telah dijelaskan sebelumnya pada dokumen SRS bagian *User Interface*, serta untuk memenuhi REQ NF – 03. Perancangan *user interface* dilakukan untuk mempermudah interaksi antara *user* dan sistem pada aplikasi yang akan dibangun. Perancangan *user interface* dan pembuatan *mockup* dilakukan pada *increment* pertama. Hasil perancangan *user interface* aplikasi SMS *Scheduler* dijelaskan pada tabel 125 sampai dengan tabel 165.

Tabel 125. Rancangan UI halaman *New Schedule (Form New Schedule)*

No.	UI – 01
Nama	<i>New Schedule (Form New Schedule)</i>
Deskripsi	Halaman untuk membuat jadwal baru.
Gambar	<p style="text-align: center;">UI - 01.1 : UI <i>New Schedule</i> untuk platform Android</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>New Schedule</p> <p>Recipient(s) <input type="text"/></p> <p>21:19, 22 November 2012 <input type="button" value="Calendar"/></p> <p>Once <input type="button" value="▼"/></p> <p><input type="text"/> 0/160</p> <p><input type="button" value="Save"/> <input type="button" value="Cancel"/></p> </div> <div style="text-align: center;"> <p>New Schedule</p> <p>Recipient(s) <input type="text"/></p> <p>21:19, 22 November 2012 <input type="button" value="Calendar"/></p> <p>Once <input type="button" value="▼"/></p> <p><input type="text"/> 0/160</p> <p>Please choose your schedule's date and time</p> <p><input checked="" type="radio"/> Defined date and time <input type="radio"/> Recipient's date</p> <p><input type="button" value="Ok"/></p> <p><input type="button" value="Save"/> <input type="button" value="Cancel"/></p> </div> </div> <p style="text-align: center;">UI - 01.2 : UI <i>New Schedule</i> untuk platform Blackberry</p> <p>Pada UI ini, perbedaannya yaitu seluruh komponen <i>form</i> akan terpotong dan harus di-<i>scroll</i> untuk melihat komponen <i>form</i> lainnya, dikarenakan pada umumnya layar <i>device</i> Blackberry tidak sebesar layar <i>device</i> Android, iOS dan Java.</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>New Schedule</p> <p>Recipients <input type="text"/></p> <p>21:19, 22 November 2012 <input type="button" value="Calendar"/></p> <p>Once <input type="button" value="▼"/></p> <p><input type="text"/> 0/160</p> <p>Lanjutan dari layar pertama</p> <p><input type="button" value="Save"/> <input type="button" value="Cancel"/></p> </div> <div style="text-align: center;"> <p>New Schedule</p> <p>Recipient(s) <input type="text"/></p> <p>21:19, 22 November 2012 <input type="button" value="Calendar"/></p> <p>Once <input type="button" value="▼"/></p> <p><input type="text"/> 0/160</p> <p>Please choose your schedule's date and time</p> <p><input checked="" type="radio"/> Defined date and time <input type="radio"/> Recipient's date</p> <p><input type="button" value="Ok"/></p> <p>Jika user menginputkan string khusus, yaitu birthday's date atau anniversary's date</p> </div> </div>

No.	UI – 01
Nama	New Schedule (<i>Form New Schedule</i>)

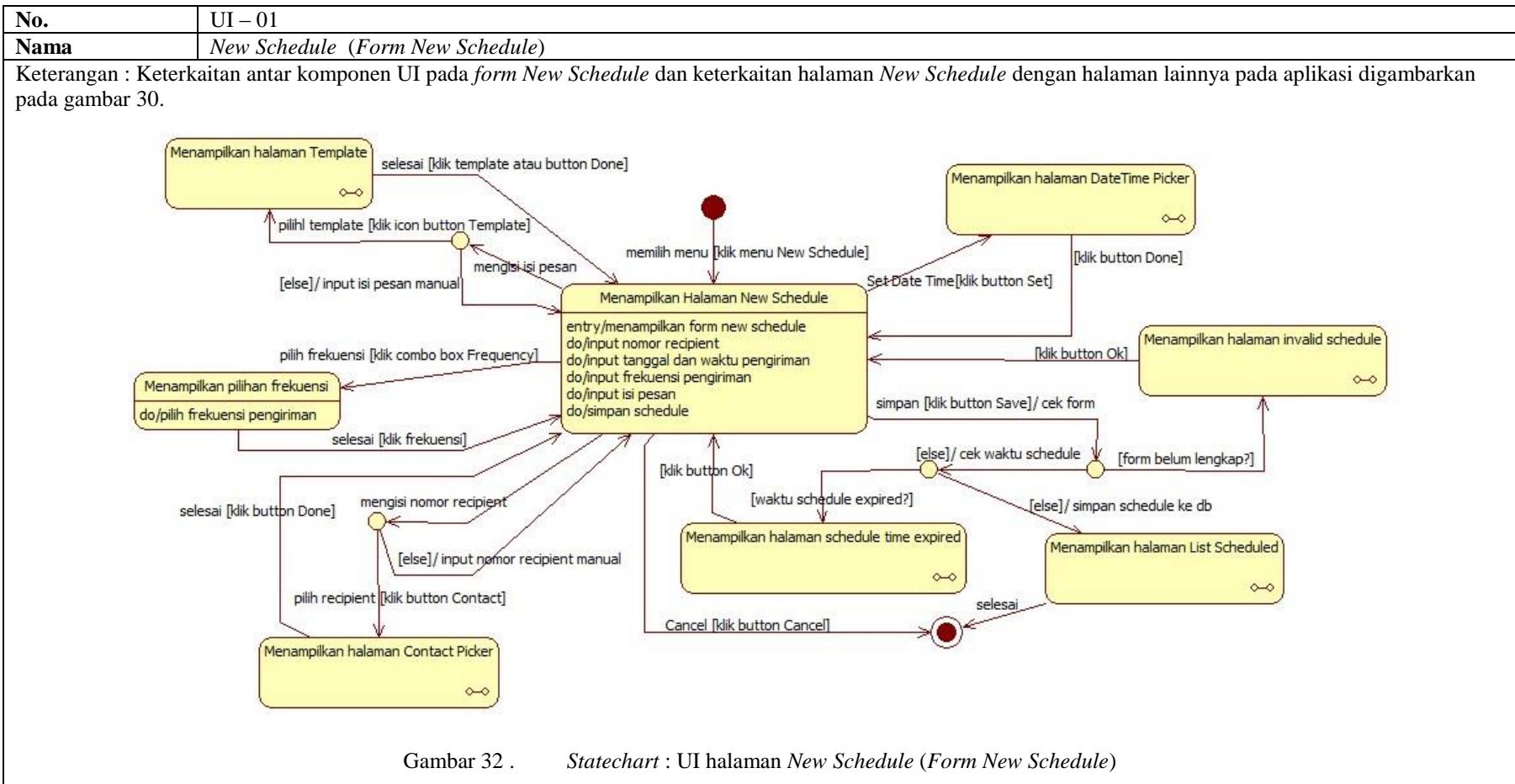
UI - 01.3 : UI New Schedule untuk platform iOS

UI untuk device iOS ini sebetulnya sama seperti Android, perbedaannya hanya pada letak *button cancel*-nya yang berada di bagian atas, sementara pada Android berada di bawah (disamping *button save*).

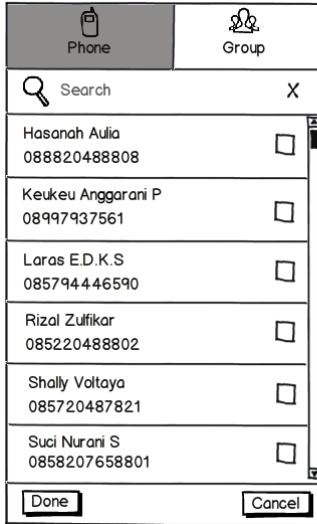
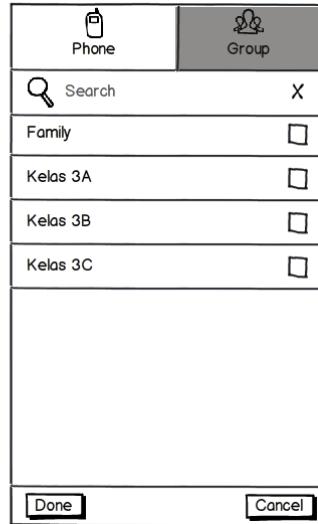
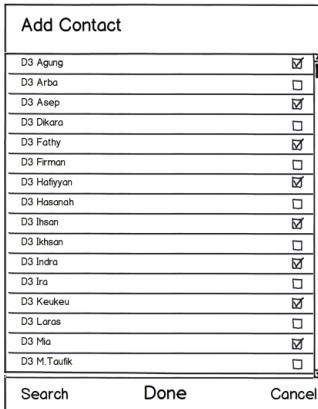
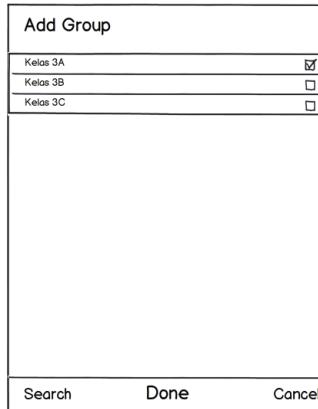
Jika user menginputkan string khusus, yaitu birthday's date atau anniversary's date

UI - 01.4 : UI New Schedule untuk platform Java/Symbian

Jika user menginputkan string khusus, yaitu birthday's date atau anniversary's date

Tabel 126. Rancangan UI halaman *New Schedule (Form New Schedule)* (Lanjutan)Gambar 32 . Statechart : UI halaman *New Schedule (Form New Schedule)*

Tabel 127. Rancangan UI halaman *New Schedule (Contact Picker)*

No.	UI - 02	
Nama	New Schedule (Contact Picker)	
Deskripsi	Halaman untuk memilih recipient dari kontak yang terintegrasi dengan device	
Gambar	 	UI - 02.1 : UI New Schedule (Contact Picker) untuk platform Android, Blackberry dan iOS
	 	UI - 02.2 : UI New Schedule (Contact Picker) untuk platform Java/Symbian

Tabel 128. Rancangan UI halaman *New Schedule (Contact Picker)* (Lanjutan)

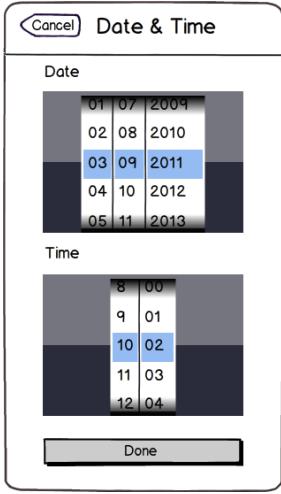
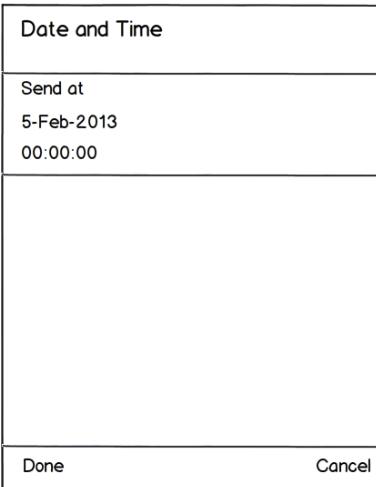
No.	UI - 02
Nama	<i>New Schedule (Contact Picker)</i>
Keterangan : Keterkaitan antar komponen UI pada halaman <i>contact picker</i> dan keterkaitan halaman <i>contact picker</i> dengan halaman lainnya pada aplikasi digambarkan pada gambar 31.	

Gambar 33 . Statechart : UI halaman *contact picker*

Tabel 129. Rancangan UI halaman *New Schedule* (Menampilkan Pengaturan Tanggal dan Waktu)

No.	UI - 03
Nama	<i>New Schedule</i> (Menampilkan pengaturan tanggal dan waktu)
Deskripsi	Halaman untuk mengatur waktu dan tanggal pengiriman pesan.
Gambar	
	UI - 03.1 : UI Date and Time picker untuk platform Android
	UI - 03.2 : UI Date and Time picker untuk platform Blackberry

Tabel 130. Rancangan UI halaman *New Schedule* (Menampilkan Pengaturan Tanggal dan Waktu) (Lanjutan)

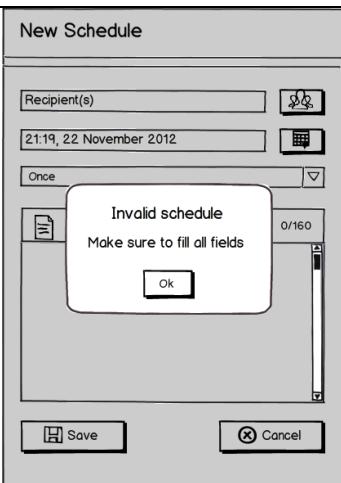
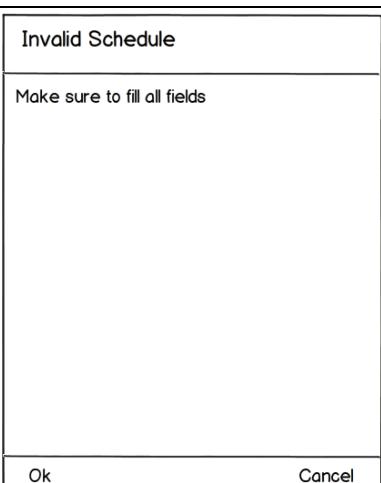
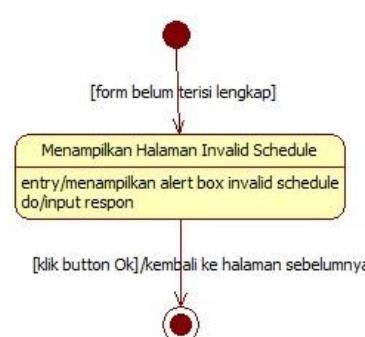
No.	UI - 03
Nama	<i>New Schedule</i> (Menampilkan pengaturan tanggal dan waktu)
Gambar	 <p>UI - 03.3 : UI Date and Time picker untuk platform iOS UI ini sebetulnya sama seperti Android, perbedaannya hanya pada letak button cancel-nya yang berada di bagian kiri atas layar.</p>  <p>UI - 03.4 : UI Date and Time picker untuk platform Java/Symbian</p>
Keterangan :	Keterkaitan antar komponen UI pada halaman <i>date and time picker</i> dan keterkaitan halaman tersebut dengan halaman lainnya pada aplikasi digambarkan pada gambar 32.

```

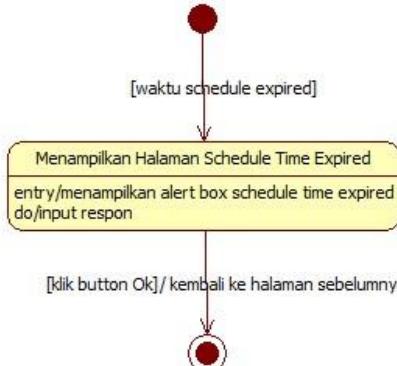
graph TD
    Start(( )) --> SetDate[Set Date Time [klik icon button Set]/ menampilkan waktu sistem]
    SetDate --> Picker[Menampilkan Halaman Date and Time Picker]
    Picker -- "entry/menampilkan waktu seuai device system  
do/tanggal pengiriman pesan  
do/waktu pengiriman pesan" --> Picker
    Picker -- "Cancel [klik button Cancel]" --> Start
    Picker -- "selesai [klik button Done]/ menampilkan halaman sebelumnya" --> End(( ))
    
```

Gambar 34 . Statechart : UI halaman *date and time picker*

Tabel 131. Rancangan UI halaman *New Schedule* (Menampilkan *warning invalid schedule*)

No.	UI – 04	
Nama	<i>New Schedule (warning invalid schedule)</i>	
Deskripsi	Halaman yang menampilkan pesan peringatan ketika <i>user</i> belum mengisi semua <i>field</i> pada <i>form new schedule</i>	
Gambar		
	UI - 04.1 : UI warning invalid schedule untuk platform Android, Blackberry dan iOS 	UI - 04.2 : UI warning invalid schedule untuk platform Java/Symbian
<p>Keterangan : Keterkaitan antar komponen UI pada halaman <i>warning invalid schedule</i> dan keterkaitan halaman tersebut dengan halaman lainnya pada aplikasi digambarkan pada gambar 33.</p>  <pre> graph TD Start(()) --> Display[Menampilkan Halaman Invalid Schedule entry/menampilkan alert box invalid schedule do/input respon] Display --> ClickOk[[klik button Ok]/kembali ke halaman sebelumnya] ClickOk --> End(()) </pre>		
<p align="center">Gambar 35 . Statechart : UI halaman <i>warning invalid schedule</i></p>		

Tabel 132. Rancangan UI halaman *New Schedule* (Menampilkan *warning schedule time expired*)

No.	UI - 05
Nama	<i>New Schedule (warning schedule time expired)</i>
Deskripsi	Halaman yang menampilkan pesan peringatan ketika <i>user</i> salah memasukkan tanggal atau waktu <i>schedule</i>
Gambar	
	 <p>UI - 05.1 : UI warning schedule time expired untuk platform Android, Blackberry dan iOS</p> <p>Schedule time is expired</p> <p>Ok Cancel</p> <p>UI - 05.1 : UI warning schedule time expired untuk platform Java/Symbian</p>
<p>Keterangan : Keterkaitan antar komponen UI pada halaman <i>warning schedule time expired</i> dan keterkaitan halaman tersebut dengan halaman lainnya pada aplikasi digambarkan pada gambar 34.</p>  <pre> graph TD Start(()) -- "[waktu schedule expired]" --> A[Menampilkan Halaman Schedule Time Expired] A -- "[klik button Ok]/ kembali ke halaman sebelumnya" --> End(()) </pre>	
<p>Gambar 36 . Statechart : UI halaman <i>warning schedule time expired</i></p>	

Tabel 133. Rancangan UI halaman *Edit Schedule (Edit All)*

No.	UI - 06
Nama	<i>Edit Schedule (Edit All)</i>
Deskripsi	Halaman untuk melakukan perubahan jadwal yang sebelumnya telah dibuat dengan menampilkan form edit schedule secara keseluruhan.
Gambar	<p>UI - 06.1 : UI Edit All Schedule untuk platform Android</p>

UI - 06.2 : UI Edit All Schedule untuk platform Blackberry

Pada UI ini, perbedaannya yaitu seluruh komponen *form* akan terpotong dan harus di-*scroll* untuk melihat komponen *form* lainnya, dikarenakan pada umumnya layar *device* Blackberry tidak sebesar layar *device* Android, iOS dan Java.

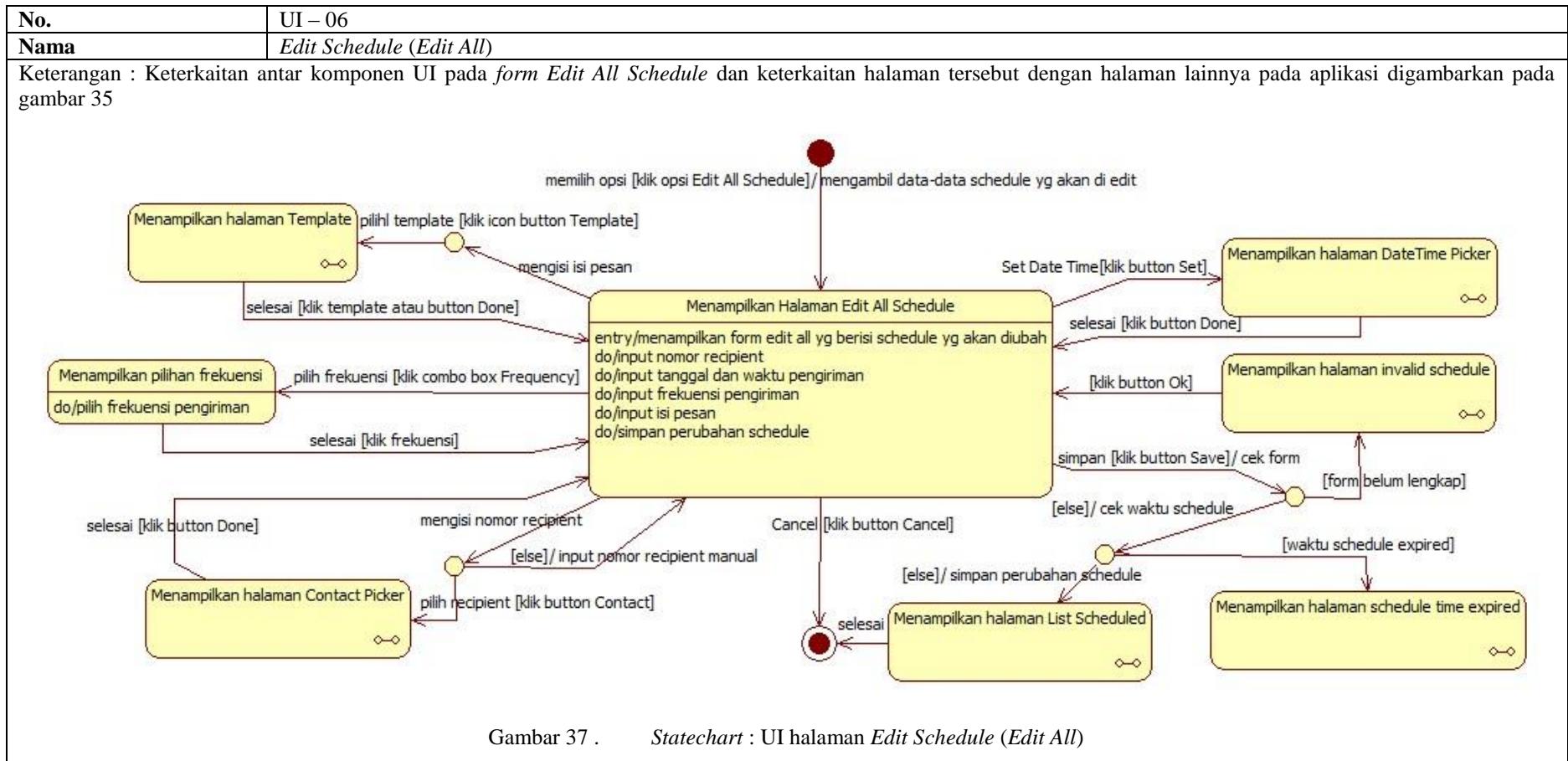
--	--

Tabel 134. Rancangan UI halaman *Edit Schedule (Edit All)* (lanjutan)

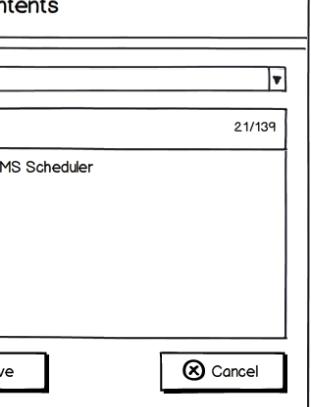
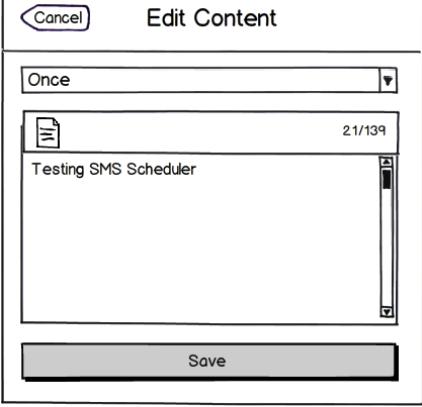
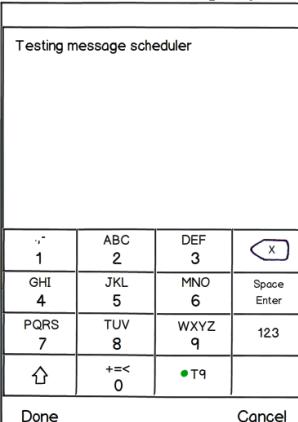
No.	UI - 06
Nama	<i>Edit Schedule (Edit All)</i>
Gambar	
<p style="text-align: center;">UI - 06.3 : UI <i>Edit All Schedule</i> untuk platform iOS</p> <p>UI untuk <i>device</i> iOS ini sebetulnya sama seperti Android, perbedaannya hanya pada letak <i>button cancel</i>-nya yang berada di bagian atas, sementara pada Android berada di bawah (disamping <i>button save</i>).</p>	

UI - 06.4 : UI *Edit All Schedule* untuk platform Java/Symbian

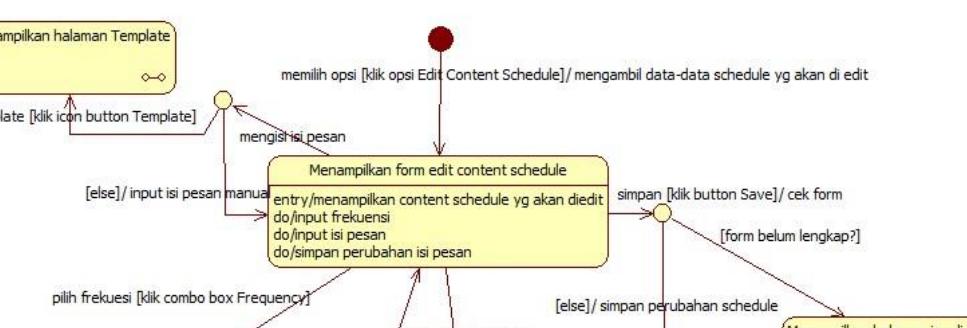
--	--

Tabel 135. Rancangan UI halaman *Edit Schedule (Edit All)* (Lanjutan)

Tabel 136. Rancangan UI halaman *Edit Schedule (Edit Content)*

No.	UI - 07
Nama	Edit Schedule (Edit Content)
Deskripsi	Halaman untuk melakukan perubahan isi pesan dari jadwal yang sebelumnya telah dibuat.
Gambar	 <p>UI - 07.1 : UI Edit Content untuk platform Android</p>  <p>UI - 07.3 : UI Edit Content untuk platform iOS</p> <p>UI ini sebetulnya sama seperti Android, perbedaannya hanya pada letak button cancel-nya yang berada di bagian kiri atas.</p>
	<p>UI - 07.4 : UI Edit Content untuk platform Java/Symbian</p> 

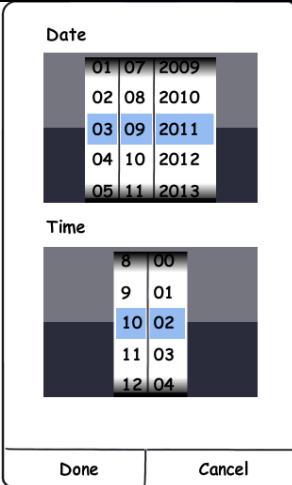
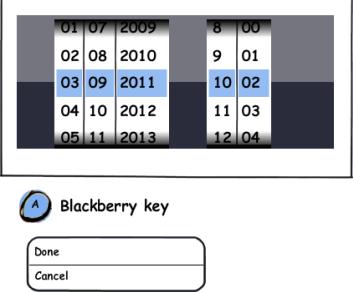
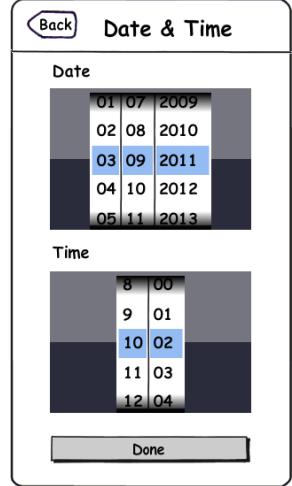
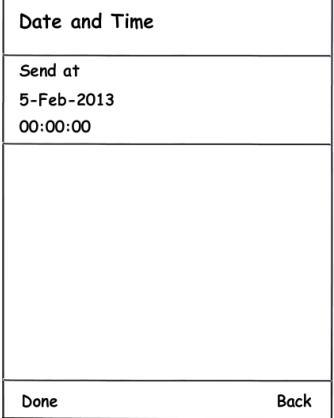
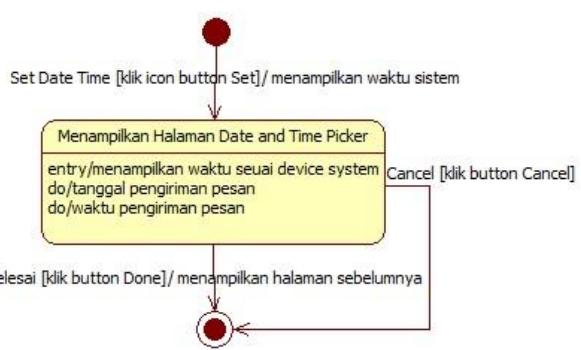
Keterangan : Keterkaitan antar komponen UI pada form *Edit Content Schedule* dan keterkaitan halaman tersebut dengan halaman lainnya pada aplikasi digambarkan pada gambar 36.



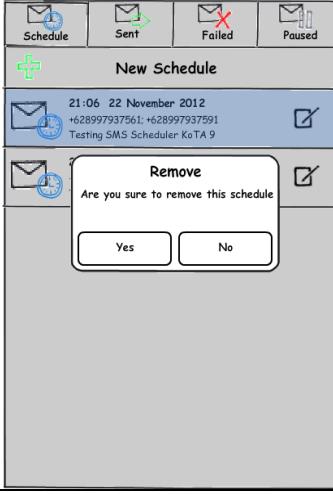
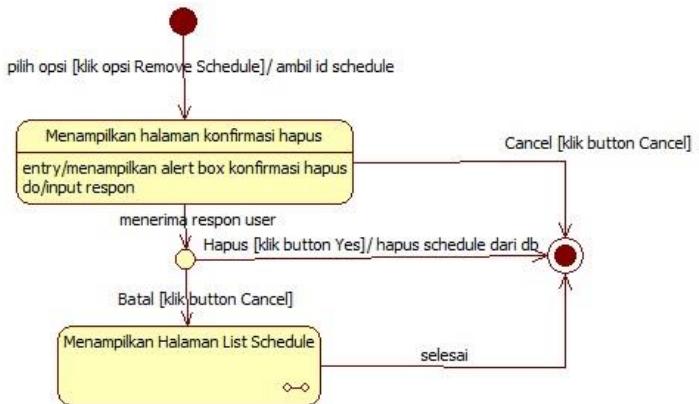
Tabel 137. Rancangan UI halaman *Edit Schedule (Edit Recipient)*

No.	UI - 08
Nama	<i>Edit Schedule (Edit Recipient)</i>
Deskripsi	Halaman untuk melakukan perubahan penerima pesan dari jadwal yang sebelumnya telah dibuat.
Gambar	<p>UI - 08.1 : UI <i>Edit Recipient</i> untuk <i>platform</i> Android dan iOS</p> <p>Recipient(s)</p> <p>08123456778 Kelas 3A D3 Laras D3 Rizal Z</p> <p>UI - 08.4 : UI <i>Edit Recipient</i> untuk <i>platform</i> Java/Symbian</p>
	<p>UI - 08.3 : UI <i>Edit Recipient</i> untuk <i>platform</i> iOS</p> <p>UI ini sebetulnya sama seperti Android, perbedaannya hanya pada letak <i>button back</i>-nya yang berada di bagian kiri atas.</p>
	<p>Keterangan : Keterkaitan antar komponen UI pada <i>form Edit Recipient Schedule</i> dan keterkaitan halaman tersebut dengan halaman lainnya pada aplikasi digambarkan pada gambar 36.</p> <pre> graph TD Start(()) -- "memilih opsi [klik opsi EditRecipient Schedule]/ mengambil data-data schedule yg akan di edit" --> Form[Menampilkan form Edit Recipient Schedule] Form -- "entry/menampilkan recipient schedule yg akan diedit do/input recipient do/simpan perubahan recipient" --> Save[simpan [klik button Save]/ cek form [form belum lengkap?]] Save -- "[else]/ simpan perubahan schedule" --> Done[Menampilkan halaman List Scheduled] Save -- "[cek form [form belum lengkap?]]" --> Invalid[Menampilkan halaman invalid schedule] Done -- "selesai [klik button Done]" --> End(()) Done -- "Cancel [klik button Cancel]" --> Contact[Menampilkan halaman Contact Picker] Contact -- "pilih recipient [klik button Contact]" --> Done </pre> <p>Gambar 39 . Statechart : UI halaman <i>Edit Schedule (Edit Recipient)</i></p>

Tabel 138. Rancangan UI halaman *Edit Schedule (Edit Date and Time)*

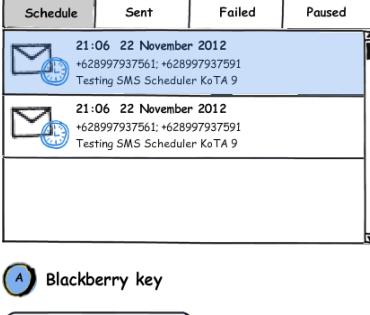
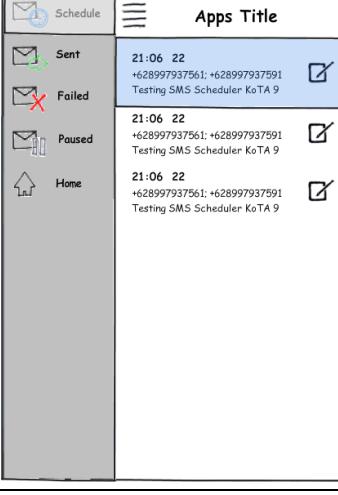
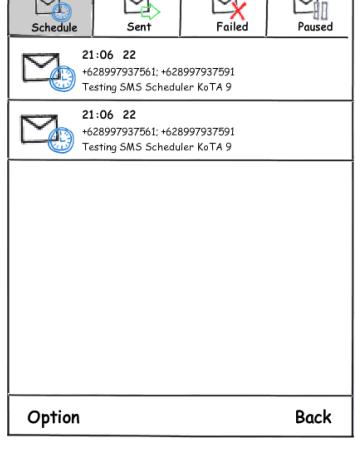
No.	UI - 09		
Nama	<i>Edit Schedule (Edit Date and Time)</i>		
Deskripsi	Halaman untuk melakukan perubahan tanggal dan waktu pengiriman pesan dari jadwal yang sebelumnya telah dibuat.		
Gambar			
	UI - 09.1 : UI Edit Date and Time untuk platform Android	 A BlackBerry key	UI - 09.2 : UI Edit Date and Time untuk platform BlackBerry
	UI - 09.3 : UI Edit Date and Time untuk platform iOS UI ini sebetulnya sama seperti Android, perbedaannya hanya pada letak button cancel-nya yang berada di bagian kiri atas layar.		UI - 09.4 : UI Edit Date and Time untuk platform Java/Symbian
Keterangan : Keterkaitan antar komponen UI pada halaman <i>Edit Date and Time Schedule</i> dan keterkaitan halaman tersebut dengan halaman lainnya pada aplikasi digambarkan pada gambar 38.			
 <pre> graph TD Start(()) --> Set[Set Date Time [klik icon button Set]/ menampilkan waktu sistem] Set --> Picker[Menampilkan Halaman Date and Time Picker
entry/menampilkan waktu seuai device system
do/tanggal pengiriman pesan
do/waktu pengiriman pesan] Picker --> Done[selesai [klik button Done]/ menampilkan halaman sebelumnya] Picker --> Cancel[Cancel [klik button Cancel]] </pre>			
Gambar 40 . Statechart : UI halaman <i>Edit Schedule (Edit Date and Time)</i>			

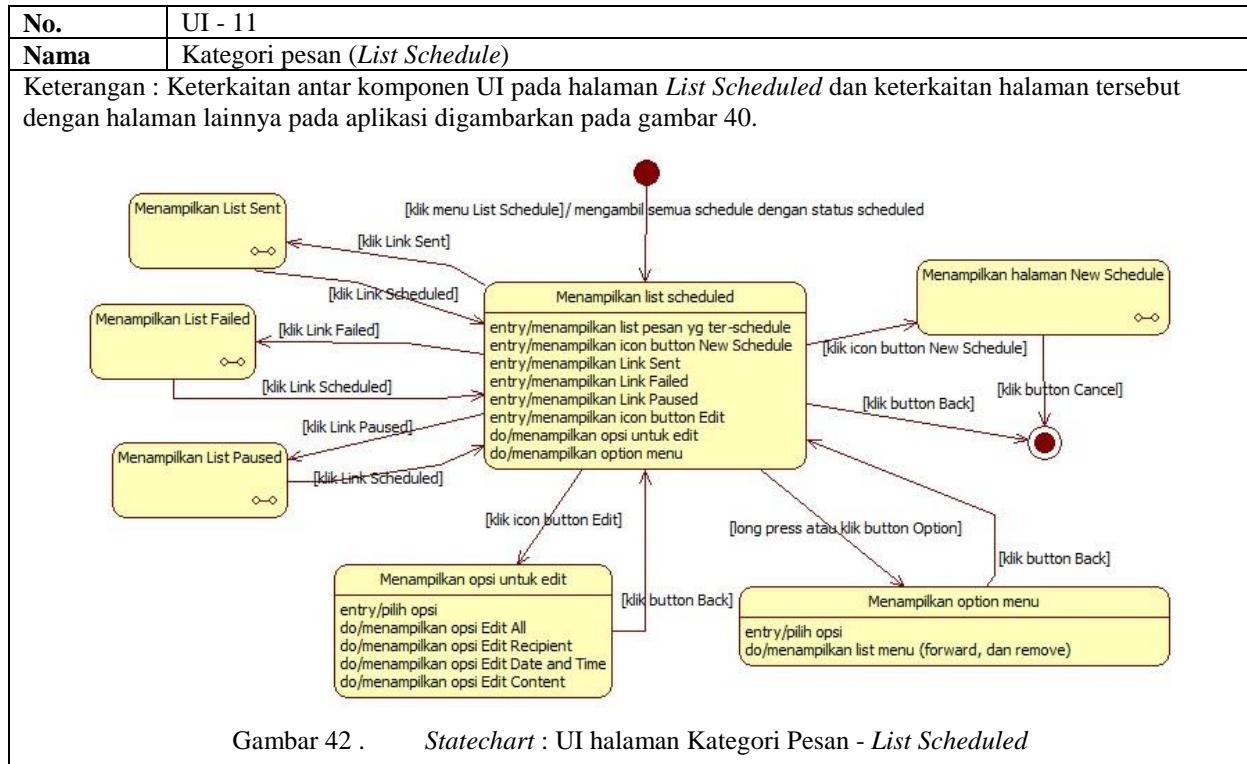
Tabel 139. Rancangan UI halaman menghapus *schedule*

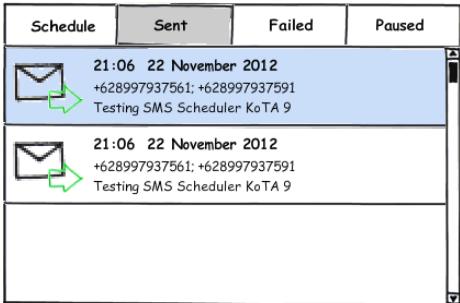
No.	UI - 10
Nama	Menghapus <i>schedule</i>
Deskripsi	Halaman untuk melakukan penghapusan jadwal yang sebelumnya telah dibuat.
Gambar	 <p>UI - 10.1 : UI menghapus <i>schedule</i> untuk platform Android, Blackberry dan iOS</p>  <p>UI - 10.4 : UI menghapus <i>schedule</i> untuk platform Java/Symbian</p>
Keterangan :	Keterkaitan antar komponen UI pada halaman menghapus <i>schedule</i> dan keterkaitan halaman tersebut dengan halaman lainnya pada aplikasi digambarkan pada gambar 32.
	 <pre> graph TD Start(()) -- "pilih opsi [klik opsi Remove Schedule]/ ambil id schedule" --> Step1[Menampilkan halaman konfirmasi hapus
entry/menampilkan alert box konfirmasi hapus
do/input respon] Step1 -- "Cancel [klik button Cancel]" --> Start Step1 -- "menerima respon user" --> Step2[Hapus [klik button Yes]/ hapus schedule dari db] Step2 -- "Batal [klik button Cancel]" --> Start Step2 -- "selesai" --> End((())) </pre>

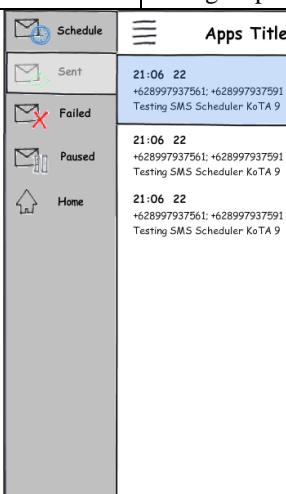
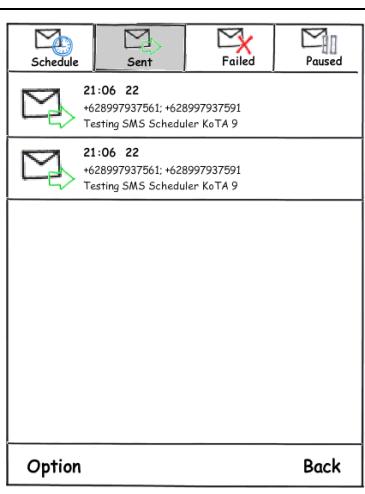
Gambar 41 . Statechart : UI halaman menghapus *schedule*

Tabel 140. Rancangan UI halaman kategori pesan (*List Schedule*)

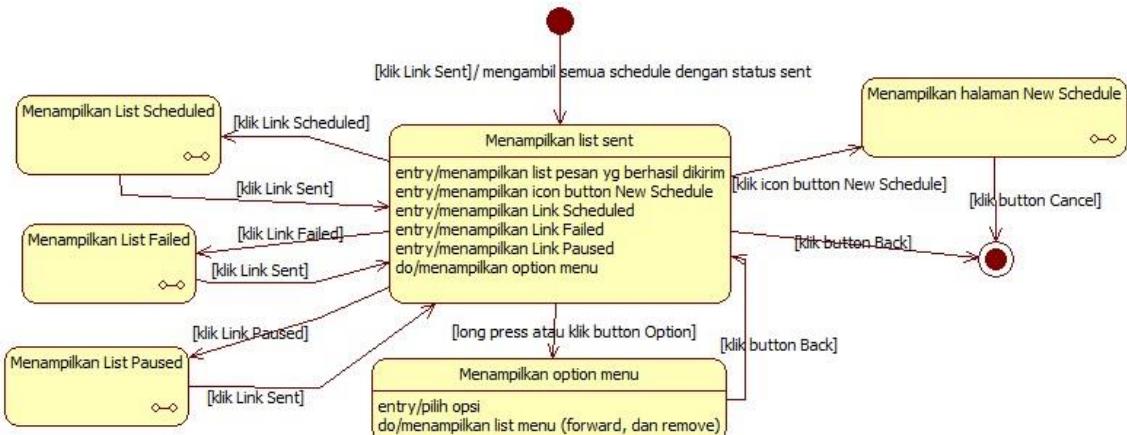
No.	UI - 11		
Nama	Kategori pesan (<i>List Schedule</i>)		
Deskripsi	Halaman untuk menampilkan <i>list</i> dari <i>schedule-schedule</i> yang sebelumnya telah dibuat.		
Gambar	 <p>UI - 11.1 : UI List Schedule untuk platform Android</p>	 <p>UI - 11.2 : UI List Schedule untuk platform BlackBerry</p>	<p>Pada UI ini, perbedaanya adalah tidak adanya gambar untuk masing-masing kategori</p>
	 <p>UI - 11.3 : UI List Schedule untuk platform iOS</p>	 <p>UI - 11.4 : UI List Schedule untuk platform Java/Symbian</p>	<p>Pada UI ini, perbedaannya ada pada button Option dan Back yang ada dibagian bawah layar</p>

Tabel 141. Rancangan UI halaman kategori pesan (*List Sent*)

No.	UI - 12
Nama	Kategori pesan (<i>List Sent</i>)
Deskripsi	Halaman untuk menampilkan <i>list</i> dari pesan yang berhasil dikirim.
Gambar	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>UI - 12.1 : UI <i>List Sent</i> untuk platform Android</p> </div> <div style="text-align: center;">  <p>UI - 12.2 : UI <i>List Sent</i> untuk platform BlackBerry</p> <p>Pada UI ini, perbedaanya adalah tidak adanya gambar untuk masing-masing kategori</p> </div> </div>

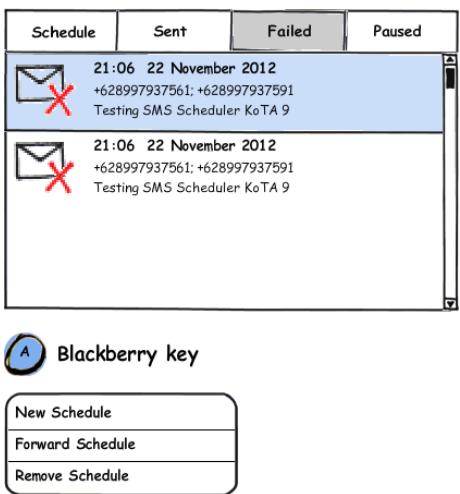
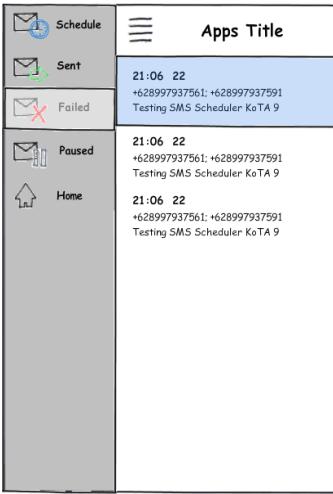
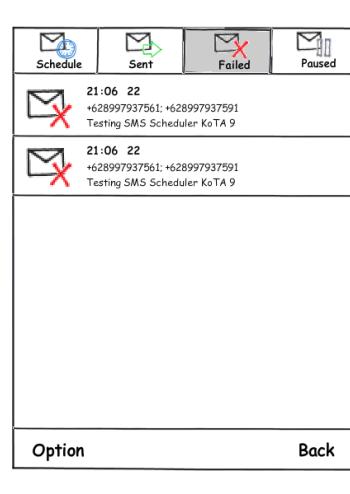
No.	UI - 12
Nama	Kategori pesan (<i>List Sent</i>)
	<p>UI - 12.3 : UI <i>List Sent</i> untuk platform iOS</p>  <p>UI - 12.4 : UI <i>List Sent</i> untuk platform Java/Symbian</p> <p>Pada UI ini, perbedaannya ada pada button Option dan Back yang ada dibagian bawah layar</p>

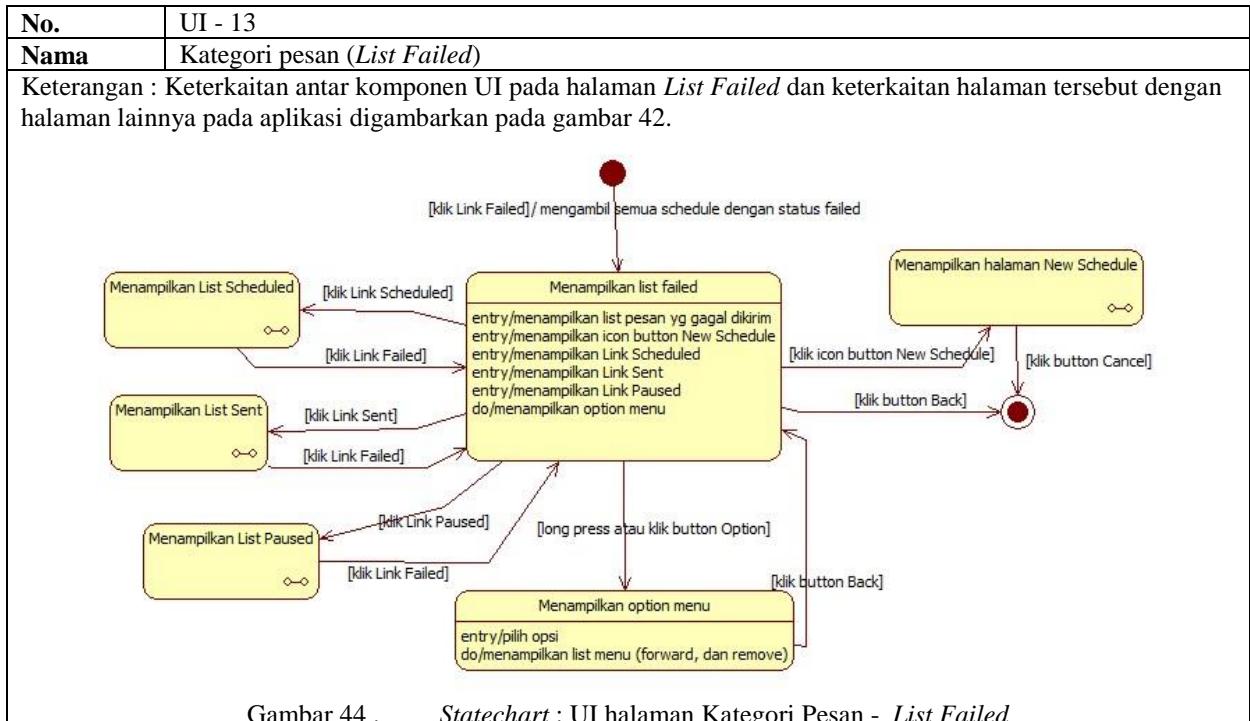
Keterangan : Keterkaitan antar komponen UI pada halaman *List Sent* dan keterkaitan halaman tersebut dengan halaman lainnya pada aplikasi digambarkan pada gambar 41.

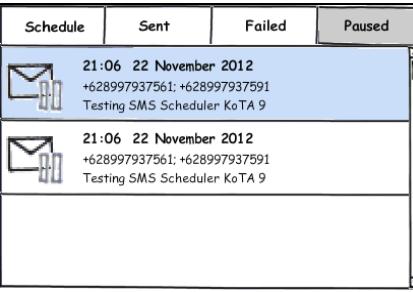
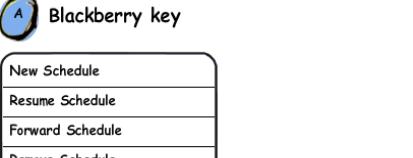


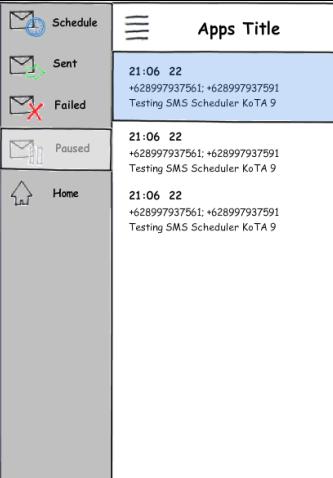
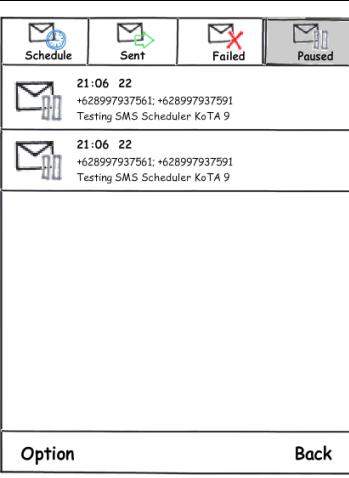
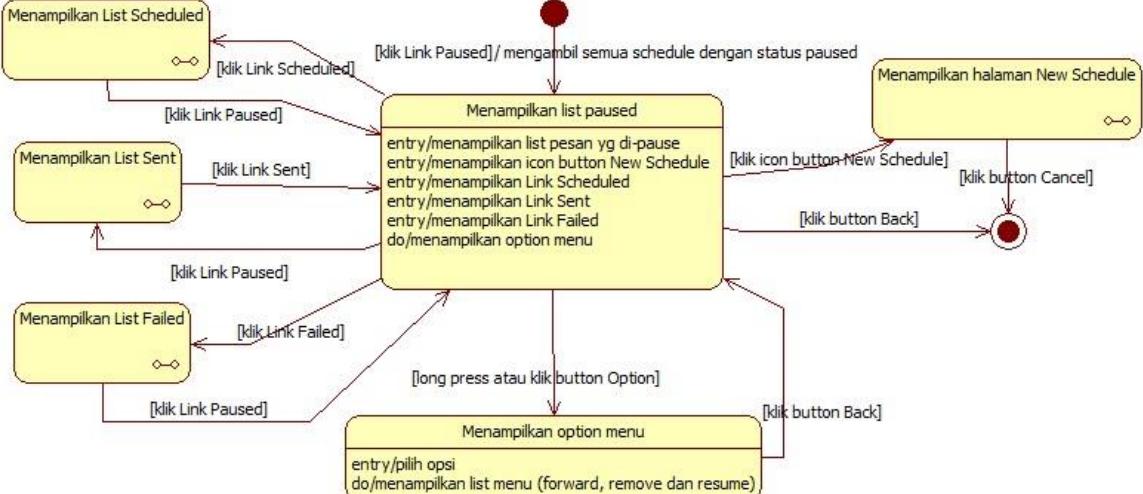
Gambar 43 . Statechart : UI halaman Kategori Pesan - *List Sent*

Tabel 142. Rancangan UI halaman kategori pesan (*List Failed*)

No.	UI - 13		
Nama	Kategori pesan (<i>List Failed</i>)		
Deskripsi	Halaman untuk menampilkan <i>list</i> dari pesan yang gagal dikirim.		
Gambar	 <p>UI - 13.1 : UI <i>List Failed</i> untuk platform Android</p>	 <p>UI - 13.2 : UI <i>List Failed</i> untuk platform BlackBerry</p>	<p>Pada UI ini, perbedaanya adalah tidak adanya gambar untuk masing-masing kategori</p>
	 <p>UI - 13.3 : UI <i>List Failed</i> untuk platform iOS</p>	 <p>UI - 13.4 : UI <i>List Failed</i> untuk platform Java/Symbian</p>	<p>Pada UI ini, perbedaannya ada pada button Option dan Back yang ada dibagian bawah layar</p>

Gambar 44 . Statechart : UI halaman Kategori Pesan - *List Failed*Tabel 143. Rancangan UI halaman kategori pesan (*List Paused*)

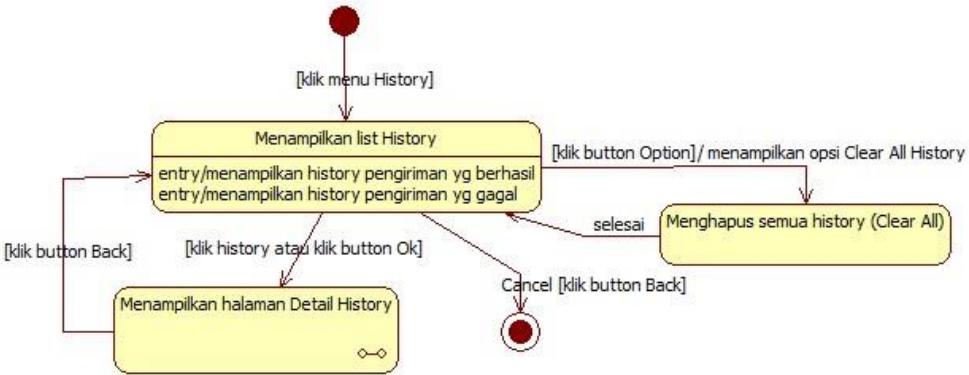
No.	UI - 14
Nama	Kategori pesan (<i>List Paused</i>)
Deskripsi	Halaman untuk menampilkan <i>list</i> dari <i>schedule-schedule</i> yang pengirimannya ditunda.
Gambar	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>UI - 14.2 : UI <i>List Paused</i> untuk platform BlackBerry</p> <p>Pada UI ini, perbedaanya adalah tidak adanya gambar untuk masing-masing kategori</p> </div> <div style="text-align: center;">  <p>UI - 14.1 : UI <i>List Paused</i> untuk platform Android</p> </div> </div>

No.	UI - 14				
Nama	Kategori pesan (<i>List Paused</i>)				
	 <p>UI - 14.3 : UI <i>List Paused</i> untuk platform iOS</p>	 <p>UI - 14.4 : UI <i>List Paused</i> untuk platform Java/Symbian</p>	Pada UI ini, perbedaannya ada pada button Option dan Back yang ada dibagian bawah layar		
<p>Keterangan : Keterkaitan antar komponen UI pada halaman <i>List Paused</i> dan keterkaitan halaman tersebut dengan halaman lainnya pada aplikasi digambarkan pada gambar 43.</p>  <p>Gambar 45 . Statechart : UI halaman Kategori Pesan - <i>List Paused</i></p>					

Tabel 144. Rancangan UI halaman *history*

No.	UI - 15
Nama	<i>History</i>
Deskripsi	Halaman untuk menampilkan <i>list</i> dari pesan yang terkirim dan gagal terkirim.
Gambar	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>UI - 15.1 : UI <i>History</i> untuk platform Android dan Blackberry</p> </div> <div style="text-align: center;">  <p>UI - 15.2 : UI <i>History</i> untuk platform iOS</p> </div> </div> <div style="text-align: center;"> <p>UI - 15.3 : UI <i>History</i> untuk platform Java/Symbian</p>  </div>

Keterangan : Keterkaitan antar komponen UI pada halaman *history* dan keterkaitan halaman tersebut dengan halaman lainnya pada aplikasi digambarkan pada gambar 44.

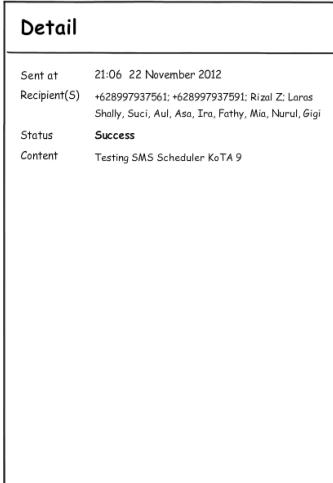
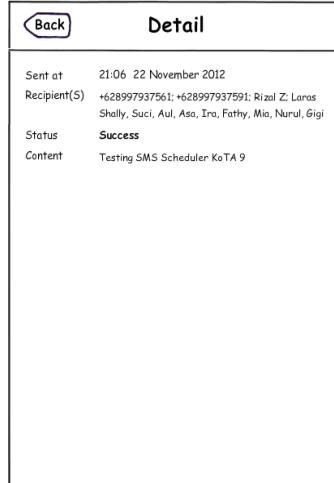
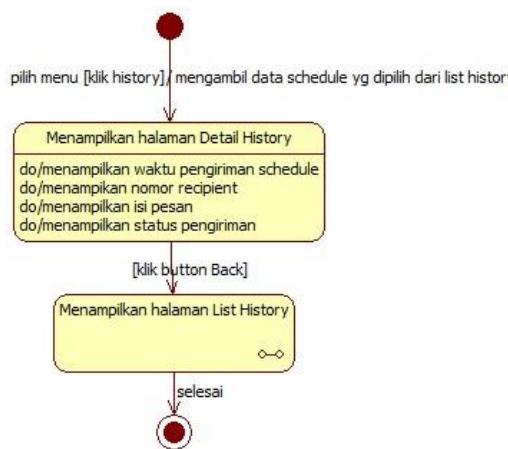


```

graph TD
    Start(( )) -- "[klik menu History]" --> HistoryList[Menampilkan list History]
    HistoryList -- "[klik history atau klik button Ok]" --> DetailHistory[Menampilkan halaman Detail History]
    DetailHistory -- "[klik button Back]" --> Start
    HistoryList -- "[klik button Option] / menampilkan opsi Clear All History" --> ClearAll[Menghapus semua history (Clear All)]
    ClearAll -- "selesai" --> End(( ))
    
```

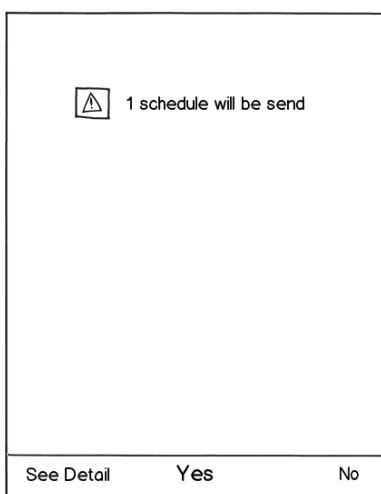
Gambar 46 . UI halaman *history*

Tabel 145. Rancangan UI halaman *detail history*

No.	UI - 16
Nama	<i>Detail history</i>
Deskripsi	Halaman untuk menampilkan <i>detail</i> dari <i>history</i> yang sebelumnya dipilih.
Gambar	 <p>UI - 16.1 : UI <i>Detail history</i> untuk platform Android, BlackBerry dan Java</p>  <p>UI - 16.3 : UI <i>Detail history</i> untuk platform iOS</p> <p>Pada UI ini, perbedaannya yaitu button Back yang berada di bagian kiri atas layar</p>
Keterangan :	Keterkaitan antar komponen UI pada halaman <i>detail history</i> dan keterkaitan halaman tersebut dengan halaman lainnya pada aplikasi digambarkan pada gambar 45.
	 <pre> graph TD Start(()) -- "pilih menu [klik history], mengambil data schedule yg dipilih dari list history" --> DetailHistory[Menampilkan halaman Detail History
do/menampilkan waktu pengiriman schedule
do/menampilkan nomor recipient
do/menampilkan isi pesan
do/menampilkan status pengiriman] DetailHistory -- "[klik button Back]" --> ListHistory[Menampilkan halaman List History] ListHistory -- "selesai" --> End((())) </pre>

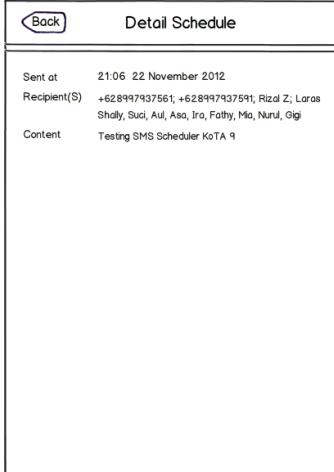
Gambar 47 . Statechart : UI halaman *detail history*

Tabel 146. Rancangan UI halaman Alert

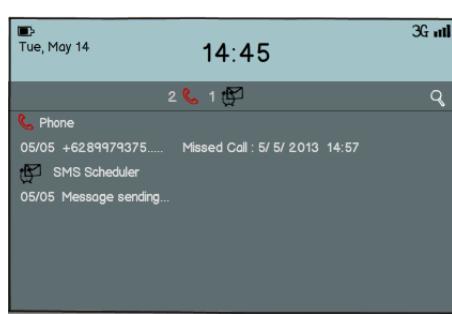
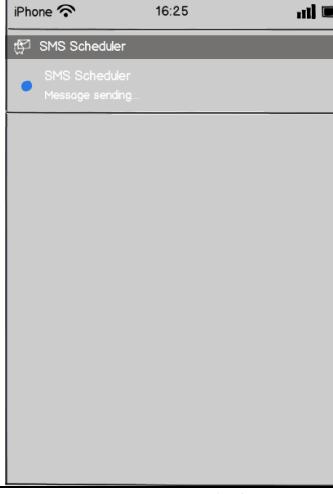
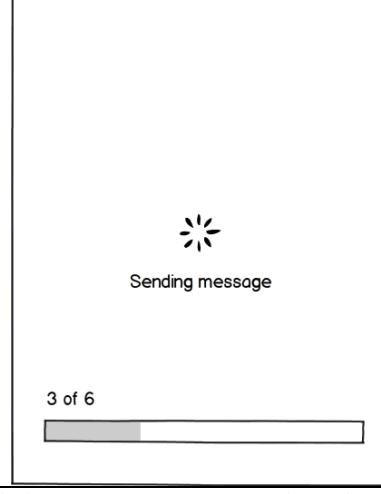
No.	UI - 17
Nama	Alert
Deskripsi	Halaman ketika memunculkan <i>alert box</i> sebelum pengiriman pesan.
Gambar	 <p>UI - 17.1 : UI alert untuk platform Android, Blackberry dan iOS</p>  <p>UI - 17.4 : UI alert untuk platform Java/ Symbian</p>
Keterangan :	Keterkaitan antar komponen UI pada halaman <i>alert</i> dan keterkaitan halaman tersebut dengan halaman lainnya pada aplikasi digambarkan pada gambar 46.
	

Gambar 48 . Statechart : UI halaman alert

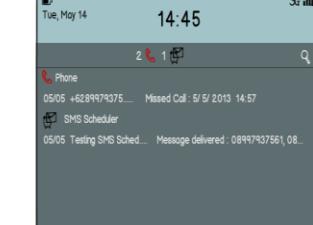
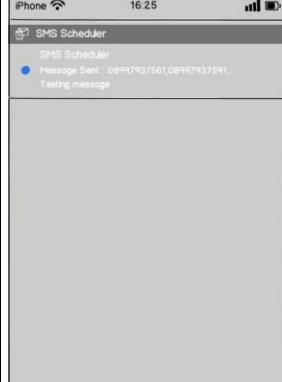
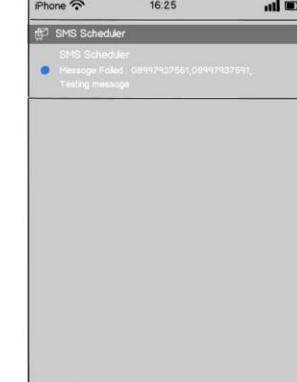
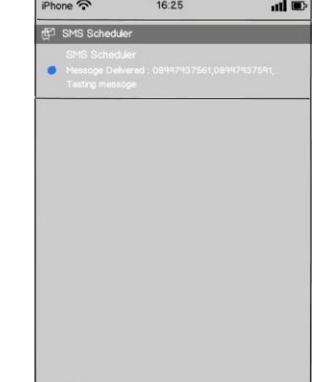
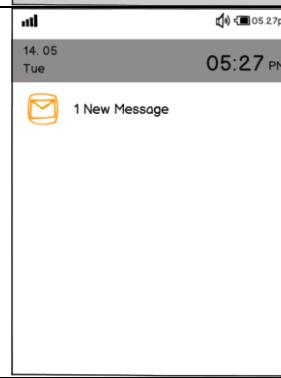
Tabel 147. Rancangan UI halaman *Alert (detail alert)*

No.	UI - 18		
Nama	<i>Alert (detail alert)</i>		
Deskripsi	Halaman yang menampilkan <i>detail schedule</i> yang akan dikirim.		
Gambar	 <p>UI - 18.1 : UI alert (detail alert) untuk platform Android, Blackberry dan Java</p> <p>A Klik button back pada Android</p>		
		<p>UI - 18.2 : UI alert (detail alert) untuk platform iOS</p>	
Keterangan :	Keterkaitan antar komponen UI pada halaman <i>detail alert</i> dan keterkaitan halaman tersebut dengan halaman lainnya pada aplikasi digambarkan pada gambar 47.		
	 <pre> graph TD Start(()) --> Action1["[klik button See Detail]/ mengambil data schedule yg akan dikirim"] Action1 --> State1[Menampilkan halaman Detail Alert] State1 --> Action2["do/menampilkan waktu pengiriman schedule yg akan dikirim do/menampilkan nomor recipient dari schedule yg akan dikirim do/menampilkan isi pesan dari schedule yg akan dikirim"] Action2 --> Action3["[klik button Back]"] Action3 --> Start </pre>		
	Gambar 49 . Statechart : UI halaman <i>detail alert</i>		

Tabel 148. Rancangan UI halaman *Notification (sending)*

No.	UI - 19
Nama	<i>Notification (sending)</i>
Deskripsi	Halaman yang menampilkan notifikasi saat proses pengiriman berlangsung.
Gambar	 <p>UI - 19.1 : UI alert (detail alert) untuk platform Android</p>  <p>UI - 19.2 : UI alert (detail alert) untuk platform BlackBerry</p>
 <p>UI - 19.3 : UI alert (detail alert) untuk platform iOS</p>	 <p>UI - 19.4 : UI alert (detail alert) untuk platform Java/ Symbian</p>
<p>Keterangan : Keterkaitan antar komponen UI pada halaman <i>notification sending</i> dan keterkaitan halaman tersebut dengan halaman lainnya pada aplikasi digambarkan pada gambar 48.</p>  <pre> graph TD Start(()) -- "[proses mengirimkan pesan]/ ambil data schedule" --> Task[Menampilkan notifikasi mengirimkan schedule
do/menampilkan kalimat \"Message Sending\""] Task --> End(()) labeled "selesai" </pre>	
<p>Gambar 50 . Statechart : UI halaman <i>notification sending</i></p>	

Tabel 149. Rancangan UI halaman *Notification (laporan pengiriman)*

No.	UI - 20						
Nama	<i>Notification (laporan pengiriman)</i>						
Deskripsi	Halaman yang menampilkan notifikasi saat pesan berhasil dikirim, gagal dikirim, dan saat sudah diterima oleh <i>recipient</i>						
Gambar				UI - 20.1 : UI <i>notification</i> untuk <i>platform</i> Android			
				UI - 20.2 : UI <i>notification</i> untuk <i>platform</i> Blackberry			
				UI - 20.3 : UI <i>notification</i> untuk <i>platform</i> iOS			
				UI - 20.4 : UI <i>notification</i> untuk <i>platform</i> Java/ Symbian			

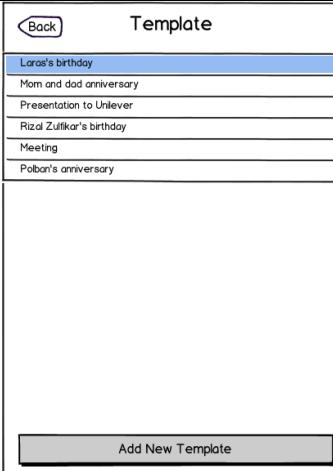
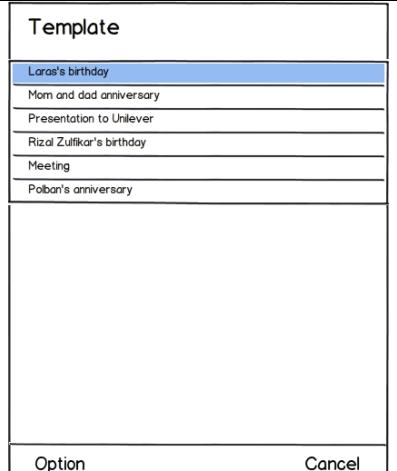
Tabel 150. Rancangan UI halaman *Notification (laporan pengiriman)* (Lanjutan)

No.	UI - 20
Nama	<i>Notification (laporan pengiriman)</i>
Keterangan : Keterkaitan antar komponen UI pada halaman <i>notification laporan pengiriman</i> dan keterkaitan halaman tersebut dengan halaman lainnya pada aplikasi digambarkan pada gambar 49.	

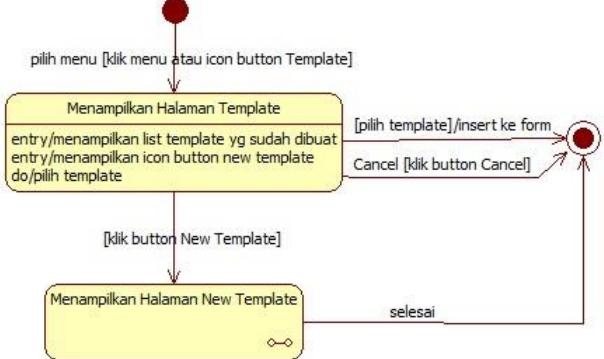
Gambar 51 . Statechart : UI halaman *notification (laporan pengiriman)*

Tabel 151. Rancangan UI halaman *template*

No.	UI - 21														
Nama	<i>Template</i>														
Deskripsi	Halaman untuk menampilkan <i>list</i> dari <i>template-template</i> yang sebelumnya sudah dibuat dan juga untuk membuat <i>template</i> baru.														
Gambar															
<table border="1"> <thead> <tr> <th>Template</th> </tr> </thead> <tbody> <tr> <td>Laras's birthday</td> </tr> <tr> <td>Mom and dad anniversary</td> </tr> <tr> <td>Presentation to Unilever</td> </tr> <tr> <td>Rizal Zulfikar's birthday</td> </tr> <tr> <td>Meeting</td> </tr> <tr> <td>Polban's anniversary</td> </tr> </tbody> </table> <p style="text-align: center;">UI - 21.1 : UI template untuk platform Android</p> <p style="text-align: center;">+ Add New Template</p>	Template	Laras's birthday	Mom and dad anniversary	Presentation to Unilever	Rizal Zulfikar's birthday	Meeting	Polban's anniversary	<table border="1"> <thead> <tr> <th>Template</th> </tr> </thead> <tbody> <tr> <td>Laras's birthday</td> </tr> <tr> <td>Mom and dad anniversary</td> </tr> <tr> <td>Presentation to Unilever</td> </tr> <tr> <td>Rizal Zulfikar's birthday</td> </tr> <tr> <td>Meeting</td> </tr> <tr> <td>Polban's anniversary</td> </tr> </tbody> </table> <p style="text-align: center;">UI - 21.2 : UI template untuk platform Blackberry</p> <p style="text-align: center;">A Blackberry key</p> <p style="text-align: center;">Add New Template</p>	Template	Laras's birthday	Mom and dad anniversary	Presentation to Unilever	Rizal Zulfikar's birthday	Meeting	Polban's anniversary
Template															
Laras's birthday															
Mom and dad anniversary															
Presentation to Unilever															
Rizal Zulfikar's birthday															
Meeting															
Polban's anniversary															
Template															
Laras's birthday															
Mom and dad anniversary															
Presentation to Unilever															
Rizal Zulfikar's birthday															
Meeting															
Polban's anniversary															

No.	UI - 21
Nama	Template
	UI - 21.3 : UI template untuk platform iOS
	UI - 21.4 : UI template untuk platform Java/Symbian

Keterangan : Keterkaitan antar komponen UI pada halaman *template* dan keterkaitan halaman tersebut dengan halaman lainnya pada aplikasi digambarkan pada gambar 50.



Statechart : UI halaman template

Tabel 152. Rancangan UI halaman Template (*New template*)

No.	UI - 22
Nama	<i>Template (New template)</i>
Deskripsi	Halaman untuk membuat template baru dengan menampilkan <i>form new template</i> .
Gambar	<p>UI – 22.1 : UI new template untuk platform Android</p>

UI - 22.2 : UI new template untuk platform Blackberry

Pada UI ini, perbedaannya yaitu seluruh komponen *form* akan terpotong dan harus di-*scroll* untuk melihat komponen *form* lainnya, dikarenakan pada umumnya layar *device* Blackberry tidak sebesar layar *device* Android, iOS dan Java.

--	--

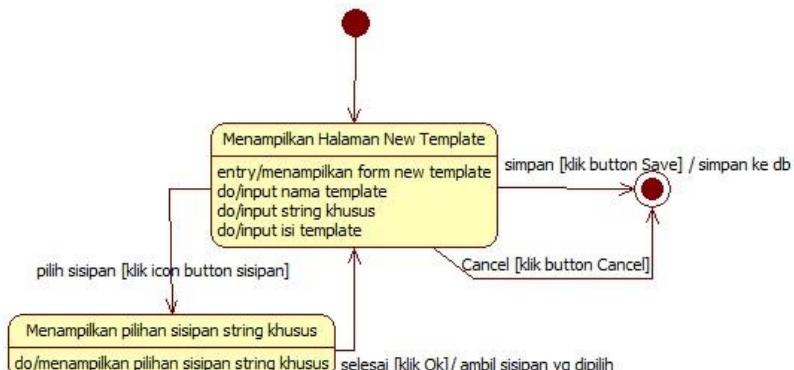
Tabel 153. Rancangan UI halaman Template (*New template*) (Lanjutan)

No.	UI - 22
Nama	<i>Template (New template)</i>

UI - 22.3 : UI new template untuk platform iOS
UI untuk device iOS ini sebetulnya sama seperti Android, perbedaannya hanya pada letak button cancel-nya yang berada di bagian atas, sementara pada Android berada di bawah (disamping button save).

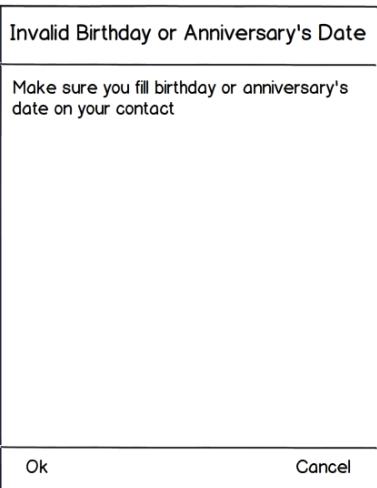
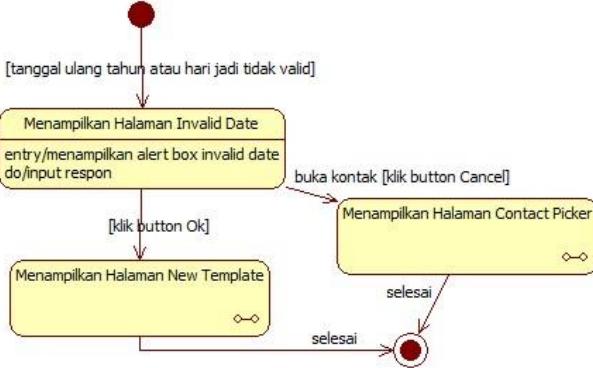
UI - 22.4 : UI new template untuk platform Java/Symbian	

Keterangan : Keterkaitan antar komponen UI pada form *new template* dan keterkaitan halaman tersebut dengan halaman lainnya pada aplikasi digambarkan pada gambar 51.

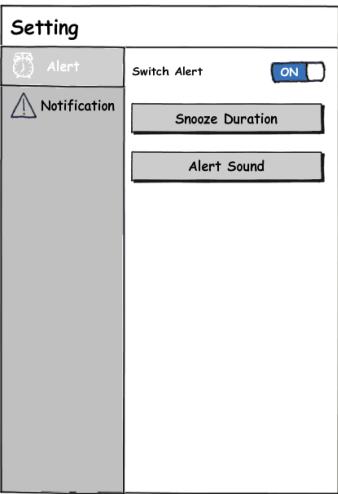
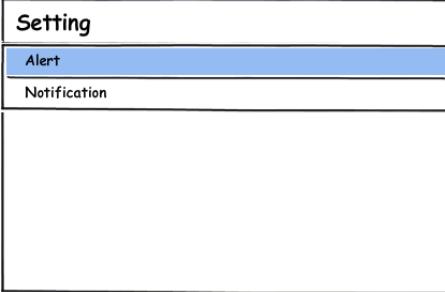
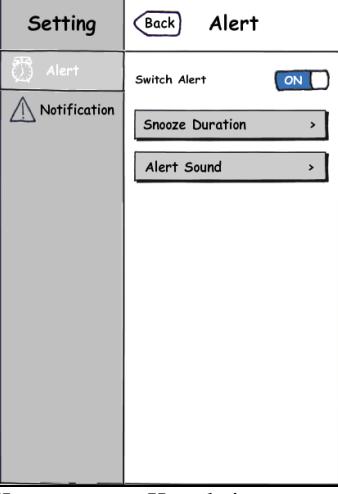
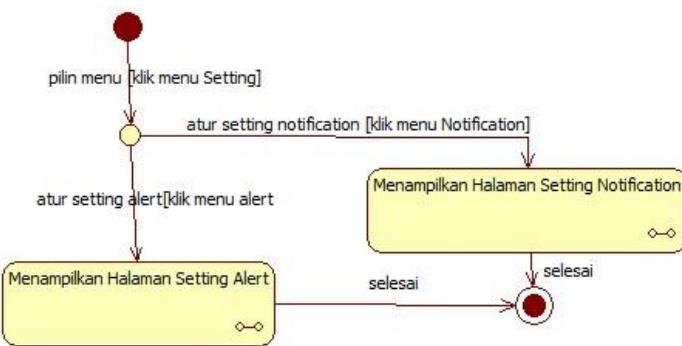


Gambar 52 . Statechart : UI halaman New Template

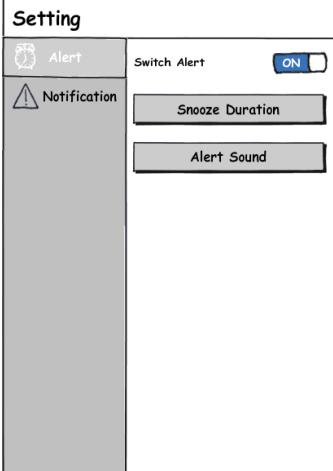
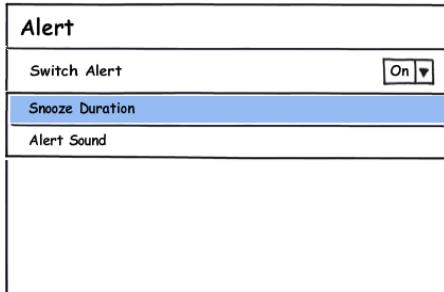
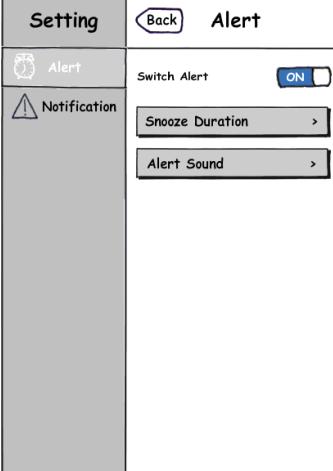
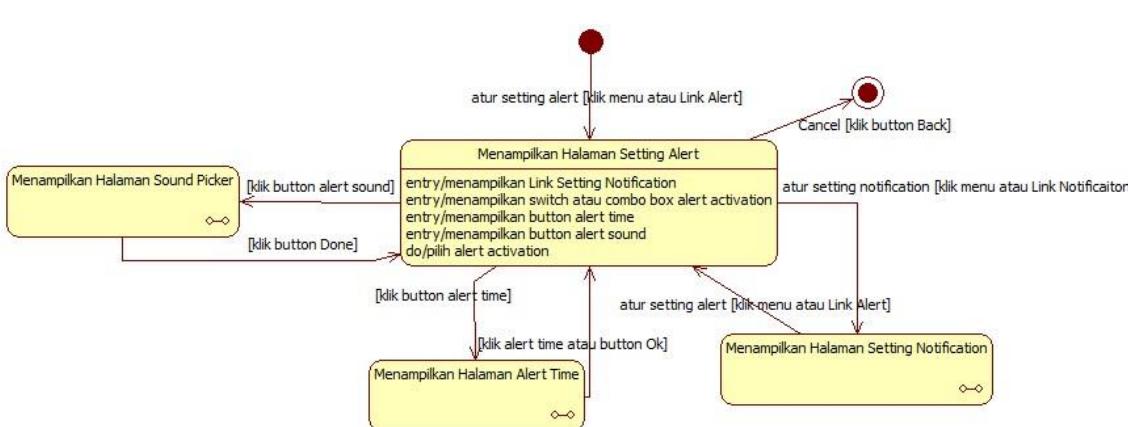
Tabel 154. Rancangan UI halaman *Template (Invalid Date)*

No.	UI - 23
Nama	<i>Template (Invalid Date)</i>
Deskripsi	Halaman untuk menampilkan pemberitahuan bahwa data tanggal yang dipilih <i>user</i> dari kontak tidak ada.
Gambar	
	 <p>UI - 23.1 : UI Set Data Count untuk platform Android, Blackberry dan iOS</p>  <p>UI - 23.2 : UI Set Data Count untuk platform Java/Symbian</p>
<p>Keterangan : Keterkaitan antar komponen UI pada halaman <i>invalid date</i> dan keterkaitan halaman tersebut dengan halaman lainnya pada aplikasi digambarkan pada gambar 52.</p>  <pre> graph TD Start(()) -- "[tanggal ulang tahun atau hari jadi tidak valid]" --> Invalid[Menampilkan Halaman Invalid Date
entry/menampilkan alert box invalid date do/input respon] Invalid -- "[klik button Ok]" --> NewTemplate[Menampilkan Halaman New Template] Invalid -- "buka kontak [klik button Cancel]" --> ContactPicker[Menampilkan Halaman Contact Picker] NewTemplate -- "selesai" --> End(()) ContactPicker -- "selesai" --> End </pre> <p>Gambar 53 . Statechart : UI halaman <i>invalid date</i></p>	

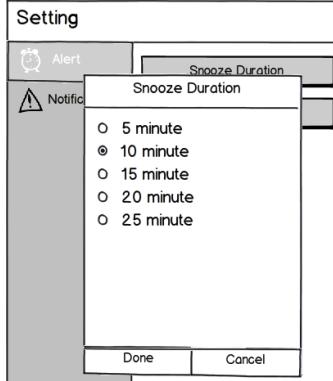
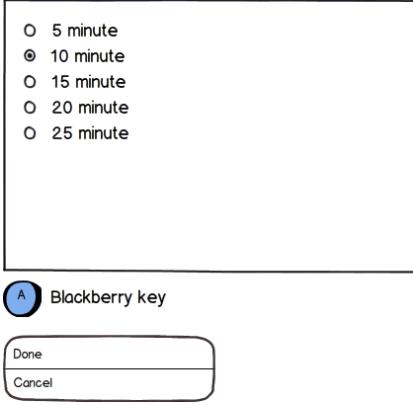
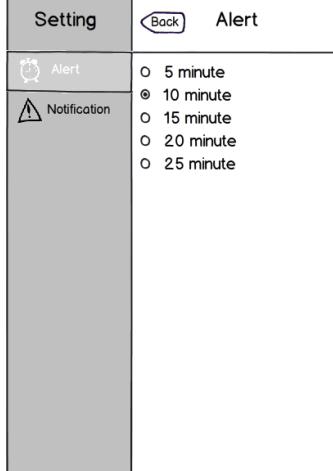
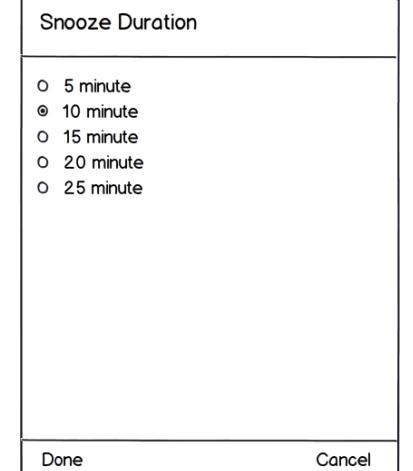
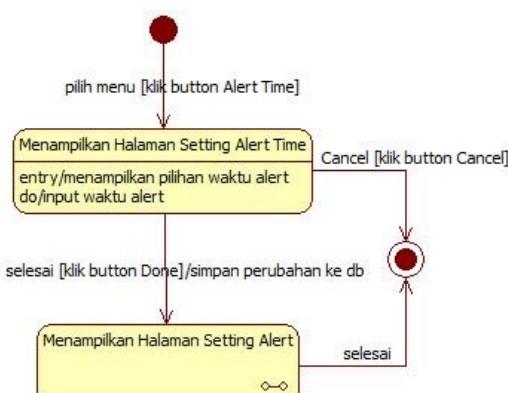
Tabel 155. Rancangan UI halaman *setting*

No.	UI - 24		
Nama	Setting		
Deskripsi	Halaman untuk melakukan pengaturan alert dan notification yang akan ditampilkan oleh aplikasi.		
Gambar	 <p>UI - 24.1 : UI <i>setting</i> untuk platform Android</p>		<p>UI - 24.2 : UI <i>setting</i> untuk platform Blackberry</p>
	 <p>UI - 24.3 : UI <i>setting</i> untuk platform iOS</p>		<p>UI - 24.4 : UI <i>setting</i> untuk platform Java/Symbian</p>
Keterangan :	Keterkaitan antar komponen UI pada halaman <i>date and time picker</i> dan keterkaitan halaman tersebut dengan halaman lainnya pada aplikasi digambarkan pada gambar 53.		
			
	<p>Gambar 54 . Statechart : UI halaman <i>setting</i></p>		

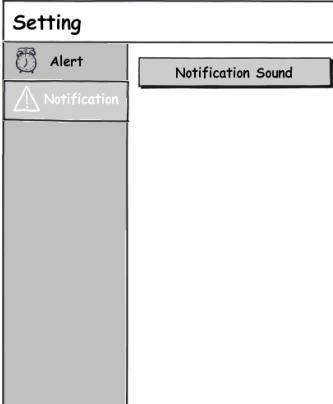
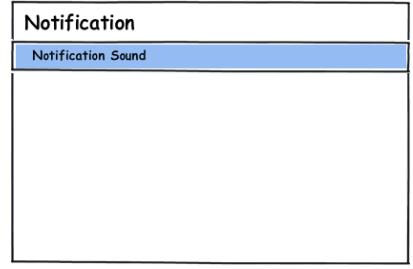
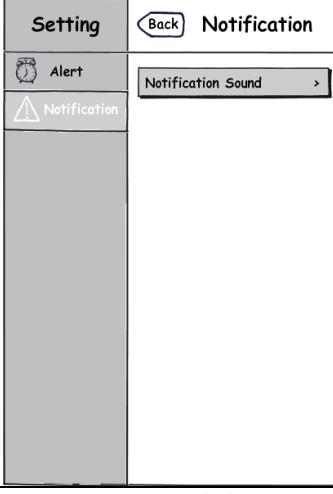
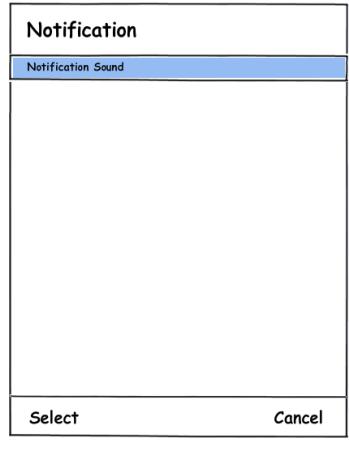
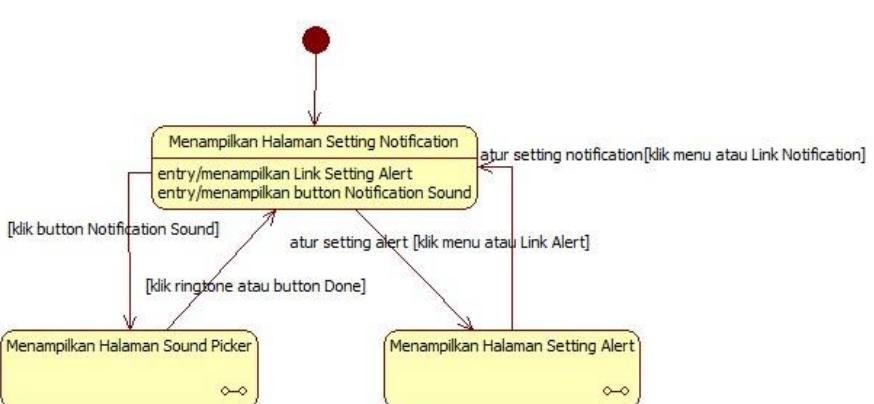
Tabel 156. Rancangan UI halaman *setting(alert)*

No.	UI - 25		
Nama	Setting(alert)		
Deskripsi	Halaman untuk melakukan pengaturan alert yang akan ditampilkan oleh aplikasi.		
Gambar	 <p>UI - 25.1 : UI setting (alert) untuk platform Android</p>	 <p>UI - 25.2 : UI setting (alert) untuk platform Blackberry</p>	
	 <p>UI - 25.3 : UI setting (alert) untuk platform iOS</p>	 <p>UI - 25.4 : UI setting (alert) untuk platform Java/Symbian</p>	
	<p>Keterangan : Keterkaitan antar komponen UI pada halaman <i>setting alert</i> dan keterkaitan halaman tersebut dengan halaman lainnya pada aplikasi digambarkan pada gambar 54.</p> 		
	<p>Gambar 55 . Statechart : UI halaman Setting Alert</p>		

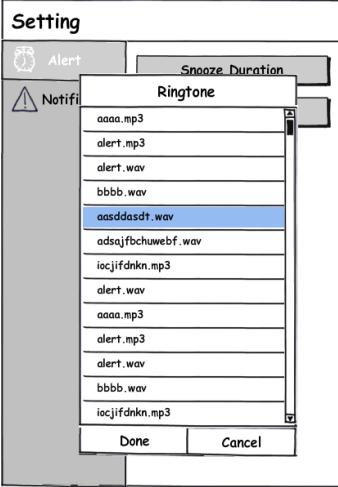
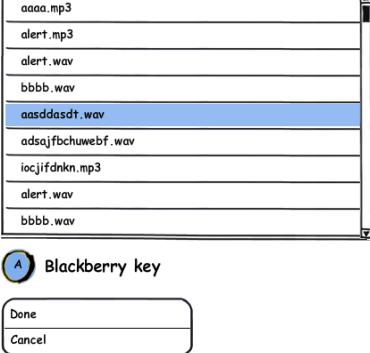
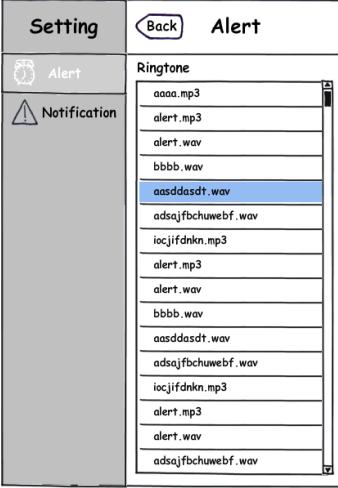
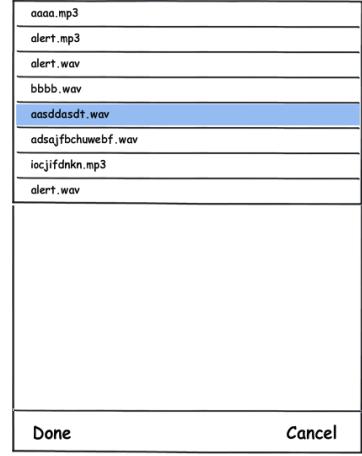
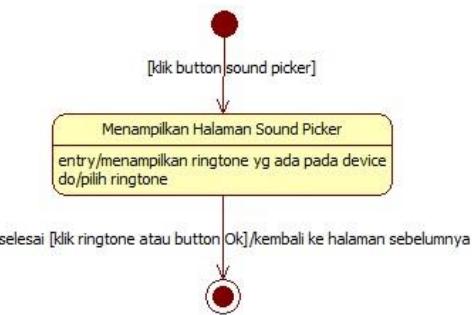
Tabel 157. Rancangan UI halaman *setting(Alert Time)*

No.	UI - 26			
Nama	Setting(Alert Time)			
Deskripsi	Halaman untuk melakukan pengaturan waktu alert.			
Gambar		UI - 26.1 : UI <i>setting (Snooze duration)</i> untuk platform Android	 A Blackberry key	UI - 26.2 : UI <i>setting (Snooze duration)</i> untuk platform BlackBerry
		UI - 26.3 : UI <i>setting (Snooze duration)</i> untuk platform iOS		UI - 26.4 : UI <i>setting (Snooze duration)</i> untuk platform Java/Symbian
Keterangan :	Keterkaitan antar komponen UI pada halaman <i>setting alert time</i> dan keterkaitan halaman tersebut dengan halaman lainnya pada aplikasi digambarkan pada gambar 55.			
				
	Gambar 56 . Statechart : UI halaman <i>setting alert time</i>			

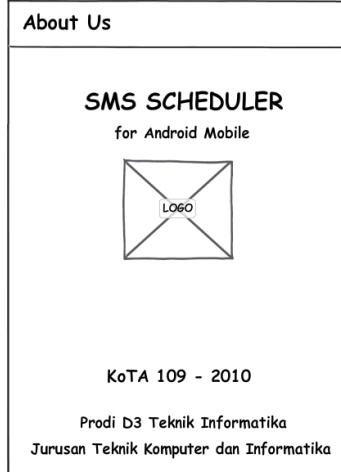
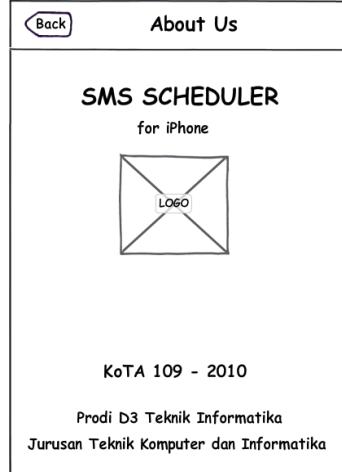
Tabel 158. Rancangan UI halaman *setting(notification)*

No.	UI - 27			
Nama	Setting(notification)			
Deskripsi	Halaman untuk melakukan pengaturan notification yang akan ditampilkan oleh aplikasi.			
Gambar	 <p>UI - 27.1 : UI setting (notification) untuk platform Android</p>	 <p>UI - 27.2 : UI setting (notification) untuk platform BlackBerry</p>		
	 <p>UI - 27.3 : UI setting (notification) untuk platform iOS</p>	 <p>UI - 27.4 : UI setting (notification) untuk platform Java/Symbian</p>		
Keterangan :	Keterkaitan antar komponen UI pada halaman <i>setting notification</i> dan keterkaitan halaman tersebut dengan halaman lainnya pada aplikasi digambarkan pada gambar 56.			
	 <pre> graph TD Start(()) --> Main[Menampilkan Halaman Setting Notification] Main -- "[klik button Notification Sound]" --> SoundPicker[Menampilkan Halaman Sound Picker] Main -- "atur setting notification[klik menu atau Link Notification]" --> AlertSetting[Menampilkan Halaman Setting Alert] SoundPicker -- "[klik ringtone atau button Done]" --> Main AlertSetting -- "atur setting alert [klik menu atau Link Alert]" --> Main </pre>			
	<p>Gambar 57 . Statechart : UI halaman <i>setting notification</i></p>			

Tabel 159. Rancangan UI halaman *setting(sound picker)*

No.	UI - 28		
Nama	Setting(<i>sound picker</i>)		
Deskripsi	Halaman untuk memilih nada dering alert atau notification.		
Gambar	 <p>UI - 28.1 : UI setting (<i>sound picker</i>) untuk platform Android</p>	 <p>UI - 28.2 : UI setting (<i>sound picker</i>) untuk platform Blackberry</p>	
	 <p>UI - 28.3 : UI setting (<i>sound picker</i>) untuk platform iOS</p>	 <p>UI - 28.4 : UI setting (<i>sound picker</i>) untuk platform Java/Symbian</p>	
Keterangan :	Keterkaitan antar komponen UI pada halaman <i>sound picker</i> dan keterkaitan halaman tersebut dengan halaman lainnya pada aplikasi digambarkan pada gambar 57.		
	 <pre> graph TD Start(()) -- "[klik button sound picker]" --> SoundPicker[Menampilkan Halaman Sound Picker entry/menampilkan ringtone yg ada pada device do/pilih ringtone] SoundPicker -- "selesai [klik ringtone atau button Ok]/kembali ke halaman sebelumnya" --> End(()) </pre>		
	<p>Gambar 58 . Statechart : UI halaman <i>sound picker</i></p>		

Tabel 160. Rancangan UI halaman *about us*

No.	UI - 29
Nama	<i>About us</i>
Deskripsi	Halaman untuk menampilkan informasi pembuat aplikasi.
Gambar	 <p>UI - 29.1 : UI <i>about us</i> untuk <i>platform</i> Android, Blackberry dan Java, namun dengan label untuk masing-masing <i>platform</i></p>  <p>UI - 29.3 : UI <i>about us</i> untuk <i>platform</i> iOS</p>

Keterangan : Keterkaitan antar komponen UI pada halaman *about us* dan keterkaitan halaman tersebut dengan halaman lainnya pada aplikasi digambarkan pada gambar 58.

Gambar 59 . Statechart : UI halaman *about us*

BAB V

Implementasi

Pada tahap ini akan dilakukan implementasi aplikasi SMS *Scheduler* berdasarkan perancangan yang telah dibuat pada tahap sebelumnya. Diharapkan dari implementasi ini didapatkan aplikasi yang sesuai dengan tujuan awal pembuatan aplikasi dan rancangan yang telah dibuat. Untuk itu, pada tahap ini dilakukan implementasi aplikasi pada *platform* Android terlebih dahulu berdasarkan perancangan yang sebelumnya telah dibuat, dengan fitur untuk membuat *schedule*, mengirimkan pesan berdasarkan *schedule* yang dibuat, menampilkan notifikasi dan *template*.

5.1 Batasan Implementasi

Implementasi untuk pembuatan aplikasi SMS *Scheduler* ini hanya dilakukan pada *platform* Android, dengan fitur membuat *schedule* baru, mengirimkan pesan sesuai *schedule* yang dibuat, menampilkan notifikasi *schedule*, dan *template*.

5.2 Implementasi Data

Dari perancangan data yang telah dibuat pada subbab 4.2, terdapat beberapa kebutuhan data yang telah terimplementasi. Tabel 161 mendefinisikan kebutuhan data yang telah terimplementasi.

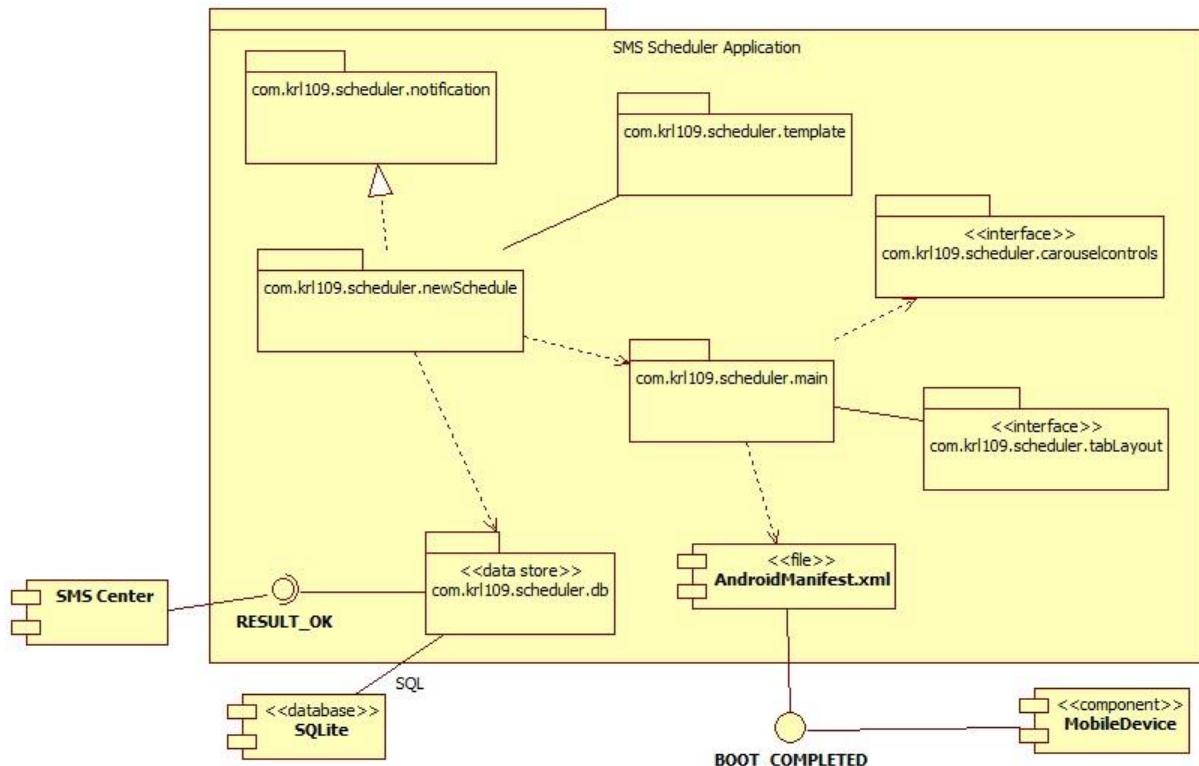
Tabel 161. Status implementasi data

Nama Tabel	Deksripsi	Status Implementasi
<i>Message</i>	Tabel ini digunakan untuk menyimpan data jadwal yang akan dikerjakan oleh aplikasi.	✓
<i>Recipient</i>	Tabel ini digunakan untuk menyimpan data penerima pesan yang digunakan.	✓
<i>Nomor</i>	Tabel ini digunakan untuk menyimpan nomor-nomor <i>recipient</i> (penerima pesan)	✓
<i>Time</i>	Tabel ini digunakan untuk menyimpan setiap waktu pengiriman yang digunakan.	✓
<i>Typical_Message</i>	Tabel ini digunakan untuk menyimpan pesan yang bersifat dinamis, yaitu pesan yang mengandung <i>defined character</i> .	✓
<i>Normal_Message</i>	Tabel ini digunakan untuk menyimpan pesan yang bersifat statis, yaitu pesan yang tidak menggunakan <i>defined</i>	✓

Nama Tabel	Deksripsi	Status Implementasi
	<i>character.</i>	
<i>Defined_Character</i>	Tabel ini digunakan untuk menyimpan data tanggal, bulan, tahun, atau usia, baik ulang tahun maupun <i>anniversary</i> .	√
<i>Template</i>	Tabel ini berfungsi untuk menyimpan data <i>template</i> yang sudah dibuat dan akan digunakan pada <i>schedule</i> .	√
<i>Category_Template</i>	Tabel ini digunakan untuk menyimpan data kategori-kategori <i>template</i> .	√
<i>Setting</i>	Tabel ini digunakan untuk menyimpan pengaturan aplikasi, seperti pengaturan nada dering.	x

5.3 Implementasi Aplikasi

Berikut ini merupakan *component diagram* untuk menggambarkan keterkaitan antar komponen aplikasi SMS Scheduler.



Gambar 60 . *Component diagram* aplikasi SMS Scheduler

Untuk fungsi-fungsi yang terkait dengan *package-package* yang ada pada *component diagram* diatas dapat dilihat pada lampiran C. Penjelasan lebih rinci mengenai komponen-komponen pada Gambar 61 dapat dilihat pada Tabel 162.

Tabel 162. Penjelasan deployment diagram

No	Nama Komponen	Penjelasan
1.	<i>package com.krl109.scheduler.db</i>	Merupakan paket yang berisi informasi logika untuk menangani penyimpanan dan pengolahan data aplikasi dan pengiriman pesan
2.	<i>package com.krl109.scheduler.notification</i>	Merupakan paket yang berisi informasi logika yang menangani notifikasi.
3.	<i>package com.krl109.scheduler.template</i>	Merupakan paket yang berisi informasi logika untuk menangani pengolahan template
4.	<i>package com.krl109.scheduler.newSchedule</i>	Merupakan paket yang berisi informasi logika untuk menangani pembuatan <i>schedule baru</i> , pengolahan data pada <i>contact</i> , dan penanganan frekuensi pengiriman pesan.
5.	<i>package com.krl109.scheduler.carouselcontrols</i>	Merupakan paket yang berisi informasi logika untuk menangani interface aplikasi awal muka halaman.
6.	<i>package com.krl109.scheduler.main</i>	Merupakan paket yang digunakan untuk mengatasi permasalahan jika device mengalami reboot dan inisialisasi awal aplikasi.
7.	<i>component MobileDevice</i>	Merupakan mobile device
8.	<i>component SMS Center</i>	Merupakan komponen pendukung yang digunakan untuk melakukan pengiriman sms kepada recipient
9.	<i>package com.krl109.scheduler.tabLayout</i>	Merupakan paket yang berisi informasi logika untuk menangani interface schedule pada aplikasi.
10.	<i>component AndroidManifest.xml</i>	Merupakan komponen yang digunakan untuk registrasi informasi android application ke operating system pada mobile device.
11.	<i>component SQLite</i>	Merupakan komponen pendukung aplikasi yang menghubungkan logika informasi aplikasi dengan database mobile device

Sebelumnya pada Bab Perancangan telah disebutkan bahwa terdapat *class* dan *method* dari *device* yang dimanfaatkan atau digunakan dalam pembuatan aplikasi SMS *Scheduler* ini. *Detail* dari *class* dan *method* dari *device* yang digunakan tersebut dapat dilihat pada tabel 163.

Tabel 163. *Class* dan *method* Android yang digunakan dalam pembuatan aplikasi

Tujuan Penggunaan	Nama Class	Deskripsi	Nama Method
Untuk menjalankan aplikasi di satu waktu tertentu	AlarmManager	<i>Class</i> menyediakan akses ke <i>system services</i> yang yang akses ke <i>alarm</i>	set(int type, long triggerAtMillis, PendingIntent operation)

Tujuan Penggunaan	Nama Class	Deskripsi	Nama Method
Meng-handle antrian sms	ArrayList	<i>Class</i> yang merupakan implementasi dari <i>list</i> , dengan nilai kembalian sebagai <i>array</i>	get(int index) remove(int index) isEmpty() add(E object) size()
Untuk memanggil <i>class</i> atau <i>activity</i> lain, baik di awal saat aplikasi dijalankan maupun pada saat aplikasi sedang berjalan	Intent	<i>Class</i> yang digunakan untuk berkomunikasi dengan <i>class</i> lain	putExtra(string name, long value) putExtra(string name, string value) startActivity(Intent intent) getIntent() getLongExtra(string name) getStringExtra(string name)
Untuk memanggil <i>class</i> atau <i>activity</i> lain disaat yg diinginkan	PendingIntent	<i>Class</i> yang digunakan untuk melakukan suatu operasi pada waktu tertentu	getActivity(Context context, int requestCode, Intent intent, int flags) getBroadcast(Context context, int requestCode, Intent intent, int flags)
Menampung data dari <i>database</i>	Cursor	<i>Class</i> yang menyediakan akses untuk <i>read</i> dan <i>write</i> pada <i>database</i>	moveToNext() moveToFirst() getColumnIndex(string status)
Mengakses komponen <i>user interface</i> untuk melakukan <i>input</i> data atau sebagai <i>trigger</i> dari suatu aksi.	Button	<i>Class</i> yang merepresentasikan <i>widget push-button</i> yang dapat diklik atau ditekan oleh user untuk menjalankan suatu aksi	setOnClickListener(View.OnClickListener I) registerForContextMenu(View v)
	EditText	<i>Class</i> yang menangani <i>text view</i> yang dapat diedit atau diubah isinya (<i>editable</i>)	getText() setText(CharSequence text)
	Spinner	<i>Class</i> yang menampilkan beberapa pilihan dalam satu waktu dan memungkinkan pengguna untuk memilih pilihan tersebut	setAdapter(SpinnerAdapter adapter) setSelection(int position) getSelectedItemPosition()
	RadioButton	<i>Class</i> yang merupakan	getText() getCheckedRadioButtonId(int id)
	ToggleButton	Menampilkan kondisi check atau uncheck dengan menggunakan button, dan secara default disertai	getTextOn() OnClickListener() getText()

Tujuan Penggunaan	Nama Class	Deskripsi	Nama Method
		dengan teks ON atau OFF	
Mendapatkan tanggal dan waktu dari sistem	Calendar	<i>Class</i> yang digunakan untuk <i>mengkonversi date</i> menjadi <i>millisecond</i>	getInstance() set(int foeld, int value) getTimeInMillis()
Menampilkan informasi untuk user yang berbentuk pesan singkat	Toast	<i>Class</i> yang membantu untuk membuat dan menampilkan <i>toast</i>	makeText(Context context, CharSequence text, int duration)
Mengakses komponen <i>user interface</i> untuk melakukan <i>set</i> tanggal dan waktu.	DatePicker TimePicker	Class yang merupakan widget untuk memilih tanggal. Class yang merupakan widget untuk memilih waktu, baik dalam format waktu 24 jam, maupun 12 jam (AM/PM).	
Melakukan pengiriman sms	SmsManager	Class yang memanage operasi-operasi seperti pengiriman data, text dan pdu SMS message.	sendTextMessage(string destinationAddress, string scAddress, string text, PendingIntent sentIntent, PendingIntent deliveryIntent)
Mengakses nada dering yang ada pada <i>device</i>	RingtoneManager	Class yang menyediakan akses untuk ringtone, notification dan tipe suara/audio lainnya.	getDefaultUri(int type) getRingtoneUri(int position)
Mengakses <i>layout view</i> (dalam bentuk <i>list, tab, dll</i>)	ListView TabHost TabSpec	<i>Class</i> untuk menangani view yang menampilkan item-itemnya dalam bentuk <i>list</i> secara <i>vertical</i> <i>Class</i> untuk menangani view dalam bentuk <i>tabbed window</i> . <i>Class</i> yang menangani indikator tab dan konten dari tab tersebut	
			getTabHost() newTabSpec() addTab(TabSpec tabSpec) setIndicator(CharSequence label, Drawable icon) setContent(Intent intent)

Untuk aplikasi yang sudah terimplementasi, dapat dilihat pada *screenshot* yang ada pada Lampiran E.

BAB VI

Pengujian

Pada tahap ini akan dilakukan pengujian terhadap aplikasi yang telah berhasil diimplementasi. Diharapkan dari kegiatan ini ditemukan kesalahan-kesalahan atau *bugs* dari aplikasi yang telah terimplementasi dan diketahui apakah aplikasi yang dibuat telah memenuhi *requirement* yang telah didefinisikan di tahap analisis. Untuk itu, dilakukan pengujian dengan metode *functional testing* berdasarkan *requirement functional* yang sudah terdefinisi di tahap analisis.

6.1 Perencanaan Pengujian

Berikut ini merupakan *test case* untuk melakukan pengujian pada aplikasi SMS *Scheduler* yang sudah diimplementasi dan dituliskan pada tabel 164.

Tabel 164. *Test Case*

<i>Use case</i>	<i>Fungsi yang Diuji</i>	<i>Test Case (Input)</i>	<i>Output yang Diharapkan</i>	<i>Status Pengujian</i>
Add schedule	Memvalidasi field <i>recipient</i>	+628997937561	Sistem menerima <i>input recipient</i>	√
		+6281320780777	nomor	√
		+62883207778743		√
		abcd@	Sistem menampilkan pesan <i>error</i>	√
		+;6283		√
		081320745		√
		0813207456		√
		1234567		√
		1234567890		√
		11234567890		√
		+6281abc7807		√
		081+62678901		√
		000000000000		x
		111111111111		x
		++6281320780777		x
	Memvalidasi waktu pengiriman	081320;081320	Sistem menerima <i>input</i> waktu pengiriman	√
		081320780777;081320780777		√
		Semua <i>alphabet</i>		√
		Semua <i>alphanumeric</i>		√

Use case	Fungsi yang Diuji	Test Case (Input)	Output yang Diharapkan	Status Pengujian
<i>Add schedule</i>	Memvalidasi waktu pengiriman	Memasukkan waktu berulang kali	Field waktu terganti sesuai dengan yang diinputkan	x
	Memvalidasi frekuensi pengiriman	Memilih “once”	Field remaining disabled dan terisi 1 otomatis	√
		Memilih selain “once”	Field remaining dapat diisi	√
	Memvalidasi field remaining	Semua <i>alphabet</i>	Sistem menampilkan pesan error	√
		Semua <i>alphanumerik</i>		√
		-15		√
		1,8		√
		2,00		√
		0		x
	Memvalidasi field content khusus typical	%%AGE	Diproses menjadi pesan normal	x
		%AGE%%		x
		AGE%%		x
		%AGE		x
		%%AG%%		x
		%%aGE%%		x
<i>menyimpan ke database</i>	menyimpan ke database	Field recipient salah	Data tidak masuk ke database	√
		Waktu telah lampau		x
		Field remaining salah		x
		Field remaining diisi 0	Terisi 1 record di table delivery_time	x
		Field recipient diisi 15	Terisi 15 record di table delivery_time	√
		Terdapat %%AGE%%	Tersimpan di table typical_message	√
		Tidak terdapat karakter khusus	Tersimpan di table normal_message	√
		Terdapat 1 recipient	Tersimpan 1 record di table recipient	√
		Terdapat 10 recipient	Tersimpan 10 record di table recipient	√
<i>Add Template</i>	Menyimpan ke database	Terdapat field yang belum terisi	Data tidak masuk ke database	√
		Nama template beru sama dengan yang sebelumnya	Menampilkan pesan error dan data tidak masuk ke database	x
		Terdapat string khusus pada content template	Tersimpan di table typical_message	√
		Tidak terdapat string khusus	Tersimpan di table normal message	√
		Memilih “anniversary”	Category tersimpan sebagai anniversary	√
		Memilih “birthday”	Category tersimpan sebagai birthday	√

<i>Use case</i>	<i>Fungsi yang Diuji</i>	<i>Test Case (Input)</i>	<i>Output yang Diharapkan</i>	<i>Status Pengujian</i>
<i>Delete template</i>	Menghapus <i>template</i> dari database	Memilih “delete”	Data <i>template</i> terhapus dari database	√
<i>Show sending notification</i>	Menampilkan notifikasi	Menerima <i>trigger</i> mengirimkan pesan	Menampilkan notifikasi	√
<i>Show failed notification</i>	Menampilkan notifikasi	<i>Trigger gagal (tidak ada pulsa)</i>	Menampilkan notifikasi	√
		<i>Trigger gagal (tidak ada sinyal)</i>		√
		<i>Trigger gagal (gangguan dari provider)</i>		√
<i>Show sent notification</i>	Menampilkan notifikasi	<i>Trigger</i> pesan terkirim	Menampilkan notifikasi pesan sudah terkirim	√
<i>Enroll delivery time</i>	Mendaftarkan waktu	Terdapat waktu yang sama	<i>Update</i> dengan yang terbaru	√
<i>Send message from schedule</i>	Mengirim pesan	Terdapat jadwal dengan waktu yang sama	Mengantrikan pesan	√
		Pesan sebelumnya belum selesai terkirim datang pesan baru		√

Keterangan : √ : berhasil ✗ : gagal

BAB VII

Penutup

Pada bab ini memaparkan kesimpulan dan saran dari pelaksanaan Tugas Akhir yang didapatkan dari seluruh proses yang terdapat pada pembuatan Tugas Akhir ini.

7.1 Kesimpulan

Berdasarkan hasil analisis, perancangan dan implementasi yang telah dilakukan, dapat diambil beberapa kesimpulan sebagai berikut :

1. Aplikasi SMS *Scheduler* yang dibuat dapat melakukan pengiriman pesan berdasarkan *schedule* yang sebelumnya sudah dibuat.
2. Aplikasi SMS *Scheduler* yang dibuat menyediakan fitur notifikasi sehingga dapat memberikan informasi pada *user* bahwa pesan sudah berhasil dikirim atau gagal dikirim.
3. Aplikasi SMS *Scheduler* yang dibuat dapat melakukan pengambilan nomor *recipient* dari kontak *device*, sehingga *user* tidak perlu menginputkan nomor *recipient* secara manual.
4. Aplikasi SMS *Scheduler* yang dibuat dapat melakukan pengambilan nomor *recipient* dari kontak *device*, sehingga dapat membantu *user* ketika melakukan pemilihan lebih dari satu *recipient*.
5. Aplikasi SMS *Scheduler* yang dibuat menyediakan *template*, baik *template* biasa maupun *template* khusus untuk memberikan ucapan selamat ulang tahun atau selamat hari jadi.
6. Pemilihan model *incremental* sebagai model pengembangan aplikasi dimaksudkan agar dalam waktu dekat sudah ada produk yang dihasilkan, minimal menghasilkan aplikasi dengan fitur utama. Hal tersebut dikarenakan waktu penggerjaan aplikasi yang terbatas pada jadwal yang telah ditentukan di awal penggerjaan *project* dan adanya tuntutan untuk dapat memperlihatkan progress pekerjaan dalam waktu dekat, yaitu setiap 2 bulan sekali.

Namun, pada pelaksanaan tugas akhir ini masih terdapat beberapa permasalahan yang belum terimplementasi, yaitu fitur untuk melakukan perubahan *schedule*, memberikan *alert* sebelum melakukan pengiriman, menampilkan pesan berdasarkan kategorinya dan menyediakan *setting* aplikasi.

7.2 Saran

Untuk melengkapi dan menyempurnakan aplikasi SMS *Scheduler* yang telah dibuat, hal-hal yang perlu dilakukan pada pengembangan selanjutnya, yaitu :

1. Menyelesaikan fitur-fitur pendukung aplikasi yang belum terimplementasi yaitu, fitur untuk melakukan perubahan *schedule*, menampilkan *alert* sebelum melakukan pengiriman, menampilkan pesan berdasarkan kategorinya (*scheduled*, *sent*, *failed* dan *paused*) dan *setting* aplikasi.
2. Menyelesaikan penggerjaan aplikasi sesuai dengan *library* pendukung untuk *platform* lain.

DAFTAR PUSTAKA

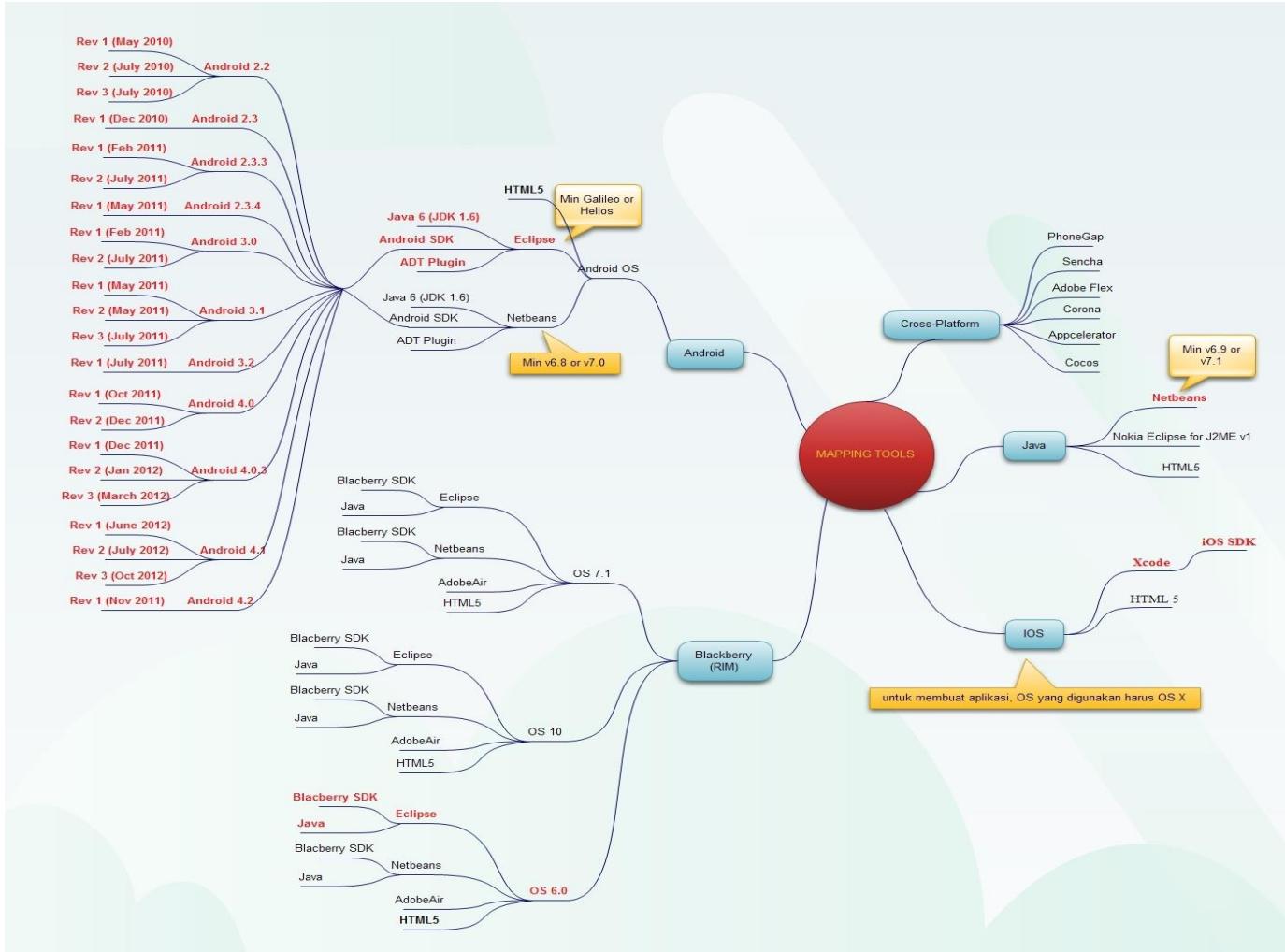
- Ambler, Scott W. *The Elements of UML 2.0 Style*. United States of America : Cambridge University Press, 2005.
- Alleman, Glen B. (2005). What is the Purpose of a Schedule ?. Diakses pada tanggal 11 Juni 2013 dari http://herdingcats.typepad.com/my_weblog/2005/11/what_is_the_pur.html.
- Budgen, David. *Software Design - Second edition*. England : Addison Wesley, 2003.
- Charland, Ade. (2011). *Mobile Application Development : Web vs Native*. Diakses pada tanggal 24 Desember 2012 dari <http://www.creativebrain.web.id/media.php?action=readnews&id=315&title=Mobile%20Application%20Development:%20Web%20vs.%20Native>.
- Eclipse. (n.d.). About *the Eclipse Foundation* : What is Eclipse and the Eclipse Foundation. Diakses pada tanggal 6 April 2013 dari <http://www.eclipse.org/org/#about>.
- Eclipse. (n.d.). Older Version of Eclipse. Diakses pada tanggal 28 November 2013 dari http://wiki.eclipse.org/Older_Versions_Of_Eclipse.
- Inc, Apple. (n.d.) *Develop Apps for iOS*. Diakses pada tanggal 6 April 2013 dari <https://developer.apple.com/technologies/ios/>.
- Inc, Apple. (n.d.). *About Objective-C*. Diakses pada tanggal 6 April 2013 dari <http://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html>.
- Inc, Apple. (n.d.). *iOS : The world's most advanced mobile operating system*. Diakses pada tanggal 6 April 2013 dari <http://www.apple.com/ios/what-is/>.
- Inc, Apple. (n.d.). *iOS Developer Program*. Diakses pada tanggal 6 April 2013 dari <https://developer.apple.com/programs/ios/develop.html>.
- Inc, Apple. (n.d.). Xcode. Diakses pada tanggal 6 April 2013 dari <https://itunes.apple.com/us/app/xcode/id497799835>.
- Latu, Arief. (2010). Apa itu SMS dan cara kerjanya. Diakses pada tanggal 7 Maret 2013 dari <http://blog.arieflatu.net/2010/10/apa-itu-sms-dan-cara-kerjanya.html/>.

- Lord. (2011). Sejarah Netbeans. Diakses pada tanggal 2 Januari 2012 dari <http://lordarea.forumpro.in/t30-sejarah-netbeans>.
- Ltd, Aims MIGITAL Technovations Pvt. (2013). SMS Scheduler Lite 1.0. Diakses pada tanggal 31 Mei 2013 dari http://www.handster.com/sms_scheduler_lite.html.
- N.N. (n.d.). Java. Diakses pada tanggal 6 April 2013 dari <http://www.webopedia.com/TERM/J/Java.html>.
- N.N. (n.d.). Platforms. Diakses pada tanggal 28 November 2012 dari <http://developer.android.com/tools/revisions/platforms.html>.
- Netbeans. (2012). Selamat datang di Netbenas dan www.netbeans.org. Diakses pada tanggal 2 Januari 2013 dari http://netbeans.org/index_id.html.
- Noviyanto. (n.d.). Pemrograman Berorientasi Objek (Pertemuan Ke-1) : Pengenalan Bahasa Pemrograman Java. Diakses pada tanggal 6 April 2013 dari http://fanny_tandi.staff.gunadarma.ac.id/Downloads/files/20647/1_Pengenalan_Java.pdf.
- Ojo, Adegboyega; Estevez, Elsa. Object-Oriented Analysis and Design with UML. UNU-IIST
- Oracle. (n.d.). About Java ME Technology. Diakses pada tanggal 5 April 2013 dari <http://www.oracle.com/technetwork/java/javame/about-java-me-395899.html>.
- Play, Google. (2011). SMS Scheduler. Diakses pada tanggal 21 Desember 2012 dari https://play.google.com/store/apps/details?id=com.shaperstudio.projects.smsscheduler&feature=search_result.
- Pressman, Roger S. *Software engineering : a practitioner's approach – 5th edition*. New York : McGraw-Hill, 2001.
- Rumbaugh J, Jacobson I, Booch G. The Inified Modeling Language Reference Manual. United States of America : Addison Wesley, 1999.
- Stieben, Danny. (2013). Send It Later : 4 Useful SMS Scheduling Apps To Text At Just The Right Time [Android]. Diakses pada tanggal 31 Mei 2013 dari <http://www.makeuseof.com/tag/useful-sms-scheduling-apps-for-android/>.
- SymbianFreak. (2008). SMS Scheduler : Schedule SMS to be sent at chosen time !. Diakses pada tanggal 31 Mei 2013 dari http://www.symbian-freak.com/downloads/freeware/cat_s60_3rd/descriptions/messaging/SMSScheduler_send_SMS_at_chosen_time.htm.

Lampiran A

Software Requirement Specification

Lampiran B
Mapping Tools



Keterangan :

Untuk tulisan dengan warna merah, artinya adalah *tools* yang digunakan dalam pembuatan aplikasi SMS *Scheduler* ini.

Lampiran C
Status Implementasi *Requirement*

Nomor <i>Requirement</i>	Deksripsi	Status Implementasi
[REQ F - 01]	Aplikasi menyediakan fitur untuk membuat <i>schedule</i> pengiriman pesan.	√
[REQ CRT - 01]	Aplikasi dapat menerima <i>input</i> nomor <i>recipient</i> secara manual.	√
[REQ CRT - 02]	Aplikasi dapat menerima <i>input</i> nomor <i>recipient</i> dari nomor kontak pada <i>device</i> .	√
[REQ CRT - 03]	Aplikasi dapat menerima <i>input</i> nomor <i>recipient</i> dari grup kontak pada <i>device</i> .	
[REQ CRT - 04]	Aplikasi dapat menggunakan <i>multiple recipient</i> .	√
[REQ CRT - 05]	Aplikasi dapat memvalidasi <i>input</i> waktu pengiriman.	√
[REQ CRT - 06]	Aplikasi dapat menampilkan pesan <i>error</i> jika waktu pengiriman yang diinputkan adalah waktu lampau.	√
[REQ CRT - 07]	Aplikasi menyediakan pilihan frekuensi pengiriman.	√
[REQ CRT - 08]	Aplikasi dapat menggunakan <i>template</i> yang sudah dibuat.	√
[REQ CRT - 09]	Aplikasi dapat menyimpan <i>schedule</i> baru di <i>database</i>	√
[REQ CRT - 10]	Aplikasi dapat melakukan pengecekan <i>content schedule</i> untuk membedakan penyimpanan <i>schedule</i> ke <i>database</i> (<i>normal message</i> dan <i>typical message</i>).	√
[REQ F - 02]	Aplikasi menyediakan fitur untuk mengubah/ mengedit <i>schedule</i> pengiriman pesan	
[REQ UPD - 01]	Aplikasi menyediakan pilihan edit, yaitu edit all (edit secara keseluruhan), edit recipient (edit penerima pesan saja), edit date and time (edit waktu pengiriman saja) dan edit content (edit isi pesan dan frekuensi pengiriman).	
[REQ UPD - 02]	Aplikasi dapat menampilkan data-data sebelumnya atau data yang sudah tersimpan di <i>database</i> , dari <i>schedule</i> yang akan diubah	
[REQ UPD - 03]	Aplikasi dapat menerima input nomor recipient secara manual.	
[REQ UPD - 04]	Aplikasi dapat menerima input nomor recipient dari nomor kontak pada <i>device</i> .	
[REQ UPD - 05]	Aplikasi dapat menerima input nomor recipient dari grup kontak pada <i>device</i> .	
[REQ UPD - 06]	Aplikasi dapat menggunakan <i>multiple recipient</i> .	
[REQ UPD - 07]	Aplikasi dapat memvalidasi input waktu pengiriman.	
[REQ UPD - 08]	Aplikasi dapat menampilkan pesan <i>error</i> jika waktu pengiriman yang diinputkan adalah waktu lampau.	
[REQ UPD - 09]	Aplikasi dapat menyediakan pilihan frekuensi pengiriman.	
[REQ UPD - 10]	Aplikasi dapat menggunakan <i>template</i> yang sudah dibuat.	
[REQ UPD - 11]	Aplikasi dapat menyimpan <i>schedule</i> yang diubah ke <i>database</i> .	
[REQ UPD - 12]	Aplikasi dapat melakukan pengecekan <i>content schedule</i> untuk	

Nomor <i>Requirement</i>	Deksripsi	Status Implementasi
	membedakan penyimpanan <i>schedule</i> ke <i>database</i> (<i>normal message</i> dan <i>typical message</i>).	
[REQ F - 03]	Aplikasi menyediakan fitur untuk menghapus <i>schedule</i> pengiriman pesan	
[REQ DEL - 01]	Aplikasi dapat menampilkan pesan konfirmasi penghapusan <i>schedule</i> .	
[REQ DEL - 02]	Aplikasi dapat menampilkan pesan bahwa <i>schedule</i> berhasil dihapus.	
[REQ DEL - 03]	Aplikasi dapat menghapus <i>schedule</i> di <i>database</i> .	
[REQ F - 04]	Aplikasi menyediakan fitur untuk menghentikan sementara <i>schedule</i> pengiriman pesan.	
[REQ PSD - 01]	Aplikasi dapat menunda <i>schedule</i> pengiriman pesan.	
[REQ PSD - 02]	Aplikasi dapat melanjutkan (resume) <i>schedule</i> pengiriman pesan.	
[REQ PSD - 03]	Aplikasi dapat melakukan validasi waktu untuk <i>schedule</i> pengiriman pesan yang di-resume.	
[REQ PSD - 04]	Aplikasi dapat menampilkan pesan error jika proses resume gagal.	
[REQ F - 05]	Aplikasi dapat mengirimkan pesan sesuai dengan <i>schedule</i> yang telah dibuat	√
[REQ SND - 01]	Aplikasi dapat melakukan pengiriman pesan sesuai dengan <i>schedule</i> yang sebelumnya dibuat oleh <i>user</i> .	√
[REQ SND - 02]	Aplikasi dapat melakukan proses perhitungan usia jika <i>user</i> menyisipkan <i>string</i> usia pada <i>content schedule</i> . Perhitungan yang dimaksud adalah menghitung usia dari waktu ketika <i>schedule</i> akan dikirimkan dengan waktu yang digunakan <i>user</i> pada <i>schedule</i> tersebut	√
[REQ SND - 03]	Aplikasi dapat menampilkan <i>string</i> tahun yang disisipkan <i>user</i> pada <i>content schedule</i> . Tahun yang ditampilkan adalah tahun dimana pesan tersebut dikirimkan.	√
[REQ SND - 04]	Aplikasi dapat menampilkan bulan atau tanggal yang disisipkan <i>user</i> pada <i>content schedule</i> sesuai dengan waktu yang digunakan <i>user</i> pada <i>schedule</i> tersebut.	√
[REQ F - 06]	Aplikasi menyediakan fitur untuk dapat mengelompokkan <i>schedule</i> pengiriman pesan sesuai dengan kategorinya.	
[REQ FOL - 01]	Aplikasi dapat mengelompokkan pesan dalam kategori <i>schedule</i> (pesan yang sedang dischedulekan dan belum dikirim).	
[REQ FOL - 02]	Aplikasi dapat mengelompokkan pesan dalam kategori <i>sent</i> (pesan yang sudah berhasil dikirim).	
[REQ FOL - 03]	Aplikasi dapat mengambil pesan untuk kategori <i>failed</i> (pesan yang gagal dikirim).	
[REQ FOL - 04]	Aplikasi dapat mengambil pesan untuk kategori <i>paused</i> (pesan yang sedang ditunda).	

Nomor Requirement	Deksripsi	Status Implementasi
[REQ FOL - 05]	Aplikasi dapat mengambil pesan untuk kategori <i>history</i> (laporan pengiriman pesan untuk pesan yang berhasil dikirim dan pesan yang gagal terkirim).	
[REQ FOL - 06]	Aplikasi dapat menampilkan detail dari <i>history</i> pengiriman pesan.	
[REQ F - 07]	Aplikasi menyediakan fitur <i>alert/ pop up</i>.	
[REQ ALR - 01]	Aplikasi dapat menampilkan pop up beberapa waktu sebelum pengiriman schedule, sesuai setting waktu yang ditentukan user.	
[REQ ALR - 02]	Aplikasi dapat menampilkan alert/ pop up secara berkala setiap 5 menit sekali jika user tidak menanggapi alert yang muncul.	
[REQ ALR - 03]	Aplikasi mampu menyertakan pilihan nada yang ada di ponsel user ketika pop up muncul.	
[REQ ALR - 04]	Aplikasi dapat menampilkan detail dari schedule yang akan dikirim.	
[REQ ALR - 05]	Aplikasi dapat menghentikan/ membatalkan pengiriman schedule.	
[REQ ALR - 06]	Aplikasi dapat meneruskan proses pengiriman schedule.	
[REQ F - 08]	Aplikasi menyediakan fitur notifikasi <i>schedule</i>	√
[REQ NTF - 01]	Aplikasi dapat menampilkan notifikasi untuk status pesan yang berhasil dikirim dan berhasil diterima <i>recipient</i> .	√
[REQ NTF - 02]	Aplikasi dapat menampilkan notifikasi untuk status pesan yang gagal dikirim.	√
[REQ NTF - 03]	Aplikasi mampu menyertakan pilihan nada yang ada di ponsel <i>user</i> ketika notifikasi muncul.	
[REQ F - 09]	Aplikasi menyediakan fitur <i>template</i>.	√
[REQ TMP - 01]	Aplikasi dapat menampilkan <i>form</i> pembuatan <i>template</i> baru.	√
[REQ TMP - 02]	Aplikasi dapat menerima data masukkan dari <i>user</i> .	√
[REQ TMP - 03]	Aplikasi dapat menyisipkan data tanggal, bulan, tahun, atau usia.	√
[REQ TMP - 04]	Aplikasi mampu menyimpan <i>template</i> yang telah dibuat oleh <i>user</i> pada <i>database</i> .	√
[REQ TMP - 05]	Aplikasi mampu menampilkan kembali <i>template</i> yang telah dibuat oleh <i>user</i> dalam bentuk <i>list</i> .	√
[REQ TMP - 06]	<i>Template</i> yang telah dibuat oleh <i>user</i> dapat digunakan kembali sebagai pesan.	√
[REQ TMP - 07]	Aplikasi mampu menghapus <i>template</i> bila <i>user</i> memilih <i>template</i> yang ingin dihapus.	√
[REQ F - 10]	Aplikasi menyediakan fitur <i>setting</i>.	
[REQ F - 10.1]	Aplikasi menyediakan fitur untuk mengatur pengaturan alert sesuai dengan keinginan user.	
[REQ SA - 01]	Aplikasi menyediakan pilihan untuk mengaktifkan <i>alert</i> dan untuk menon-aktifkan <i>alert</i> .	

Nomor <i>Requirement</i>	Deksripsi	Status Implementasi
[REQ SA - 02]	Aplikasi mampu menampilkan pilihan waktu untuk <i>alert</i> .	
[REQ SA - 03]	Aplikasi mampu menampilkan pilihan alert sound baik dari memori kartu ataupun memori perangkat.	
[REQ SA - 04]	Aplikasi dapat menerima pilihan dari <i>user</i> .	
[REQ SA - 05]	Aplikasi dapat menyimpan pilihan <i>user</i> , baik pada database maupun pada <i>file</i> di <i>device system</i> dan menjalankannya.	
[REQ F - 10.2]	Aplikasi menyediakan fitur untuk mengatur pengaturan notifikasi sesuai dengan keinginan <i>user</i> .	
[REQ SN- 01]	Aplikasi mampu menampilkan pilihan <i>notification sound</i> baik dari memori kartu ataupun memori perangkat.	
[REQ SN - 02]	Aplikasi mampu menerima <i>input</i> pilihan dari <i>user</i> .	
[REQ SN - 03]	Aplikasi dapat menyimpan pilihan <i>user</i> , baik pada database maupun pada <i>file</i> di <i>device system</i> dan menjalankannya.	

Nomor Exception	Deksripsi	Status Implementasi
[EXCEPTION - 01]	Aplikasi membatalkan proses pembuatan atau perubahan <i>schedule</i> baru jika <i>device</i> mati atau <i>hang</i> saat dilakukan konfirmasi pembuatan atau pengubahan <i>schedule</i>	✓
[EXCEPTION - 02]	Aplikasi membatalkan penyimpanan <i>schedule</i> ke <i>database</i> jika <i>device</i> mati atau <i>hang</i> saat proses penyimpanan berlangsung	✓
[EXCEPTION - 03]	Aplikasi membatalkan proses penghapusan <i>schedule</i> dari <i>database</i> jika <i>device</i> mati atau <i>hang</i> saat proses penghapusan berlangsung	
[EXCEPTION - 04]	Aplikasi membatalkan proses <i>pause schedule</i> jika <i>device</i> mati atau <i>hang</i> saat dilakukan konfirmasi untuk melakukan <i>pause schedule</i>	
[EXCEPTION - 05]	Aplikasi membatalkan proses <i>pause schedule</i> jika <i>device</i> mati atau <i>hang</i> saat dilakukan konfirmasi untuk melakukan <i>pause schedule</i>	✓
[EXCEPTION - 06]	Aplikasi membatalkan pengiriman pesan saat proses pengiriman berlangsung jika pulsa pada <i>device</i> tidak mencukupi	✓
[EXCEPTION - 07]	Aplikasi membatalkan pengiriman pesan jika sinyal tidak tersedia saat proses pengiriman berlangsung	✓
[EXCEPTION - 08]	Aplikasi membatalkan pengiriman pesan jika <i>device</i> mati atau <i>hang</i> saat proses pengiriman berlangsung	
[EXCEPTION - 09]	Aplikasi membatalkan proses pembuatan <i>template</i> baru jika <i>device</i> mati atau <i>hang</i> saat dilakukan konfirmasi pembuatan <i>template</i>	✓
[EXCEPTION - 10]	Aplikasi membatalkan penyimpanan <i>template</i> ke <i>database</i> jika <i>device</i> mati atau <i>hang</i> saat proses penyimpanan berlangsung	✓

Keterangan ✓ : sudah terimplementasi

Lampiran D
Form Kuisioner

SMS Scheduler

SMS Scheduler adalah suatu aplikasi untuk melakukan pengiriman SMS yang sebelumnya telah diatur jadwalnya terlebih dahulu. Maksudnya, anda dapat mengatur kapan SMS ingin dikirimkan, dan kepada siapa SMS tersebut ditujukan. Nantinya, aplikasi ini yang akan melakukan pengiriman SMS yang sudah anda buat sesuai dengan waktu yang anda tentukan. Untuk keperluan Tugas Akhir, kami akan menambahkan beberapa fitur, yaitu template, edit, pause-resume, alert, delete, dan notification. Untuk itu, dimohon bantuanmu untuk mengisi kuisioner berikut ini.

* Wajib

Menurut anda, apakah fitur template diperlukan dalam pembuatan SMS ? *

- Ya
 Tidak

Menurut anda, apakah akan membantu jika aplikasi memiliki kemampuan yang dapat mempermudah anda untuk mengirimkan pesan ucapan ulang tahun atau hari jadi ? *

Misalnya, ketika anda ingin mengirimkan pesan "Selamat ulang tahun yang ke-xx", dimana xx merupakan usia seseorang yang dapat berubah sesuai dengan usia orang yang dituju

- Ya
 Tidak

Menurut anda, apakah jadwal yang sudah dibuat perlu disediakan pilihan untuk merubahnya ? *

- Ya
 Tidak

Menurut anda, apakah perlu di sediakan pilihan edit yang dilakukan secara keseluruhan (Edit All) ? *

- Ya
 Tidak

Menurut anda, apakah perlu di sediakan pilihan edit yang dilakukan hanya pada isi messagenya saja (Edit Message) ? *

Form edit yang ditampilkan hanya untuk merubah konten/ isi pesan saja, sehingga data lainnya dianggap tetap

- Ya
 Tidak

Menurut anda, apakah perlu di sediakan pilihan edit yang dilakukan hanya pada tanggal dan waktu pengiriman saja (Edit Date and Time) ? *

Form edit yang ditampilkan hanya untuk merubah tanggal dan waktu pengiriman saja, sehingga data lainnya dianggap tetap

- Ya
 Tidak

Menurut anda, apakah perlu di sediakan pilihan edit yang dilakukan hanya pada penerima pesan saja (Edit Recipient) ? *

Form edit yang ditampilkan hanya untuk merubah penerima pesan saja, sehingga data lainnya dianggap tetap

- Ya
 Tidak

Menurut anda, apakah perlu disediakan pilihan untuk melakukan pause pada jadwal yang telah dibuat ? *

Misalnya jika anda ingin menghentikan sementara pengiriman pesan dari jadwal yang sudah anda buat

- Ya
 Tidak

Menurut anda, apakah perlu disediakan pilihan untuk melakukan resume pada jadwal yang telah dipause ? *

Untuk melanjutkan pengiriman pesan dari jadwal yang dipause

- Ya
 Tidak

Menurut anda, apakah perlu disediakan pilihan untuk melakukan pause pada jadwal yang telah dibuat ? *

Misalnya jika anda ingin menghentikan sementara pengiriman pesan dari jadwal yang sudah anda buat

- Ya
 Tidak

Menurut anda, apakah perlu disediakan pilihan untuk melakukan resume pada jadwal yang telah dipause ? *

Untuk melanjutkan pengiriman pesan dari jadwal yang dipause

- Ya
 Tidak

Menurut anda, apakah perlu disediakan pilihan untuk menghapus jadwal yang telah dibuat ? *

- Ya
 Tidak

Menurut anda, apakah perlu disediakan pengingat (alert) untuk pesan yang akan dikirim ? *

contohnya, 5 menit sebelum jadwal akan dilaksanakan, akan ada pengingat bahwa ada pesan yang akan dikirimkan

- Ya
 Tidak

Menurut anda, apakah perlu disediakan notifikasi ketika pesan sedang dikirimkan ? *

contoh notifikasi : "Sending message..."

- Ya
 Tidak

Menurut anda, apakah perlu disediakan notifikasi ketika pesan gagal dikirimkan ? *

Nomor-nomor pesan yang gagal dikirimkan ditampilkan

- Ya
 Tidak

Menurut anda, apakah perlu disediakan pilihan untuk melakukan pengiriman ulang (resend) terhadap pesan yang gagal pada pertanyaan diatas ? *

- Ya
 Tidak

Menurut anda, apakah perlu disediakan notifikasi ketika pesan berhasil dikirimkan ? *

Nomor-nomor pesan yang berhasil dikirimkan ditampilkan

- Ya
 Tidak

Menurut anda, apakah perlu disediakan pengelompokan (group) terhadap pesan ? *

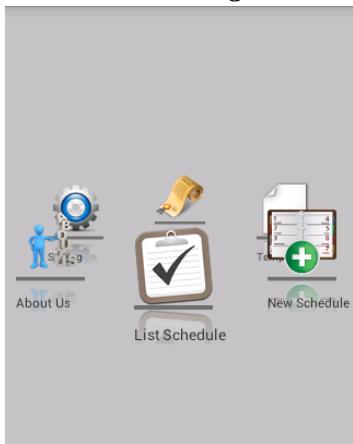
Misalnya, kelompok pesan yang masih terjadwal, kelompok pesan yang berhasil dikirimkan, kelompok pesan yang gagal dikirimkan, atau kelompok pesan yang dipause

- Ya
 Tidak

Lampiran E

Screenshot Implementasi Aplikasi SMS Scheduler

Screenshot Home Page



Ketika aplikasi dibuka, halaman ini yang akan ditampilkan terlebih dahulu. Pada halaman ini terdapat menu-menu dari aplikasi SMS Scheduler, yaitu, New Schedule, List Schedule, Template, History, Setting dan About Us

Screenshot New Schedule (Form New Schedule)

New Schedule

Recipient(s)

Date and time

Once Stop after:

Add data to message

Type your message here

Save Cancel

Untuk membuat jadwal baru, akan ditampilkan halaman *New Schedule* yang merupakan *form* isian, dimana terdapat data *recipient* (penerima pesan), tanggal dan waktu pengiriman, frekuensi pengiriman, serta isi dari pesan yang akan dikirimkan untuk membuat sebuah jadwal baru, seperti yang ditampilkan pada *screenshot* disamping.

Screenshot New Schedule (Contact Picker)

Select Contact

Search

5510	<input type="checkbox"/>
5554	<input type="checkbox"/>
5556	<input type="checkbox"/>
5558	<input type="checkbox"/>

Add CANCEL

(a)

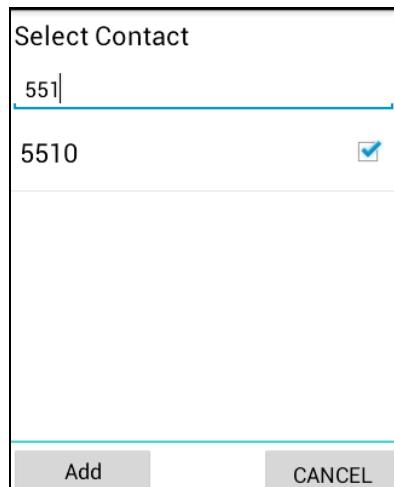
Select Contact

Search

5510	<input checked="" type="checkbox"/>
5554	<input checked="" type="checkbox"/>
5556	<input checked="" type="checkbox"/>
5558	<input type="checkbox"/>

Add CANCEL

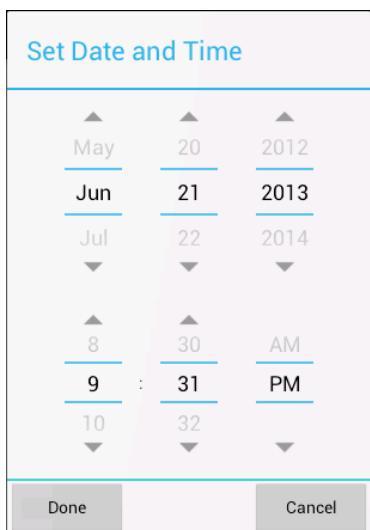
(b)



(c)

Ketika *user* menginginkan kontak *recipient* diambil dari kontak yang ada pada *device*, maka akan ditampilkan halaman *contact picker* seperti yang digambarkan pada gambar diatas. Pada gambar (a) merupakan tampilan list kontak yang ada pada *device*. Pada gambar (b) merupakan tampilan untuk kontak yang akan digunakan pada *schedule*. Dan pada gambar (c) merupakan tampilan untuk fungsi *predictive search*.

Screenshot New Schedule (Menampilkan Pengaturan Tanggal dan Waktu)



Untuk mengatur tanggal dan waktu pengiriman pesan, ditampilkan *date and time picker* seperti yang digambarkan pada gambar disamping.

Screenshot Save Schedule

New Schedule

5510;5554;5556;5558

21/6/2013 ; 21:31

Once

Stop after :

Add data to message

SMS Scheduler is an application to help you to send a pending sms...

Scheduled

Save Cancel

Screenshot pada gambar disamping menggambarkan permberitahuan pembuatan *schedule*, yaitu bahwa *schedule* telah berhasil dibuat.

Screenshot Notification

New Schedule

Recipient(s)

Date and time

Stop after :

Add data to message

Type your message here

SMS sent

Save Cancel

Screenshot pada gambar disamping menggambarkan permberitahuan pengiriman pesan (memberikan informasi bahwa pesan dari *schedule* yang dibuat telah berhasil dikirim atau gagal dikirim).