

**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, MT

Joe Lian Min, M.Eng

Disusun oleh :

Keukeu Anggarani Putri **101511015**

Laras Ervintyana Deni K.S **101511016**

Rizal Zulfikar Rinanda **101511028**



**JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA
POLITEKNIK NEGERI BANDUNG
2013**

ABSTRAK

Tugas akhir ini berkaitan dengan pembuatan aplikasi pengiriman pesan berupa SMS yang sebelumnya telah dijadwalkan, dengan nama *SMS Scheduler*. Permasalahan yang belum ditangani pada aplikasi-aplikasi *SMS Scheduler* yang ada saat ini, baik pada *platform* Android, Blackberry, iOS, maupun Java ME adalah tidak adanya informasi apakah pesan yang dikirimkan berhasil atau gagal, apakah pesan telah diterima *recipient* atau belum, tidak adanya pemberitahuan jika pesan akan dikirimkan, dan tidak adanya fitur yang menangani pengiriman pesan khusus untuk ucapan selamat ulang tahun atau selamat hari jadi (*anniversary*), dimana pada pesan tersebut terdapat tanggal, bulan, tahun atau usia yang dapat diproses setiap kali pesan dikirimkan. Masalah lainnya terdapat pada aplikasi untuk *platform* Java ME, yaitu nomor *recipient* yang digunakan harus ditulis secara manual (tidak dapat memilih dari kontak *device*). Selain itu, ada pula masalah dengan mekanisme pengambilan nomor penerima pesan pada aplikasi untuk *platform* Blackberry, yaitu pengambilan nomor dari kontak harus diulang sebanyak penerima pesan yang akan digunakan dan ketika melakukan pemilihan kontak tidak terlihat kontak mana yang sebelumnya sudah dipilih. Metode pengembangan aplikasi yang digunakan adalah metode *Incremental*, dimana setiap *increment*-nya ditargetkan untuk menyelesaikan setiap fitur untuk aplikasi ini, kecuali pada *increment* pertama yang ditargetkan untuk menganalisis aplikasi yang ada saat ini dan menganalisis aplikasi yang akan dibuat.

Aplikasi *SMS Scheduler* ini berupa aplikasi *mobile device* untuk *platform* Android, Blackberry, iOS, dan Java ME yang dapat digunakan untuk membuat jadwal pengiriman pesan dan mengirimkan pesan tersebut sesuai jadwal yang dibuat, serta dengan beberapa fitur tambahan untuk mendukung penggunaan aplikasi ini, yaitu fitur untuk memilih nomor *recipient* dari kontak *device*, memilih lebih dari satu *recipient* (*multiple recipient*), memberikan informasi jika pesan sudah diterima oleh *recipient*, memberikan informasi jika pesan gagal dikirim, menyediakan fitur *template* khusus, serta menyediakan *alert* sebelum pesan dikirimkan.

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

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 Penguji I yang telah menguji dan memberikan masukan pada Seminar I.
- Bapak Titis Sutisna, S.ST. selaku Pembimbing II dan Penguji II pada Seminar I, yang telah memberikan bimbingan, dukungan, pengarahan, saran, dan motivasi kepada penulis dalam menyusun dan menyelesaikan Tugas Akhir ini, serta selaku Penguji II yang telah menguji dan memberikan masukan pada Seminar I.
- Bapak Ade Chandra Nugraha, S.Si., MT. selaku Penguji I yang telah menguji dan memberikan masukan pada Seminar II.
- 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 membalas amal yang telah diberikan kepada kami. Amin.

Bandung, Juni 2013

Kelompok TA 109

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	iii
DAFTAR GAMBAR	v
DAFTAR TABEL.....	vii
DAFTAR SINGKATAN	xii
DAFTAR SIMBOL	xiii
BAB I Pendahuluan.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	3
1.3 Tujuan dan Manfaat Sistem.....	3
1.4 Metode Pengembangan Aplikasi.....	4
1.5 Batasan Aplikasi.....	10
1.6 Lingkup Sistem.....	11
BAB II Tinjauan Pustaka.....	12
2.1 Teori Pendukung	12
2.1.1 <i>Short Message Services (SMS)</i>	12
2.1.2 <i>SMS Scheduler</i>	13
2.1.3 <i>Model Incremental</i>	14
2.1.4 Android	15
2.1.5 Blackberry.....	18
2.1.6 iOS	18
2.1.7 Java ME/ J2ME.....	19
2.1.8 Bahasa Pemrograman Java	20
2.1.9 <i>Predictive Search</i>	21
2.1.10 Konsep <i>Object Oriented</i>	22
2.1.11 <i>Native Development</i>	22
2.1.12 <i>Mapping Tools</i> (<i>Pemetaan Tools</i>)	23
2.1.13 Teknologi Pendukung	23
2.1.14 Eclipse IDE.....	23
2.1.15 Netbeans IDE.....	24
2.1.16 Xcode IDE	25
2.1.17 ADT <i>Plugin</i>	25
2.1.18 Blackberry <i>Plugin</i>	25
2.1.19 Android SDK.....	26
2.1.20 Blackberry SDK.....	26
2.1.21 iOS SDK	26
BAB III Hasil Analisis dan Pembahasan.....	27
3.1 Analisis dan Evaluasi Sistem Saat Ini	27
3.1.1 Analisis Sistem Saat Ini	27
3.1.1.1 SMS Scheduler (<i>Author</i> : Lylynx.com)	28
3.1.1.2 Auto SMS (<i>Author</i> : Thein Min Naing).....	29
3.1.1.3 SMS Scheduler (<i>Author</i> : DiorDNA Apps)	30

3.1.1.4 SMS Scheduler (<i>Author : Vinsol</i>)	31
3.1.1.5 Holo SMS Scheduler (<i>Author : Casey Vockrodt, Destruction Labs</i>)	32
3.1.2 Evaluasi Sistem Saat Ini	33
3.2 Analisis Kebutuhan	44
3.2.1 Domain Kajian.....	45
3.2.2 Teknologi	46
3.2.2.1 Teknik Pembuatan Aplikasi	46
3.2.2.2 Tools.....	47
3.3 Pengumpulan Data (Survei).....	49
3.3.1 Kesimpulan Hasil Survei	50
3.4 Kesimpulan Analisis dan Pembahasan	50
BAB IV Perancangan.....	52
4.1 Perancangan Data	52
4.2 Arsitektur Sistem	56
4.2.1 Perancangan Arsitektur Aplikasi SMS Scheduler	56
4.2.2 Perancangan Proses Aplikasi SMS Scheduler	65
4.2.2.1 Sequence Diagram Membuat Schedule	65
4.3 Perancangan User Interface	118
BAB V Implementasi.....	156
5.1 Batasan Implementasi.....	156
5.2 Implementasi Data	156
5.3 Implementasi Requirement	157
5.4 Implementasi Fitur Aplikasi/ Sistem	157
5.4.1 Fitur New Schedule	157
5.4.1.1 Screenshot Home Page	157
5.4.1.2 Screenshot New Schedule (Form New Schedule)	158
5.4.1.3 Screenshot New Schedule (Contact Picker).....	158
5.4.1.4 Screenshot New Schedule (Menampilkan Pengaturan Tanggal dan Waktu).....	159
BAB VI Pengujian.....	161
6.1 Batasan Implementasi	161
6.2 Perencanaan Pengujian	161
6.3 Hasil Pengujian.....	162
DAFTAR PUSTAKA	164
Lampiran A	A-1
Lampiran B	B-1
Lampiran C	C-1
Lampiran D	D-1

DAFTAR GAMBAR

Gambar 1 . Model SDLC <i>Incremental</i>	14
Gambar 2 . Arsitektur Android.....	16
Gambar 3 . Java <i>Platform, Micro Edition</i> (Java ME).....	20
Gambar 4 . Cara kerja Java.....	21
Gambar 5 . <i>Screenshot</i> aplikasi SMS <i>Scheduler</i> (<i>Author</i> : Lylynx.com)	29
Gambar 6 . <i>Screenshot</i> aplikasi Auto SMS (<i>Author</i> : Thein Min Naing).....	30
Gambar 7 . <i>Screenshot</i> aplikasi SMS <i>Scheduler</i> (<i>Author</i> : DiorDNA Apps).....	31
Gambar 8 . <i>Screenshot</i> aplikasi SMS <i>Scheduler</i> (<i>Author</i> : Vinsol)	32
Gambar 9 . <i>Screenshot</i> aplikasi Holo SMS <i>Scheduler</i> (<i>Author</i> : Casey Vockrodt, Destruction Labs)	33
Gambar 10 . <i>Use case diagram current system SMS Scheduler</i>	39
Gambar 11 . <i>Physical Data Model (PDM) Database SMS Scheduler</i>	56
Gambar 12 . <i>Package diagram</i> aplikasi SMS <i>Scheduler</i>	56
Gambar 13 . <i>Class diagram</i> SMS <i>Scheduler</i>	58
Gambar 14 . <i>Sequence diagram New Schedule</i>	66
Gambar 15 . <i>Sequence diagram interrupt</i>	72
Gambar 16 . <i>Sequence diagram queue dan send message</i>	73
Gambar 17 . <i>Sequence diagram notification</i>	77
Gambar 18 . <i>Sequence diagram Menampilkan Kategori Pesan</i>	78
Gambar 19 . <i>Sequence diagram edit all</i>	80
Gambar 20 . <i>Sequence diagram edit recipient</i>	88
Gambar 21 . <i>Sequence diagram edit date and time schedule</i>	92
Gambar 22 . <i>Sequence diagram edit content schedule</i>	98
Gambar 23 . <i>Sequence diagram pause schedule</i>	101
Gambar 24 . <i>Sequence diagram resume schedule</i>	103
Gambar 25 . <i>Sequence diagram menampilkan alert</i>	105
Gambar 26 . <i>Sequence diagram membuat template</i>	107
Gambar 27 . <i>Sequence diagram menghapus template</i>	110
Gambar 28 . <i>Sequence diagram setting</i>	112
Gambar 29 . <i>Screenshot Home Page</i>	157
Gambar 30 . <i>Screenshot New Schedule (Form New Schedule)</i>	158

Gambar 31 .	<i>Screenshot New Schedule (Contact Picker)</i>	159
Gambar 32 .	<i>Screenshot New Schedule</i> (Menampilkan Pengaturan Tanggal dan Waktu)	159
Gambar 33 .	<i>Screenshot Save Schedule</i>	160
Gambar 34 .	<i>Screenshot Notification</i>	160

DAFTAR TABEL

Tabel 1.	Jadwal <i>increment</i>	4
Tabel 2.	Hasil analisis Aplikasi SMS <i>Scheduler</i> yang ada saat ini	35
Tabel 3.	Hasil analisis Aplikasi SMS <i>Scheduler</i> yang ada saat ini (Lanjutan).....	36
Tabel 4.	Hasil analisis Aplikasi SMS <i>Scheduler</i> yang ada saat ini (Lanjutan).....	37
Tabel 5.	Hasil analisis Aplikasi SMS <i>Scheduler</i> yang ada saat ini (Lanjutan).....	38
Tabel 6.	Deskripsi aktor.....	39
Tabel 7.	Deskripsi <i>case</i>	39
Tabel 8.	Skenario <i>case</i> menampilkan <i>form schedule</i>	40
Tabel 9.	Skenario <i>case</i> memvalidasi <i>form</i> dan waktu pengiriman	40
Tabel 10.	Skenario <i>case</i> memvalidasi <i>form</i> dan waktu pengiriman (Lanjutan)	41
Tabel 11.	Skenario <i>case</i> menyimpan <i>schedule</i> ke <i>database</i>	42
Tabel 12.	Skenario <i>case</i> melakukan pengecekan waktu pengiriman.....	42
Tabel 13.	Skenario <i>case</i> mengirimkan pesan.....	43
Tabel 14.	Skenario <i>case</i> menghapus <i>schedule</i>	44
Tabel 15.	Skenario <i>case</i> menampilkan notifikasi	44
Tabel 16.	Skenario <i>case</i> menampilkan <i>history</i>	44
Tabel 17.	Hasil Survei	50
Tabel 18.	Kamus data	52
Tabel 19.	Kamus data (Lanjutan)	53
Tabel 20.	Kamus data (Lanjutan)	54
Tabel 21.	Kamus data (Lanjutan)	55
Tabel 22.	Keterangan <i>Package Diagram</i>	57
Tabel 23.	Keterangan <i>Class Schedule</i>	59
Tabel 24.	Keterangan <i>Class ListScheduleView</i>	59
Tabel 25.	Keterangan <i>Class NewSchedule</i>	60
Tabel 26.	Keterangan <i>Class SMSActivity</i>	60
Tabel 27.	Keterangan <i>Class SMSManager</i>	60
Tabel 28.	Keterangan <i>Class TimeListDatabaseHelper</i>	61
Tabel 29.	Keterangan <i>Class DatabaseOpenHelper</i>	62
Tabel 30.	Keterangan <i>Class Frequency</i>	62
Tabel 31.	Keterangan <i>Class EditSchedule</i>	62
Tabel 32.	Keterangan <i>Class EditTime</i>	62

Tabel 33.	Keterangan <i>Class</i> EditRecipient	62
Tabel 34.	Keterangan <i>Class</i> EditContent.....	62
Tabel 35.	Keterangan <i>Class</i> Alert.....	63
Tabel 36.	Keterangan <i>Class</i> NewTemplate	63
Tabel 37.	Keterangan <i>Class</i> TemplateDatabaseHelper	63
Tabel 38.	Keterangan <i>Class</i> Template	63
Tabel 39.	Keterangan <i>Class</i> Setting.....	64
Tabel 40.	Keterangan <i>Class</i> DefaultSetting.....	64
Tabel 41.	Keterangan <i>sequence diagram</i> New Schedule	67
Tabel 42.	Spesifikasi proses addTimemillis	68
Tabel 43.	Spesifikasi proses setMessageType	68
Tabel 44.	Spesifikasi proses getType	69
Tabel 45.	Spesifikasi proses saveScheduleToMessage.....	69
Tabel 46.	Spesifikasi proses saveScheduleToType	69
Tabel 47.	Spesifikasi proses saveScheduleToTime.....	70
Tabel 48.	Spesifikasi proses saveScheduleToContact	70
Tabel 49.	Spesifikasi proses saveScheduleToRecipient	71
Tabel 50.	Spesifikasi proses repetition	71
Tabel 51.	Keterangan <i>sequence diagram</i> interrupt	74
Tabel 52.	Keterangan <i>sequence diagram</i> queue dan send message	74
Tabel 53.	Spesifikasi proses getMessageIdFromTime	75
Tabel 54.	Spesifikasi proses getMessageFromMessage	75
Tabel 55.	Spesifikasi proses getRecipientFromRecipient	76
Tabel 56.	Spesifikasi proses sendSMS	76
Tabel 57.	Keterangan <i>sequence diagram</i> notification	77
Tabel 58.	Keterangan <i>sequence diagram</i> Menampilkan Kategori Pesan	79
Tabel 59.	Keterangan <i>sequence diagram</i> edit all schedule	81
Tabel 60.	Spesifikasi proses getRecipient	82
Tabel 61.	Spesifikasi proses getMessageDatetime	83
Tabel 62.	Spesifikasi proses getMessageFrequency.....	83
Tabel 63.	Spesifikasi proses getTime	83
Tabel 64.	Spesifikasi proses getMessage.....	84
Tabel 65.	Spesifikasi proses deleteRecipient.....	84
Tabel 66.	Spesifikasi proses saveScheduleToRecipient	85

Tabel 67. Spesifikasi proses getType	85
Tabel 68. Spesifikasi proses updateType.....	85
Tabel 69. Spesifikasi proses updateContent	86
Tabel 70. Spesifikasi proses deleteTime.....	86
Tabel 71. Spesifikasi proses saveScheduleToTime.....	87
Tabel 72. Spesifikasi proses repetition	87
Tabel 73. Keterangan <i>sequence diagram edit recipient</i>	89
Tabel 74. Spesifikasi proses getRecipient	89
Tabel 75. Spesifikasi proses deleteRecipient.....	90
Tabel 76. Spesifikasi proses saveScheduleToRecipient	90
Tabel 77. Keterangan <i>sequence diagram edit date and time schedule</i>	93
Tabel 78. Spesifikasi proses updateTime	94
Tabel 79. Spesifikasi proses getTime	95
Tabel 80. Spesifikasi proses count.....	95
Tabel 81. Spesifikasi proses deleteTime.....	95
Tabel 82. Spesifikasi proses getMessageFrequency.....	96
Tabel 83. Spesifikasi proses saveScheduleToTime	96
Tabel 84. Spesifikasi proses repetition	97
Tabel 85. Keterangan <i>sequence diagram edit content schedule</i>	98
Tabel 86. Spesifikasi proses getMessage.....	99
Tabel 87. Spesifikasi proses updateType.....	100
Tabel 88. Spesifikasi proses updateContent	100
Tabel 89. Keterangan <i>sequence diagram pause schedule</i>	101
Tabel 90. Spesifikasi proses pauseSchedule.....	102
Tabel 91. Spesifikasi proses updateStatus	102
Tabel 92. Keterangan <i>sequence diagram resume schedule</i>	104
Tabel 93. Keterangan <i>sequence diagram menampilkan alert</i>	105
Tabel 94. Spesifikasi proses showAlert.....	106
Tabel 95. Keterangan <i>sequence diagram membuat template</i>	108
Tabel 96. Spesifikasi proses saveTemplateToCategory.....	108
Tabel 97. Spesifikasi proses getCategoryId.....	109
Tabel 98. Spesifikasi proses saveTemplateToTemplate	109
Tabel 99. Keterangan <i>sequence diagram menghapus template</i>	110
Tabel 100. Spesifikasi proses removeTemplate	110

Tabel 101. Keterangan <i>sequence diagram setting</i>	113
Tabel 102. Spesifikasi proses getDefaultAlertActivation	114
Tabel 103. Spesifikasi proses getDefaultAlertTime	114
Tabel 104. Spesifikasi proses getDefaultAlertTone	114
Tabel 105. Spesifikasi proses getDefaultNotifTone	115
Tabel 106. Spesifikasi proses saveSettingToSetting	115
Tabel 107. Spesifikasi proses getAlertActivation.....	115
Tabel 108. Spesifikasi proses updateAlertActivation.....	116
Tabel 109. Spesifikasi proses getAlertTime	116
Tabel 110. Spesifikasi proses updateAlertTime	116
Tabel 111. Spesifikasi proses getAlertTone	117
Tabel 112. Spesifikasi proses updateAlertTone	117
Tabel 113. Spesifikasi proses getNotifTone	118
Tabel 114. Spesifikasi proses updateNotifTone	118
Tabel 115. Rancangan UI halaman <i>New Schedule (Form New Schedule)</i>	119
Tabel 116. Rancangan UI halaman <i>New Schedule (Form New Schedule)</i> (Lanjutan)	120
Tabel 117. Rancangan UI halaman <i>New Schedule (Form New Schedule)</i> (Lanjutan)	121
Tabel 118. Rancangan UI halaman <i>New Schedule (Contact Picker)</i>	122
Tabel 119. Rancangan UI halaman <i>New Schedule (Contact Picker)</i> (Lanjutan).....	123
Tabel 120. Rancangan UI halaman <i>New Schedule</i> (Menampilkan Pengaturan Tanggal dan Waktu)	123
Tabel 121. Rancangan UI halaman <i>New Schedule</i> (Menampilkan Pengaturan Tanggal dan Waktu) (Lanjutan)	124
Tabel 122. Rancangan UI halaman <i>New Schedule</i> (Menampilkan <i>warning invalid schedule</i>)	125
Tabel 123. Rancangan UI halaman <i>New Schedule</i> (Menampilkan <i>warning schedule time expired</i>)	126
Tabel 124. Rancangan UI halaman <i>Edit Schedule (Edit All)</i>	127
Tabel 125. Rancangan UI halaman <i>Edit Schedule (Edit All)</i> (lanjutan)	128
Tabel 126. Rancangan UI halaman <i>Edit Schedule (Edit All)</i> (Lanjutan)	129
Tabel 127. Rancangan UI halaman <i>Edit Schedule (Edit Content)</i>	130
Tabel 128. Rancangan UI halaman <i>Edit Schedule (Edit Recipient)</i>	130
Tabel 129. Rancangan UI halaman <i>Edit Schedule (Edit Date and Time)</i>	131
Tabel 130. Rancangan UI halaman menghapus <i>schedule</i>	133
Tabel 131. Rancangan UI halaman kategori pesan (<i>List Schedule</i>)	133

Tabel 132. Rancangan UI halaman kategori pesan (<i>List Sent</i>)	135
Tabel 133. Rancangan UI halaman kategori pesan (<i>List Failed</i>)	136
Tabel 134. Rancangan UI halaman kategori pesan (<i>List Paused</i>)	137
Tabel 135. Rancangan UI halaman <i>history</i>	137
Tabel 136. Rancangan UI halaman <i>detail history</i>	138
Tabel 137. Rancangan UI halaman <i>Alert</i>	140
Tabel 138. Rancangan UI halaman <i>Alert (detail alert)</i>	141
Tabel 139. Rancangan UI halaman <i>Notification (sending)</i>	142
Tabel 140. Rancangan UI halaman <i>Notification (laporan pengiriman)</i>	143
Tabel 141. Rancangan UI halaman <i>Notification (laporan pengiriman) (Lanjutan)</i>	144
Tabel 142. Rancangan UI halaman <i>template</i>	144
Tabel 143. Rancangan UI halaman <i>template (Lanjutan)</i>	145
Tabel 144. Rancangan UI halaman <i>Template (New template)</i>	145
Tabel 145. Rancangan UI halaman <i>Template (New template) (Lanjutan)</i>	146
Tabel 146. Rancangan UI halaman <i>Template (New template) (Lanjutan)</i>	147
Tabel 147. Rancangan UI halaman <i>Template (Invalid Date)</i>	147
Tabel 148. Rancangan UI halaman <i>Template (Invalid Date)</i>	148
Tabel 149. Rancangan UI halaman <i>setting</i>	149
Tabel 150. Rancangan UI halaman <i>setting(alert)</i>	150
Tabel 151. Rancangan UI halaman <i>setting(Snooze duration)</i>	151
Tabel 152. Rancangan UI halaman <i>setting(notification)</i>	152
Tabel 153. Rancangan UI halaman <i>setting(sound picker)</i>	153
Tabel 154. Rancangan UI halaman <i>setting(sound picker) (Lanjutan)</i>	154
Tabel 155. Rancangan UI halaman <i>about us</i>	154
Tabel 156. Rancangan UI halaman <i>about us (Lanjutan)</i>	155
Tabel 157. Tabel status implementasi data.....	156
Tabel 158. <i>Test Case-1</i>	161
Tabel 159. <i>Test Case-2</i>	162
Tabel 160. <i>Test Case-3</i>	162
Tabel 161. Hasil pengujian	162

DAFTAR SINGKATAN

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

DAFTAR SIMBOL

Simbol	Nama	Penggunaan	Keterangan
	Atribut	<i>E-R diagram</i>	Menggambarkan atribut dari setiap <i>entity</i> .
	Entity	<i>E-R diagram</i>	Menggambarkan <i>entity</i> .
	Relationship	<i>E-R diagram</i>	Menggambarkan relasi antar <i>entity</i> .
	Aktor	<i>Use case diagram</i>	Menggambarkan aktor pada diagram kelas.
	UseCase	<i>Use case diagram</i>	Menggambarkan proses pada diagram <i>use case</i> .
	Association	<i>Use case diagram</i>	Menggambarkan hubungan asosiasi.
	Include	<i>Use case diagram</i>	Menggambarkan relasi <i>use case</i> dimana proses yang bersangkutan akan dilanjutkan ke proses yang dituju.
	Extend	<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.
	Object	<i>Sequence diagram</i>	Menggambarkan objek pada diagram.
	Object message	<i>Sequence diagram</i>	Menggambarkan pesan antar dua objek
	Return message	<i>Sequence diagram</i>	Menggambarkan pengembalian dari pemanggilan prosedur.
	Message to self	<i>Sequence diagram</i>	Menggambarkan pesan yang menuju dirinya sendiri.
	State	<i>Statechart diagram</i>	Menggambarkan suatu <i>state</i> pada diagram.
	Transition	<i>Statechart diagram</i>	Menggambarkan transisi dari suatu aktivitas ke aktivitas lainnya.

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. Hal ini dapat menimbulkan kesalahpahaman antara pengirim pesan dengan 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 data 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 ditujukan untuk menyelesaikan setiap fitur yang ada pada aplikasi SMS Scheduler. 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 	Mock up UI	-

Artifak dari *increment* 1 adalah hasil analisis aplikasi yang sudah ada saat ini, hasil analisis aplikasi yang akan dibuat, rancangan *database* aplikasi yang akan dibuat (secara umum), dan *mock up user interface* aplikasi yang akan dibuat (secara umum).

	Analisis	Design	Implementasi	Testing
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</i> (<i>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>. - menangani <i>typical message</i> (pesan dengan data yang dapat diproses setiap kali dilakukan pengiriman) 	<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> - Menampilkan pesan <i>error</i> ketika hasil validasi tidak valid 	<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>. 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>. 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 <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 - 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>. 	<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>. 	<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). 	<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)

	Analisis	Design	Implementasi	Testing
Increment 3 [Fitur mengubah <i>schedule</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"> - 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. - <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"> • 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> • 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"> - 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>. - 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 alert]	<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 - menangani pilihan yang disediakan pada <i>alert box</i> - melanjutkan pengiriman <i>schedule</i> - menghentikan pengiriman <i>schedule</i> - menampilkan <i>detail</i> 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> 	<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>, 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. - menghentikan pengiriman <i>schedule</i> - menangani <i>alert</i> yang tidak direspon oleh <i>user</i> hingga waktu pengirimannya tiba - mengambil <i>detail schedule</i> yang akan dikirimkan <i>User interface</i> untuk : <ul style="list-style-type: none"> - Informasi dan pilihan yang ditampilkan pada <i>alert box</i> - Menampilkan <i>detail</i> dari pesan yang akan dikirimkan 	<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. Menghentikan pengiriman <i>schedule</i> Mengambil dan menampilkan <i>detail 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> <i>User interface</i> 	<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' - menghentikan/membatalkan pengiriman pesan ketika <i>user</i> memilih 'no' - menampilkan <i>detail</i> 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 	<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 	<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 	<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>

	Analisis	Design	Implementasi	Testing
Increment 6 [Fitur notification]	<ul style="list-style-type: none"> Menentukan cara untuk menangani pengiriman ulang (<i>resend</i>) pesan yang gagal dikirim. 	<ul style="list-style-type: none"> Membuat rancangan algoritma untuk menangani pengiriman ulang (<i>resend</i>) <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 proses <i>resend</i> untuk pesan yang gagal dikirim - <i>User interface</i> 	<ul style="list-style-type: none"> - menampilkan notifikasi saat pesan gagal terkirim - melakukan pengiriman ulang untuk pesan yang gagal terkirim - 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"> - menghentikan sementara pengiriman <i>schedule (pause)</i> - me-resume <i>schedule</i> - menangani perubahan waktu pengiriman <i>schedule</i> yang sudah tersusun sebelumnya, dikarenakan <i>user</i> melakukan <i>paused</i> atau <i>resume schedule</i>. 	<ul style="list-style-type: none"> Membuat rancangan algoritma untuk : <ul style="list-style-type: none"> - melakukan <i>pause schedule</i> - 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"> Menghentikan sementara pengiriman <i>schedule (pause)</i> • 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 tes untuk : <ul style="list-style-type: none"> - melakukan <i>pause schedule</i> - 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 pengelompokkan 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 pengelompokkan kategori, aplikasi dengan fitur yang sebelumnya dibuat, ditambah dengan fitur pengelompokkan 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> - mengakses nada dering 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> - mengambil <i>setting</i> yang digunakan pada <i>device</i> - mengambil nada dering 	<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> • Membuat <i>default setting</i> 	<ul style="list-style-type: none"> Melakukan tes untuk : <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>

	Analisis	Design	Implementasi	Testing
Increment 10 [Fitur setting]	<ul style="list-style-type: none"> - 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"> - 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"> • 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"> - 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, *Scheduler* dan aplikasi SMS *Scheduler* itu sendiri, 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.

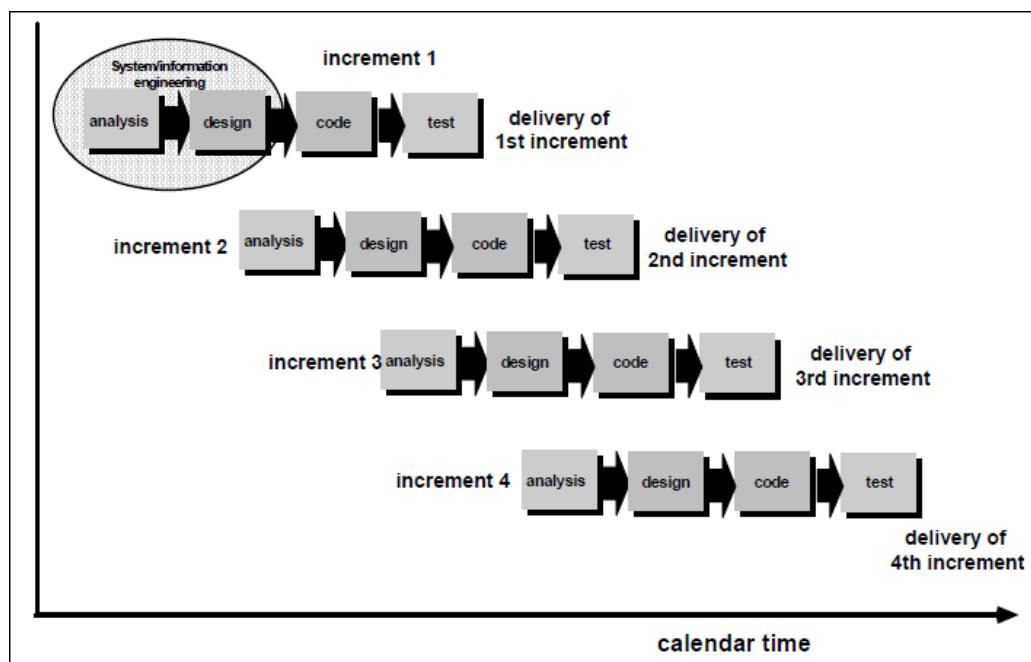
Kebanyakan para *developer* aplikasi SMS *Scheduler* membuat aplikasi tersebut dengan tujuan untuk membantu orang-orang yang sibuk, memberikan ucapan selamat ulang tahun atau hari jadi (*anniversary*) di tengah malam, mengingatkan hal-hal penting kepada teman atau keluarga, dan lainnya. Dapat disimpulkan bahwa aplikasi SMS *Scheduler* ini dibuat untuk membantu menyampaikan informasi, terutama informasi-informasi penting bagi Pembuatan Aplikasi SMS *Scheduler*

seseorang, yang harus disampaikan kepada orang lain pada waktu yang akan datang. Menurut Derie (2011, p.1), aplikasi ini merupakan aplikasi yang dirancang bagi orang-orang dengan keterbatasan waktu untuk melakukan hal-hal yang sangat kecil namun penting.

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 Model *Incremental*

Menurut Roger S. Pressman (2001 : 35) “model *incremental* menggabungkan elemen-elemen sekuensial *linier* (diaplikasikan secara berulang) dengan filosofi *prototype* iteratif. Setiap urutan *linier* menghasilkan pertambahan (*increment*) perangkat lunak [MDE93]”. Model *incremental* juga dapat diartikan sebagai model pengembangan sistem pada rekayasa perangkat lunak berdasarkan *requirement software* yang dibagi menjadi beberapa bagian sehingga model pengembangannya dilakukan secara bertahap. Daur hidup *incremental* dapat dilihat pada gambar 1.



Gambar 1 . Model SDLC *Incremental*

Sumber : Pressman (2001, p.35)

Model *incremental* menghasilkan bagian-bagian kecil *software* yang sudah dapat dipergunakan, yang disebut dengan *increment*. Biasanya, setiap *increment* dibangun bersadarkan produk *increment* yang sebelumnya telah selesai dibuat.

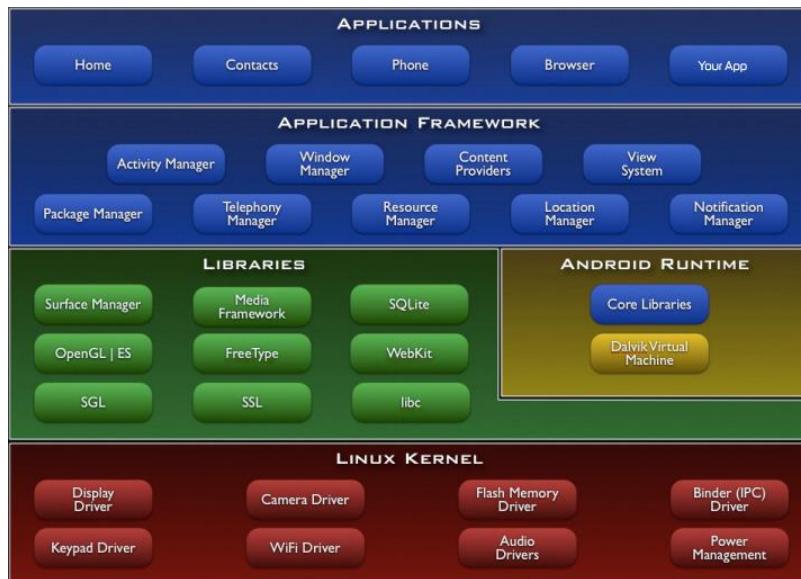
Ketika menggunakan model *increment*, maka produk yang dihasilkan adalah produk inti (*core product*) yang sudah dapat dipergunakan dan telah memenuhi kebutuhan dasar. *Core product* digunakan oleh *customer* (atau untuk dilakukan tinjauan lebih rinci). Hasil penggunaan atau tinjauan tersebut dibuat rencana untuk melakukan pembangunan pada *increment* berikutnya. Rencana tersebut membahas mengenai modifikasi atau perbaikan dari *core product* untuk memenuhi kebutuhan *customer*. Proses ini dilakukan secara berulang-ulang sampai produk yang dihasilkan sudah lengkap. 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.
2. Desain, merupakan proses perancangan desain aplikasi.
3. *Code*, merupakan proses implementasi dari perancangan yang sudah dibuat.
4. *Test*, merupakan proses pengujian 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* (optimisasi 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 :

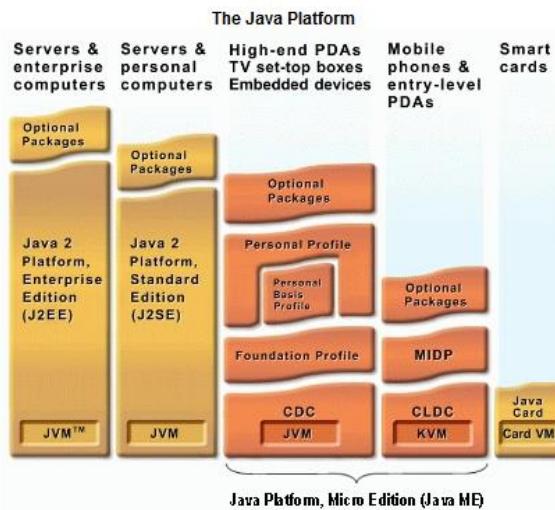
1. Konfigurasi (*configuration*), menyediakan kumpulan *library* utama dan kemampuan *virtual machine* untuk berbagai perangkat. Dua konfigurasi utama pada J2ME, yaitu konfigurasi untuk *small mobile device* yang disebut *Connected Limited Device Configuration* (CLDC) dan konfigurasi untuk *device* yang 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.



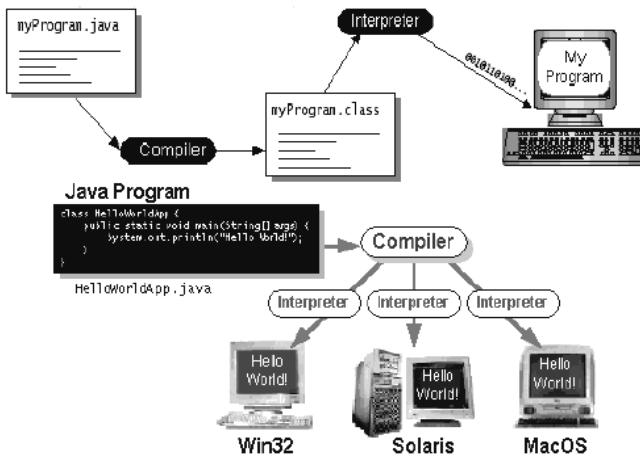
Gambar 3 . Java Platform, Micro Edition (Java ME)

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 Konsep *Object Oriented*

Salah satu konsep pengembangan aplikasi yang dipakai adalah konsep *object oriented*, karena pembuatan aplikasi akan menggunakan bahasa java dan telah diketahui bahwa pemakaian bahasa java lebih baik bila didukung dengan konsep *object oriented*. *Object oriented* merupakan salah satu teknik yang dapat memberikan kemudahan dalam pengembangan aplikasi.

Dalam konsep ini dikenal istilah *object* dan *class*, dimana sebuah *class* dapat menghasilkan beberapa *object*. *Class* merupakan pola atau *template* yang menggambarkan kumpulan *object* yang mempunyai sifat yang sama, sementara *object* adalah implementasi dari *class*.

2.1.11 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.

2.1.12 **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.

2.1.13 **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.

2.1.14 **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.

2.1.15 Netbeans IDE

Untuk membuat aplikasi SMS *Scheduler* pada *platform Java ME/ J2ME*, *tools* yang digunkannya 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.

2.1.16 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.

2.1.17 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.

2.1.18 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.

2.1.19 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.

2.1.20 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.

2.1.21 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

Analisis sistem dilakukan untuk mempelajari fungsi-fungsi apa saja yang ada pada aplikasi SMS *Scheduler* yang ada saat ini dan fungsi-fungsi apa saja yang dapat dikembangkan dari aplikasi tersebut. Selain itu, analisis juga dilakukan untuk mendapatkan *requirement-requirement* untuk aplikasi yang akan dibuat dan analisis terhadap teknologi yang akan digunakan pada aplikasi yang akan dibuat.

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 : Lylynnx.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.

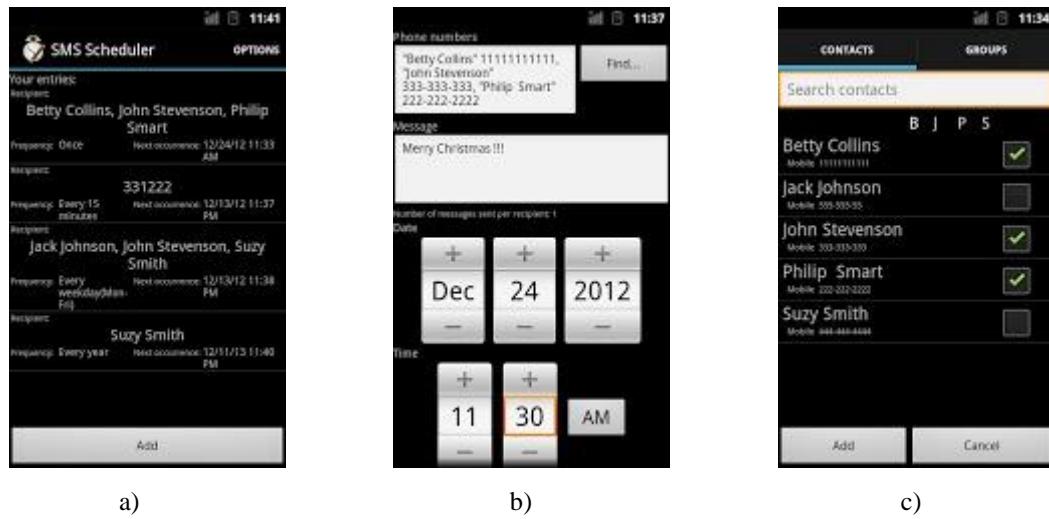
3.1.1.1 SMS Scheduler (*Author : Lylynx.com*)

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

- 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).
- Reply Message* (dapat merubah *auto-response* SMS message untuk *missed calls* dan SMS).
- Widget* (widget untuk menghidupkan/ mematikan *auto-response*).
- Auto On/Off* (dapat mengatur durasi menghidupkan *auto-response*, sehingga tidak perlu khawatir jika lupa untuk mematikannya).
- Silent Mode* (dapat mengatur Ringer Mode menjadi silent mode selama aplikasi dihidupkan dan mengembalikannya ke Ringer Mode secara otomatis setelah aplikasi dimatikan).
- 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.

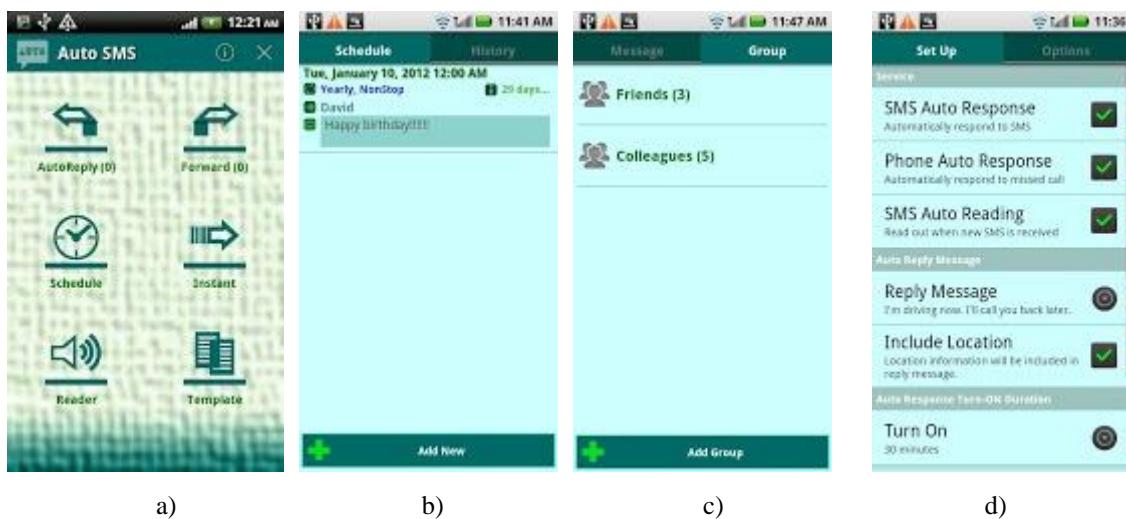
3. SMS READER

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.



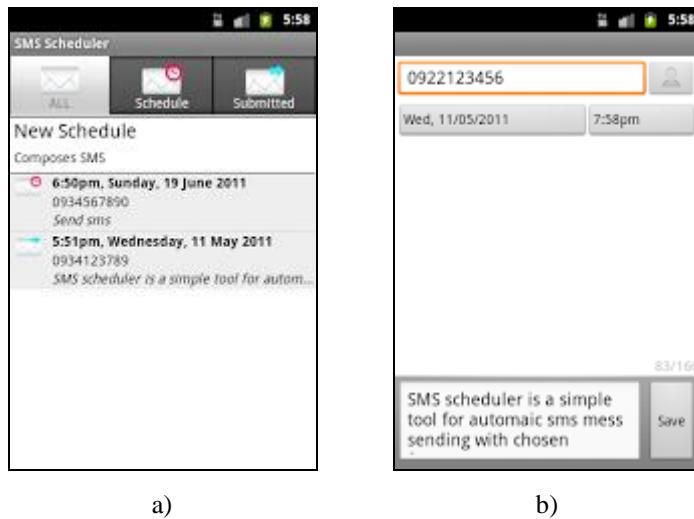
Gambar 6 . Screenshot aplikasi Auto SMS (Author : Thein Min Naing)

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*).



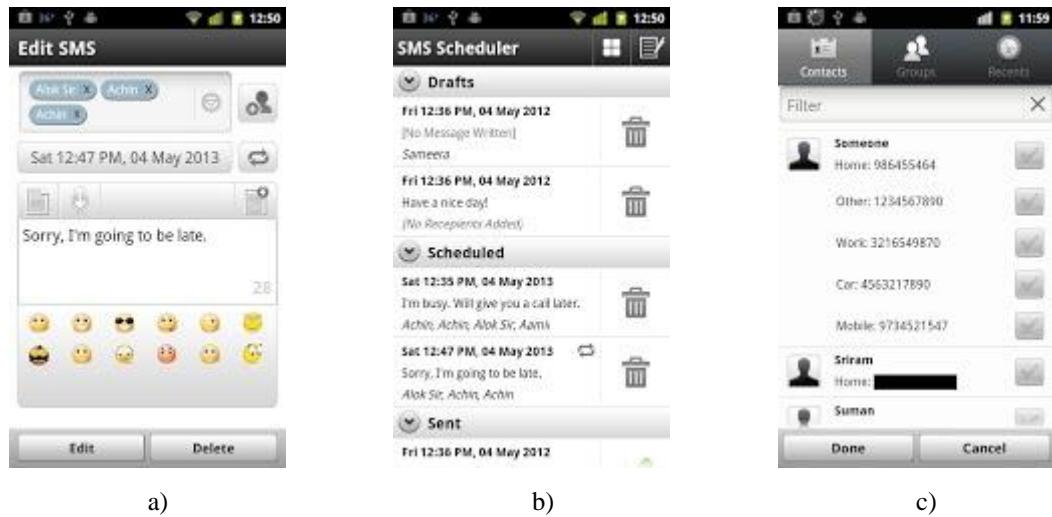
Gambar 7 . Screenshot aplikasi SMS Scheduler (Author : DiorDNA Apps)

- 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*.



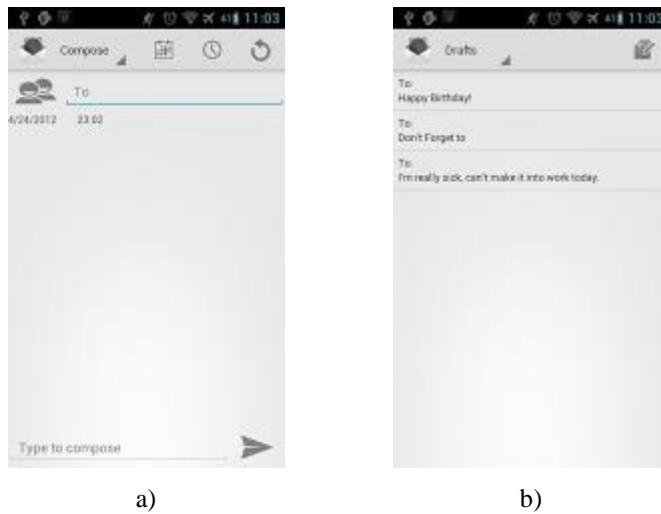
Gambar 8. Screenshot aplikasi SMS Scheduler (Author : Vinsol)

- a) halaman untuk membuat jadwal pengiriman pesan, b) halaman yang menampilkan pengelompokkan 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*.

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, Destrcution Labs
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
Dapat melakukan <i>filtering</i> saat melakukan pemilihan <i>recipient</i>	√	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) Shaperstu dio	(9) Liuge, Good App	(10) Casey Vockrodt, Destrcution Labs
Dapat melakukan pengiriman ke lebih dari satu recipient	√	√	√	√	√	√	√	√	√	√
Dapat melakukan pengiriman ke nomor dan pada jam yang sama, tetapi dengan jadwal yang berbeda	√	√	√	√	√	√	√	√	√	√
Memberikan pilihan frekuensi pengiriman pesan (dikirim satu kali, setiap 5 menit, setiap hari, sampai dengan setiap tahun)	√	√	√	√	√	√	√	√	√	√

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) Shaperstu dio	(9) Liuge, Good App	(10) Casey Vockrodt, Destrcution Labs
Terdapat notifikasi ketika pesan dikirim	√	√	√	√	√	√	√	√	√	√
Terdapat keterangan berhasil dikirim atau gagal	√	√	x	x	√	-	x	x	√	x
Dapat melakukan pengiriman ulang untuk schedule yang gagal.	x	√	x	x	√	-	x	x	x	x
Terdapat history	√	x	x	√	√	-	x	x	√	√
Menjalankan fungsi schedule	√	√	√	√	√	x	√	√	√	√
Jadwal yang sudah dibuat dapat diedit atau dihapus	√	√	√	√	√	√	√	√	√	√

Keterangan :

√ : Ya

x : Tidak

- : Tidak Jelas

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

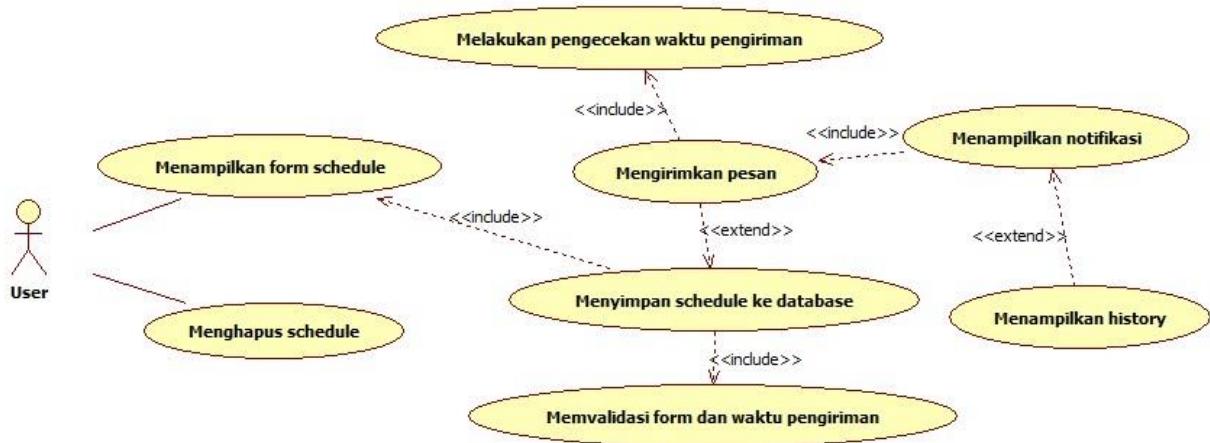
Fungsi	(1) Lylynx.com	(2) cuubonandroid	(3) ZYXSolutions	(4) Thein Min Naing	(5) DiorDNA Apps	(6) Globalapps R	(7) VinSol	(8) Shaperstu dio	(9) Liuge, Good App	(10) Casey Vockrodt, Destruction Labs
Mengkategorikan pesan (<i>schedule, sent, failed, dll</i>)	x	x	x	x	✓	x	✓	x	x	x
Menyediakan <i>template</i> pesan	x	x	x	✓	x	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 9.



Gambar 10 . Use case diagram current system SMS Scheduler

Aktor-aktor yang terlibat dalam *use case diagram* diatas dijelaskan pada tabel 4.

Tabel 6. Deskripsi aktor

No	Aktor	Deskripsi
A01	User	User merupakan orang yang akan mengirimkan pesan. User juga merupakan orang yang akan mengatur kapan dan kemana pesan akan dikirimkan

Deskripsi dari masing-masing *use case* yang ada pada *use case diagram* diatas dijelaskan pada tabel 5.

Tabel 7. Deskripsi case

No	Case	Deskripsi
UC01	Menampilkan form schedule	Case yang digunakan untuk menampilkan form ketika user membuat <i>schedule</i> pengiriman pesan yang baru atau ketika user mengubah <i>schedule</i> yang sebelumnya sudah dibuat.
UC02	Memvalidasi form dan waktu pengiriman	Case ini digunakan untuk melakukan validasi <i>form</i> dan waktu pengiriman pesan dari <i>schedule</i> yang dibuat, sebelum dimasukkan ke <i>database</i> .
UC03	Menyimpan schedule ke database	Case ini digunakan untuk menyimpan <i>schedule</i> baru atau mengupdate <i>schedule</i> yang sebelumnya telah dibuat ke <i>database</i> .
UC04	Melakukan pengecekan waktu pengiriman	Case ini digunakan untuk melakukan pengecekan waktu <i>schedule</i> dan waktu sistem.
UC05	Mengirimkan pesan	Case ini digunakan untuk mengirimkan pesan sesuai urutan yang dihasilkan proses pengecekan waktu pengiriman
UC06	Menghapus schedule	Case yang digunakan ketika user menghapus <i>schedule</i> pengiriman pesan yang sudah dibuat sebelumnya.
UC07	Menampilkan notifikasi	Case ini akan menampilkan notifikasi ketika proses pengiriman pesan dilakukan atau sedang berlangsung.
UC08	Menampilkan history	Case ini digunakan untuk menampilkan laporan pengiriman jadwal yang telah dilakukan, baik pengiriman jadwal yang berhasil maupun yang gagal.

Skenario yang terdapat dalam masing-masing *case* pada *use case diagram* dijelaskan pada tabel 7 sampai dengan tabel 20.

Tabel 8. Skenario *case* menampilkan *form schedule*

Case ID	1	
Nama Case	Menampilkan <i>form schedule</i>	
Aktor yang berperan	<i>User</i>	
Deskripsi	Sistem menampilkan <i>form</i> untuk membuat <i>schedule</i> baru atau mengubah <i>schedule</i> yang sudah ada.	
Skenario Normal	Aksi	Reaksi
	Jika <i>user</i> memilih menu untuk membuat <i>schedule</i> baru	Sistem menampilkan <i>form</i> untuk membuat <i>schedule</i> baru yang terdiri dari data <i>recipient, date and time, sending frequency</i> dan <i>message</i> .
	Jika <i>user</i> memilih menu untuk mengubah <i>schedule</i> yang sudah ada	Sistem menampilkan <i>form</i> yang sudah berisi informasi <i>schedule</i> pengiriman pesan yang dipilih <i>user</i> untuk dilakukan perubahan data.
	<i>User</i> mengisi <i>form</i> dan melakukan konfirmasi pembuatan atau perubahan <i>schedule</i> baru.	Sistem menerima masukan dari <i>user</i> berupa data-data yang akan disimpan atau <i>di-update</i> ke <i>database</i> .
	Skenario Alternatif I	<i>User</i> memilih <i>template</i> untuk mengisi <i>content message</i>
	Skenario Alternatif II	<i>User</i> membatalkan pemilihan menu untuk membuat <i>schedule</i> baru atau untuk mengubah <i>schedule</i> (menekan tombol <i>back</i> pada <i>device</i>)

Tabel 9. Skenario *case* memvalidasi *form* dan waktu pengiriman

Case ID	2	
Nama Case	Memvalidasi <i>form</i> dan waktu pengiriman	
Aktor yang berperan	<i>User</i>	
Deskripsi	Sistem melakukan validasi <i>form</i> serta waktu pengiriman jadwal yang diatur <i>user</i> sebelum menyimpan ke <i>database</i>	
Skenario Normal	Aksi	Reaksi
	<i>User</i> memasukkan <i>form</i> yang sudah lengkap dan dengan waktu pengiriman yang bukan merupakan waktu lampau pada <i>form new schedule</i> atau <i>edit schedule</i> .	
	<i>User</i> mengkonfirmasi pembuatan <i>schedule</i> atau perubahan <i>schedule</i> .	Sistem melakukan validasi pada setiap data yang diinputkan <i>user</i> untuk memastikan bahwa tidak ada data yang masih kosong.
		Sistem melakukan validasi pada waktu pengiriman untuk memastikan waktu yang diinputkan bukan waktu yang lampau.

Tabel 10. Skenario *case* memvalidasi *form* dan waktu pengiriman (Lanjutan)

Case ID	2	
Nama Case	Memvalidasi <i>form</i> dan waktu pengiriman	
Skenario Normal	Aksi	Reaksi
		Sistem mengirimkan <i>form</i> yang sudah terisi data-data yang <i>valid</i> ke <i>database</i> untuk selanjutnya dilakukan proses penyimpanan <i>schedule</i> .
Skenario Alternatif I <i>(User</i> menginputkan <i>form</i> yang belum lengkap)	<i>User</i> memasukkan <i>form</i> yang belum lengkap pada <i>form new schedule</i> atau <i>edit schedule</i> .	
	<i>User</i> mengkonfirmasi pembuatan <i>schedule</i> atau perubahan <i>schedule</i> .	Sistem melakukan validasi pada setiap data yang diinputkan <i>user</i> untuk memastikan bahwa tidak ada data yang masih kosong.
		Sistem menampilkan pesan <i>error</i> dan meminta <i>user</i> untuk melengkapi data yang masih kosong.
Skenario Alternatif II <i>(User</i> menginputkan waktu pengiriman yang sudah lampau)	<i>User</i> memasukkan <i>form</i> yang sudah lengkap, namun dengan waktu pengiriman yang merupakan waktu lampau pada <i>form new schedule</i> atau <i>edit schedule</i> .	
	<i>User</i> mengkonfirmasi pembuatan <i>schedule</i> atau perubahan <i>schedule</i> .	Sistem melakukan validasi pada setiap data yang diinputkan <i>user</i> untuk memastikan bahwa tidak ada data yang masih kosong.
		Sistem melakukan validasi pada waktu pengiriman untuk memastikan waktu yang diinputkan bukan waktu yang lampau.
Skenario Alternatif II <i>(User</i> menginputkan waktu pengiriman yang sudah lampau)		Sistem menampilkan pesan <i>error</i> dan meminta <i>user</i> untuk mengganti waktu pengiriman.
Exception <i>(Device mati atau hang ketika user mengkonfirmasi pembuatan schedule atau perubahan schedule)</i>		Sistem membatalkan proses pembuatan <i>schedule</i> atau perubahan <i>schedule</i> .

Tabel 11. Skenario case menyimpan *schedule* ke *database*

Case ID	3	
Nama Case	Menyimpan <i>schedule</i> ke <i>database</i>	
Aktor yang berperan	<i>User</i>	
Deskripsi	Sistem menyimpan data untuk <i>schedule</i> baru atau data untuk <i>schedule</i> yang diubah oleh <i>user</i> ke <i>database</i> .	
Skenario Normal	Aksi	Reaksi
	<i>User</i> membuat jadwal baru atau mengubah jadwal yang telah ada	Sistem melakukan validasi untuk setiap data yang diinputkan <i>user</i> .
		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> .
		<i>Form</i> yang sudah lengkap dan dengan waktu pengiriman yang valid disimpan ke tabel <i>schedule</i> yang ada di <i>database</i> .
Exception <i>(Device mati atau hang ketika sistem akan menyimpan <i>schedule</i> ke <i>database</i>)</i>		Sistem membatalkan proses penyimpanan <i>schedule</i> ke <i>database</i> .

Tabel 12. Skenario case melakukan pengecekan waktu pengiriman

Case ID	4	
Nama Case	Melakukan pengecekan waktu pengiriman	
Aktor yang berperan	<i>User</i>	
Deskripsi	Sistem melakukan pengecekan waktu pengiriman pada <i>schedule</i> yang sudah tersimpan di <i>database</i> untuk kemudian diurutkan sesuai dengan waktu pengiriman dan waktu ketika menyimpan <i>schedule</i> .	
Skenario Normal	Aksi	Reaksi
	<i>User</i> membuat jadwal baru atau mengubah jadwal yang telah ada	Sistem melakukan proses validasi dan penyimpanan jadwal ke tabel <i>schedule</i> di <i>database</i>
		Sistem melakukan pengecekan waktu pengiriman antara waktu yang ada pada <i>schedule</i> dengan waktu sistem (waktu <i>device</i>).
Skenario Alternatif I <i>(Belum ditemukan waktu <i>schedule</i> yang sama dengan waktu sistem)</i>		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 II <i>(Terdapat beberapa <i>schedule</i> yang harus dilaksanakan pada waktu yang sama)</i>		Jika pada saat pengecekan sistem menemukan beberapa <i>schedule</i> yang harus dilaksanakan pada waktu yang sama, sistem akan melakukan pengecekan berdasarkan waktu penyimpanannya.

Tabel 13. Skenario *case* mengirimkan pesan

Case ID	5	
Nama Case	Mengirimkan pesan	
Aktor yang berperan	<i>User</i>	
Deskripsi	Sistem melakukan pengiriman pesan sesuai hasil pengecekan waktu pengiriman yang sudah terurut.	
Skenario Normal	Aksi	Reaksi
	<i>User</i> membuat jadwal baru atau mengubah jadwal yang telah ada	Sistem melakukan proses validasi dan penyimpanan jadwal ke tabel <i>schedule</i> di <i>database</i>
		Sistem melakukan pengecekan waktu antara waktu yang ada pada <i>schedule</i> dengan waktu sistem (waktu <i>device</i>).
		Ketika waktu pada <i>schedule</i> sama dengan waktu sistem, sistem mengambil data-data yang terkait dengan <i>schedule</i> tersebut
		Sistem mengirimkan pesan sesuai dengan data-data yang terkait dengan <i>schedule</i> , dimana waktu <i>schedule</i> tersebut sudah sama dengan waktu sistem.
Skenario Alternatif (Ketika sedang melakukan pengiriman, waktu sistem sudah maju ke waktu berikutnya, sistem menyelesaikan pengiriman <i>schedule</i> yang belum selesai tersebut terlebih dahulu)		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 II (Terdapat beberapa <i>schedule</i> yang harus dilaksanakan di waktu yang sama)		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 I (Pulsa habis ketika akan melakukan pengiriman atau sedang melakukan pengiriman)		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 II (Sinyal tiba-tiba tidak tersedia ketika akan melakukan pengiriman atau sedang melakukan pengiriman)		Jika sistem akan melakukan pengiriman atau sedang melakukan pengiriman dan sinyal tidak tersedia, maka pesan tidak dikirimkan dan <i>schedule</i> dianggap gagal.
Exception III (<i>Device</i> mati atau <i>hang</i> ketika akan melakukan pengiriman atau sedang melakukan pengiriman)		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 14. Skenario *case* menghapus *schedule*

Case ID	6	
Nama Case	Menghapus <i>schedule</i>	
Aktor yang berperan	<i>User</i>	
Deskripsi	<i>User</i> menghapus <i>schedule</i> yang telah dibuat	
Skenario Normal	Aksi	Reaksi
	<i>User</i> memilih <i>schedule</i> yang akan dihapus	
	<i>User</i> memilih menu untuk menghapus <i>schedule</i> yang dipilih	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 <i>(User menekan tombol back pada device)</i>		Pada <i>database</i> , dilakukan penghapusan data yang terkait dengan <i>schedule</i> yang dipilih untuk dihapus.
	<i>User</i> membatalkan pemilihan menu untuk menghapus <i>schedule</i> (menekan tombol <i>back</i> pada <i>device</i>)	Sistem menampilkan halaman <i>list schedule</i>
Exception <i>(Device mati atau hang ketika sistem akan menghapus <i>schedule</i> ke <i>database</i>)</i>		Sistem membatalkan proses penghapusan <i>schedule</i> ke <i>database</i> .

Tabel 15. Skenario *case* menampilkan notifikasi

Case ID	7	
Nama Case	Menampilkan notifikasi	
Aktor yang berperan	<i>User</i>	
Deskripsi	Sistem menampilkan notifikasi ketika melakukan pengiriman pesan	
Skenario Normal	Aksi	Reaksi
	<i>User</i> membuat jadwal baru atau mengubah jadwal yang telah ada	Sistem menjalankan proses pengecekan dan mengirimkan pesan sesuai dengan jadwal yang telah disimpan pada <i>database</i> .
		Sistem menampilkan notifikasi pengiriman pesan.

Tabel 16. Skenario *case* menampilkan *history*

Case ID	8	
Nama Case	Menampilkan <i>history</i>	
Aktor yang berperan	<i>User</i>	
Deskripsi	Sistem menampilkan <i>history</i> pengiriman pesan yang sudah dilakukan.	
Skenario Normal	Aksi	Reaksi
	<i>User</i> membuka notifikasi pengiriman pesan atau memilih menu <i>history</i>	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, blackberry, 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 secara *native*.

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*. Sebetulnya 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 Pembuatan Aplikasi SMS *Scheduler*

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 17. 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 pada *increment* pertama, didapatkan hasil bahwa dari aplikasi yang ada saat ini (*current system*) masih terdapat beberapa masalah yang belum ditangani. Masalah-masalah tersebut 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, nomor penerima pesan dapat diambil dari kontak *device*, hanya saja jika ingin mengirim ke lebih dari satu nomor harus dilakukan

satu per satu, yaitu dengan kembali memilih menu “tambah kontak”. 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.

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

Bab ini menjelaskan tentang rancangan aplikasi SMS *Scheduler* yang akan dibuat. Pada bab ini akan dijelaskan mengenai perancangan data, perancangan arsitektur dan perancangan *user interface*. Perancangan data bertujuan untuk mengetahui kebutuhan data dan informasi yang diperlukan pada aplikasi yang akan dibuat dan direpresentasikan dengan menggunakan *Physical Data Model*, perancangan arsitektur dengan menggunakan *package diagram* (pengelompokan *class-class* yang digunakan pada aplikasi), *class diagram* (menggambarkan keterkaitan antar *class*), *sequence diagram* (menggambarkan rangkaian method yang dikirim antar objek beserta interaksinya) dan *activity diagram* (menggambarkan aliran aktivitas).

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*. Model dirancang berdasarkan kebutuhan data yang telah dijelaskan pada dokumen SRS bagian Kebutuhan Data (*Logical Database Requirements*). Perancangan data dilakukan pada tahap perancangan (*design*) di *increment* pertama dan kedua. *Detail* dari masing-masing *entity* yang ada pada E-R diagram (lihat gambar 11 pada SRS) dijelaskan pada tabel 21 sampai tabel 24.

Tabel 18. Kamus data

Nama Atribut	Tipe Data	Length	Contoh Data	Deskripsi
<i>Message_id</i>	Long int	-	1 2 3 4 5 6 n	Digunakan sebagai <i>id schedule</i> dan penomorannya bersifat <i>sequencing</i> .
<i>Message_message</i>	Varchar	160	Percobaan aplikasi <i>Scheduler</i>SMS	Digunakan untuk menampung isi pesan yang akan dikirim. Panjang isi pesan max 160 karakter (1 halaman).

Tabel 19. Kamus data (Lanjutan)

Nama Atribut	Tipe Data	Length	Contoh Data	Deskripsi
<i>Message_type</i>	Varchar	10	normal typical	Digunakan untuk menampung tipe <i>message</i>
<i>Message_timedate</i>	Date&Time	-	20130406	Digunakan untuk menampung data tanggal dan waktu pengiriman pesan.
<i>Message_status</i>	Varchar	10	Scheduled Sent Failed Paused	Digunakan untuk menampung status pengiriman pesan.
<i>Message_song</i>	Varchar	1000	Song.mp3	Digunakan untuk menampung data nada dering.
<i>Message_alert</i>	Varchar	1000		Digunakan untuk menampung informasi dari <i>schedule</i> yang akan dikirim
<i>Message_timemillist</i>	Varchar	30	12345235567678457646548587	Digunakan untuk menampung data waktu yang sudah dikonversi menjadi <i>time millis</i> .
<i>Message_freq</i>	Varchar	10	Once Hourly Weekly Yearly	Digunakan untuk menampung data frekuensi pengiriman pesan.
<i>Recipient_id</i>	Long int	-	1 2 3 4 5 6 ... n	Digunakan sebagai <i>id template</i> dan penomorannya bersifat <i>sequencing..</i>
<i>Recipient_number</i>	Varchar	20	08997937500	Digunakan untuk menampung nomor <i>recipient</i> .
<i>Recipient_status</i>	Varchar	10	Sent Failed	Digunakan untuk menampung status pengiriman untuk masing-masing <i>recipient</i> .
<i>Timemillis</i>	Varchar	30	12345235567678457646548587	Digunakan untuk menampung data waktu yang sudah dikonversi menjadi <i>time millis</i> .
<i>Time_status</i>	Varchar	10	Scheduled Sent Failed Paused	Digunakan untuk menampung status pengiriman untuk masing-masing waktu pengiriman.

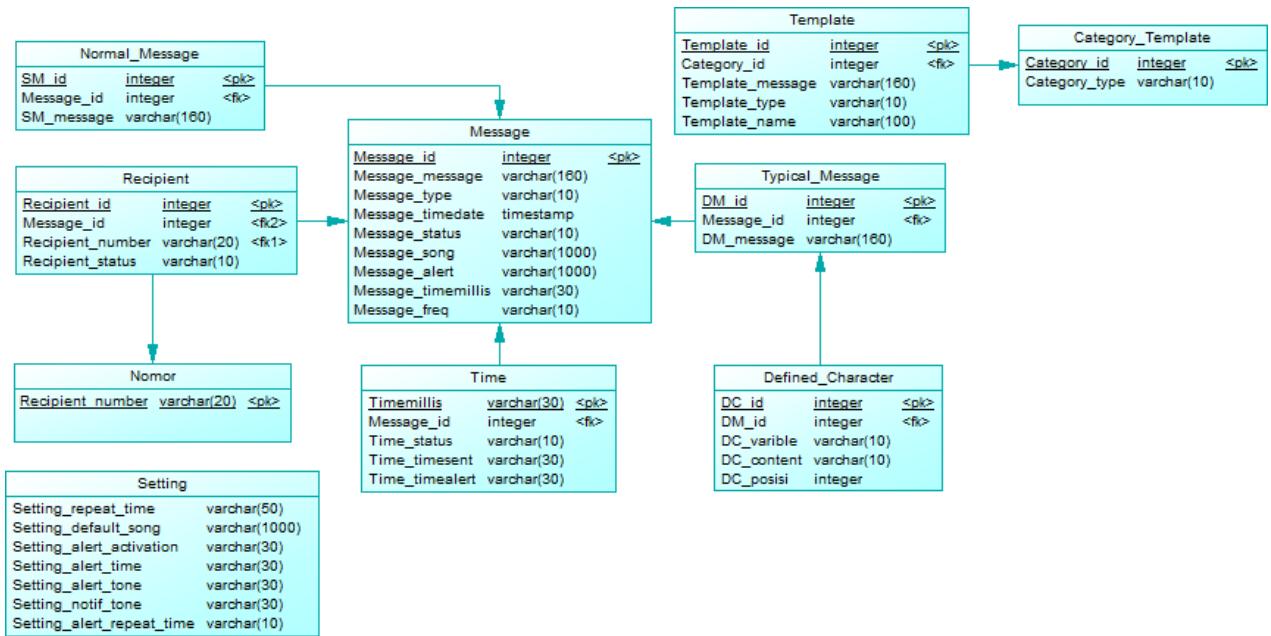
Tabel 20. Kamus data (Lanjutan)

Nama Atribut	Tipe Data	Length	Contoh Data	Deskripsi
<i>Time_timesent</i>	Varchar	30	12345235567678457646	Digunakan untuk menampung waktu pengiriman
<i>Time_timealert</i>	Varchar	30	12345235567678457646	Digunakan untuk menampung waktu untuk menampilkan alert
<i>Recipient_number</i>	Varchar	20	083452678374, 62834574839,....	Digunakan untuk menampung nomor penerima pesan
<i>DM_id</i>	Int	-	1 2 3 4 5 6 n	Digunakan sebagai <i>id typical_message</i> dan penomorannya bersifat <i>sequencing</i> .
<i>DM_message</i>	Varchar	160	Percobaan aplikasi SMS Scheduler tanggal 21 Mei 2013.....	Digunakan untuk menampung isi pesan yang akan dikirim. Panjang isi pesan max 160 karakter (1 halaman).
<i>SM_id</i>	Int	-	1 2 3 4 5 6 n	Digunakan sebagai <i>id static_message</i> dan penomorannya bersifat <i>sequencing</i> .
<i>SM_message</i>	Varchar	160	Percobaan aplikasi Scheduler.....	Digunakan untuk menampung isi pesan yang akan dikirim. Panjang isi pesan max 160 karakter (1 halaman).
<i>DC_id</i>	Int	-	1 2 3 4 5 6 n	Digunakan sebagai <i>id defined_character</i> dan penomorannya bersifat <i>sequencing</i> .
<i>DC_variable</i>	Varchar	10	1 2 3 n	Digunakan untuk menampung banyaknya <i>variable</i> yang akan digunakan.
<i>DC_content</i>	Varchar	10	2010 21 13 Agustus 2013 n	Digunakan untuk menampung isi <i>variable</i> .

Tabel 21. Kamus data (Lanjutan)

Nama Atribut	Tipe Data	Length	Contoh Data	Deskripsi
<i>DC_position</i>	Int	-	5 2 10 ... n	Digunakan untuk menampung posisi <i>variable</i> pada isi pesan
<i>Template_id</i>	Long int	-	1 2 3 4 5 6 ... n	Digunakan sebagai <i>id template</i> dan penomorannya bersifat <i>sequencing</i> .
<i>Template_message</i>	Varchar	160	<i>Happy birthday</i>	Digunakan untuk menampung isi <i>template</i> yang akan digunakan ketika mengisi isi pesan. Panjang isi <i>template</i> max 160 karakter (1 halaman).
<i>Template_type</i>	Varchar	10	birthday anniversary other	Digunakan untuk menampung tipe/kategori <i>template</i> .
<i>Template_name</i>	Varchar	100	Birthday A	Digunakan untuk menampung nama <i>template</i> .
<i>Category_id</i>	Int	-	1 2 3 4 5 6 ... n	Digunakan sebagai <i>id category_template</i> dan penomorannya bersifat <i>sequencing</i> .
<i>Category_type</i>	Varchar	10	birthday anniversary other	Digunakan untuk menampung nama tipe/kategori untuk <i>template</i> .
<i>Setting_repeat_time</i>	Varchar	50	5 10 15 20 25	Digunakan untuk menampung waktu pengulangan
<i>Setting_default_song</i>	Varchar	1000	Song.mp3	Digunakan untuk menampung nada dering yang digunakan
<i>Setting_alert_activation</i>	Varchar	30	ON OFF	Digunakan untuk menampung aktivasi <i>alert</i>
<i>Setting_alert_time</i>	Varchar	30	5 minute 10 minute 15 minute 20 minute 25 minute	Digunakan untuk menampung waktu untuk menampilkan <i>alert</i>
<i>Setting_alert_tone</i>	Varchar	30	Song.mp3	Digunakan untuk menampung nada dering <i>alert</i>
<i>Setting_notif_tone</i>	Varchar	30	Song.mp3	Digunakan untuk menampung nada dering <i>notification</i>

Berikut ini merupakan model data yang didapatkan dari E-R Diagram yang sebelumnya telah dibuat pada SRS.

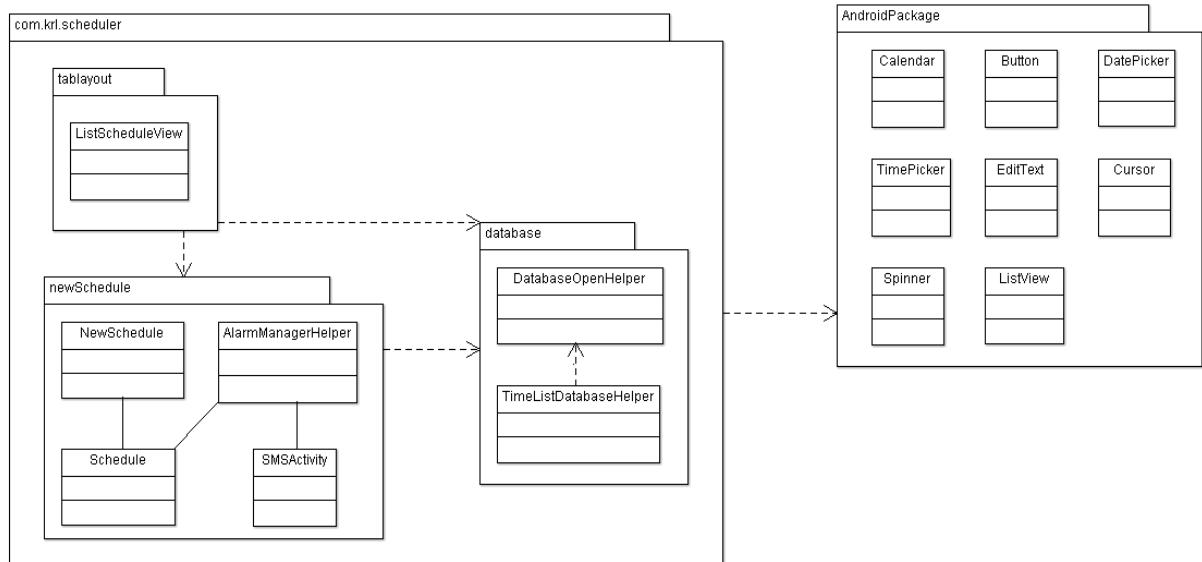


Gambar 11 . Physical Data Model (PDM) Database SMS Scheduler

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 12 menggambarkan pengelompokan kelas-kelas dan hubungan antar *package* menggunakan *tools package* diagram.



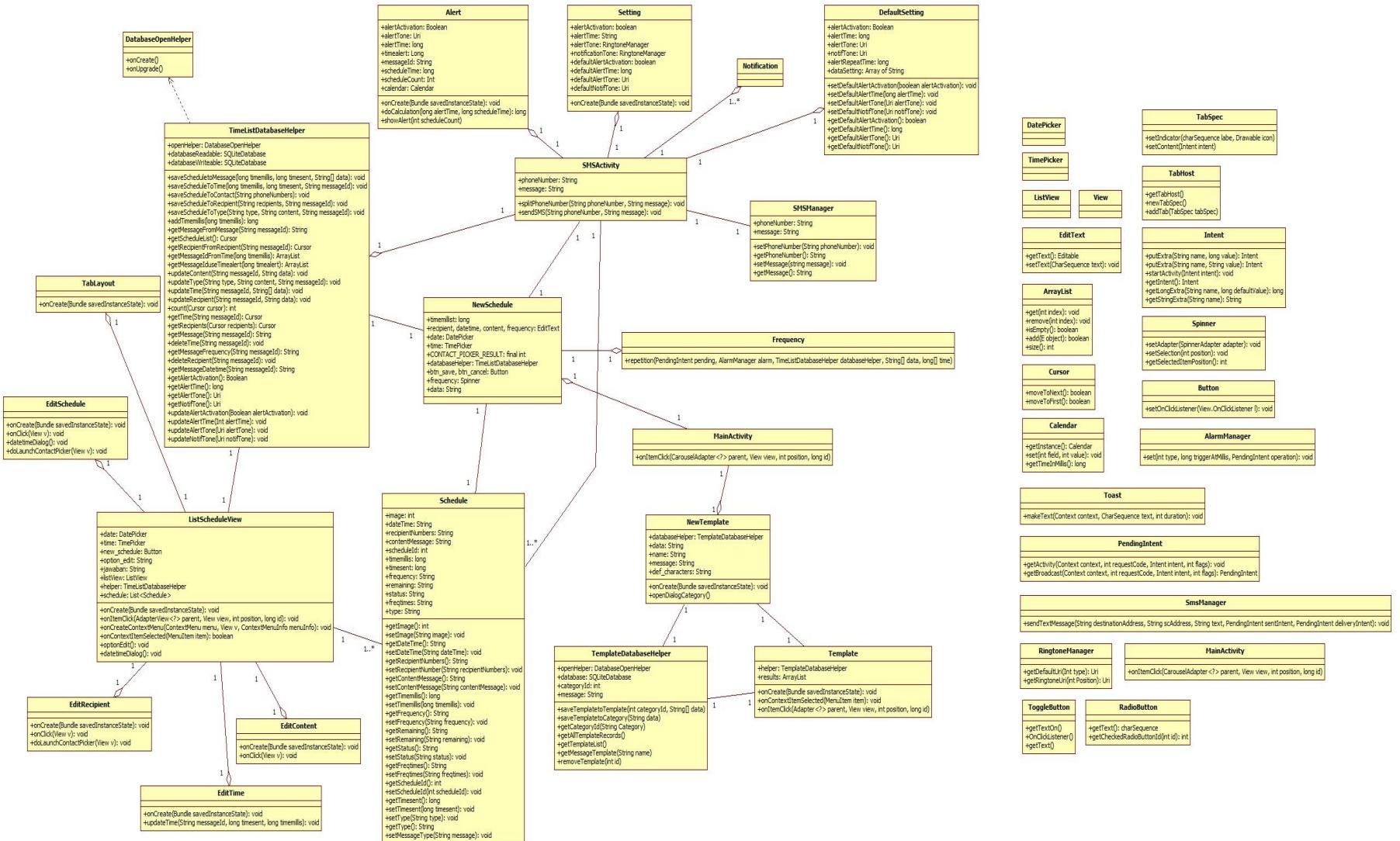
Gambar 12 . Package diagram aplikasi SMS Scheduler

Penjelasan dari *package diagram* diatas, dijelaskan pada tabel 22.

Tabel 22. Keterangan *Package Diagram*

Nama Package	Deskripsi
Package com.krl.scheduler.tablayout	<i>Package</i> ini berfungsi untuk menampung <i>class</i> yang berfungsi menampung data yang berbentuk <i>tab</i> .
Package com.krl.scheduler.newSchedule	<i>Package</i> 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
Package com.krl.scheduler.database	<i>Package</i> 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

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



Gambar 13 . *Class diagram SMS Scheduler*

Berikut ini merupakan penjelasan dari masing-masing *class* yang ada pada gambar *class diagram*.

Tabel 23. Keterangan *Class Schedule*

Class Schedule				
<i>Class schedule</i> , sebagai penampung data jadwal.				
No	Nama Attribute	Tipe	Deskripsi	Nama Method
1.	image	String	Untuk menampung data image/gambar.	1. getImage() 2. setImage(String image) 3. getDate()
2.	dateTime	String	Untuk menampung data waktu dan tanggal.	4. setDate(string dateTime) 5. getRecipientNumbers() 6. setRecipientNumber(String recipientNumbers)
3.	recipientNumbers	String	Untuk menampung data kontak penerima pesan dalam satu schedule.	7. getContentMessage() 8. setContentMessage(String contentMessage) 9. getTimemillis()
4.	contentMessage	String	Untuk menampung data pesan.	10. setTimemillis(long timemillis) 11. getFrequency() 12. setFrequency(String frequency) 13. getRemaining()
5.	scheduleId	Int	Untuk menampung id schedule	14. setRemaining(String remaining) 15. getStatus() 16. setStatus(String status) 17. getFreqtimes()
6.	timemillis	Long	untuk menampung data waktu dan tanggal yang telah diubah menjadi millisecond.	18. setFreqtimes(String freqtimes) 19. getScheduleId() 20. setScheduleId(int scheduleId) 21. getTimesent() 22. setTimesent(long timesent) 23. setType(String type) 24. getType() 25. setMessageType(String message)
7.	timesent	Long	Untuk menampung waktu pengiriman yang sudah diconvert ke dalam milisecond	
8.	frequency	String	Untuk menampung pengulangan pengiriman schedule	
9.	status	String	Untuk menampung status schedule	
11.	freqtime	String	Untuk menampung banyaknya sisa schedule yang harus dikirimkan	
12.	type	String	Untuk menampung tipe message (<i>normal</i> atau <i>typical message</i>)	

Tabel 24. Keterangan *Class ListScheduleView*

Class ListScheduleView				
<i>Class ListScheduleView</i> ini berfungsi untuk menampung jadwal yang kemudian akan ditampilkan dalam bentuk list.				
No	Nama Attribute	Tipe	Deskripsi	Nama Method
1.	date	DatePicker	untuk menampung tanggal yang dipilih.	1. onCreate(Bundle savedInstanceState) 2. onItemClick(AdapterView<?> parent, View view, int position, long id)
2.	time	TimePicker	untuk menampung waktu yang dipilih.	3. onCreateContextMenu (ContextMenu menu, View v, ContextMenuInfo menuInfo) 4. onContextItemSelected (MenuItem item) 5. optionEdit() 6. datetimeDialog()
3.	new_schedule	Button	menampung aksi untuk pembuatan jadwal baru.	
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	TimeListData baseHelper	variable instansiasi dari TimeListDatabaseHelper	

Class ListScheduleView				
No	Nama Attribute	Tipe	Deskripsi	
8.	schedule	List <Schedule>	untuk menampung data jadwal yang telah melalui proses perubahan ke dalam list.	

Tabel 25. Keterangan Class NewSchedule

Class NewSchedule				
<i>Class NewSchedule berfungsi untuk pembuatan jadwal baru.</i>				
No	Nama Attribute	Tipe	Deskripsi	Nama Method
1.	timemillis	long	untuk menampung data waktu dan tanggal yang telah diubah menjadi millisecond.	onCreate(Bundle savedInstanceState)
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_PIC KER_RESULT	final int	untuk menampung contact yang dipilih	
6.	databaseHelper	TimeListData baseHelper	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 SMSActivity

Class SMSActivity				
<i>Class ini berfungsi untuk melakukan pengiriman SMS</i>				
No	Nama Attribute	Tipe	Deskripsi	Nama Method
1.	phoneNumber	String	untuk menampung data kontak.	1. splitPhoneNumber(String phoneNumber, String message) 2. sendSMS(String phoneNumber, String message)
2.	message	String	untuk menampung data pesan atau isi pesan.	

Tabel 27. Keterangan Class SMSManager

Class SMSManager				
<i>Class ini digunakan untuk nomor penerima pesan dan isi pesan yang akan dikirimkan.</i>				
No	Nama Attribute	Tipe	Deskripsi	Nama Method
1.	phoneNumber	String	Untuk menampung nomor telepon penerima pesan	1. setPhoneNumber(String phoneNumber) 2. getPhoneNumber() 3. setMessage(string message) 4. getMessage()
2.	message	String	Untuk menampung isi pesan	

Tabel 28. Keterangan *Class TimeListDatabaseHelper*

Class TimeListDatabaseHelper				
<i>Class</i> ini digunakan sebagai penghubung untuk mengakses data-data yang sudah tersimpan di <i>database</i> , menginputkan data-data baru, maupun mengubah data-data yang sudah tersimpan di <i>database</i>				
No	Nama Attribute	Tipe	Deskripsi	Nama Method
1.	openHelper	Database OpenHelper	variabel penampung instansiasi dari DatabaseOpenHelper.	1. saveScheduleToMessage (long timemillis, long timesent, String[] data) 2. saveScheduleToTime (long timemillis, long timesent, String messageId) 3. saveScheduleToContact (String phoneNumbers) 4. saveScheduleToRecipient (Schedule schedule) 5. saveScheduleToType (Schedule schedule) 6. addTimemillis(long timemillis) 7. getMessageFromMessage (String messageId) 8. getScheduleList() 9. getRecipientFromRecipient (String messageId) 10. getMessageIdFromTime (Long timemillis) 11. getMessageIduseTimealert(long timealert) 12. updateContent(String messageId, String data) 13. updateType(String type, String content, String messageId) 14. updateTime(String messageId, String[] data) 15. updateRecipient(String messageId, String data) 16. count(Cursor cursor) 17. getTime(String messageId) 18. getRecipients(String messageId) 19. getMessage(String messageId) 20. deleteTime(String messageId) 21. getMessageFrequency (String messageId) 22. getAllTime() 23. deleteRecipient(String messageId) 24. getMessageDatetime(String messageId) 25. saveSettingToSetting(Boolean alertActivation, long alertTime, Uri alertTone, Uri notifTone) 26. getAlertActivation() 27. getAlertTime() 28. getAlertTone() 29. getNotifTone() 30. updateAlertActivation(Boolean alertActivation) 31. updateAlertTime(Int alertTime) 32. updateAlertTone(Uri alertTone) 33. updateNotifTone(Uri notifTone)
2.	database Readable	SQLite Database	Inisiasi dari kelas DatabaseOpenHelper untuk mengambil/membaca data dari <i>database</i>	
3.	database Writeable	SQLite Database	Inisiasi dari kelas DatabaseOpenHelper untuk menginput data dari <i>database</i>	

Tabel 29. Keterangan *Class DatabaseOpenHelper*

Class DatabaseOpenHelper				
<i>Class DatabaseOpenHelper</i> 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
				1. onCreate() 2. onUpgrade()

Tabel 30. Keterangan *Class Frequency*

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

Tabel 31. Keterangan *Class EditSchedule*

Class EditSchedule				
<i>Class EditSchedule</i> 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
				1. onCreate(Bundle savedInstanceState)
				2. onClick(View v)
				3. doLaunchContactPicker(View v)
				4. datetimeDialog()

Tabel 32. Keterangan *Class EditTime*

Class EditTime				
<i>Class EditTime</i> ini digunakan untuk menangani perubahan waktu pengiriman <i>schedule</i>				
No	Nama Attribute	Tipe	Deskripsi	Nama Method
				1. onCreate(Bundle savedInstanceState)
				2. updateTime(String messageId, long timesent, long timemillis)

Tabel 33. Keterangan *Class EditRecipient*

Class EditRecipient				
<i>Class EditRecipient</i> ini digunakan untuk menangani perubahan nomor penerima pesan pada <i>schedule</i>				
No	Nama Attribute	Tipe	Deskripsi	Nama Method
				1. onCreate(Bundle savedInstanceState)
				2. onClick(View v)
				3. doLaunchContactPicker(View v)

Tabel 34. Keterangan *Class EditContent*

Class EditContent				
<i>Class EditContent</i> ini digunakan untuk menangani perubahan isi pesan pada <i>schedule</i>				
No	Nama Attribute	Tipe	Deskripsi	Nama Method
				1. onCreate(Bundle savedInstanceState)
				2. onClick(View v)

Tabel 35. Keterangan *Class Alert*

Class Alert				
Class ini digunakan untuk menampilkan <i>alert</i>				
No	Nama Attribute	Tipe	Deskripsi	Nama Method
1.	alertActivation	Boolean	Untuk menampung aktivasi <i>alert</i>	1. onCreate(Bundle savedInstanceState) 2. doCalculation(Long alertTime, Cursor scheduleTime) 3. showAlert(String recipient, String message)
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 36. Keterangan *Class NewTemplate*

Class NewTemplate				
Class ini digunakan untuk menangani pembuatan <i>template</i> baru				
No	Nama Attribute	Tipe	Deskripsi	Nama Method
1.	databaseHelper	TemplateData baseHelper	Variabel penampung instansiasi dari DatabaseOpenHelper.	1. onCreate(Bundle savedInstanceState) 2. openDialogCategory()
2.	def_character	String	Untuk menampung string khusus yang diinputkan oleh <i>user</i>	

Tabel 37. Keterangan *Class TemplateDatabaseHelper*

Class TemplateDatabaseHelper				
Class ini digunakan sebagai penghubung untuk mengakses data-data yang sudah tersimpan di <i>database</i> , menginputkan data-data baru, maupun mengubah data-data yang sudah tersimpan di <i>database</i>				
No	Nama Attribute	Tipe	Deskripsi	Nama Method
1.	openHelper	Database OpenHelper	Variabel penampung instansiasi dari DatabaseOpenHelper.	1. saveTemplatetoTemplate(int categoryId, String[] data) 2. saveTemplatetoCategory(String data) 3. getCategoryID(String Category) 4. getAllTemplateRecords() 5. getTemplateList() 6. getMessageTemplate(String name) 7. removeTemplate(int id)
2.	database	SQLite Database	Inisiasi dari kelas DatabaseOpenHelper untuk mengambil/membaca data dari <i>database</i>	
3.	categoryId	Int	Untuk menampung <i>id</i> dari kategori <i>template</i>	
4.	message	String	Untuk menampung isi <i>template</i>	

Tabel 38. Keterangan *Class Template*

Class Template				
Class ini digunakan untuk menampilkan dan menghapus <i>template</i>				
No	Nama Attribute	Tipe	Deskripsi	Nama Method
1.	helper	Template Database Helper	Variabel penampung instansiasi dari DatabaseOpenHelper.	1. onCreate(Bundle savedInstanceState) 2. onContextItemSelected(MenuItem)

Class Template				
No	Nama Attribute	Tipe	Deskripsi	
2.	result	ArrayList	Untuk menampung <i>list template</i> dari <i>database</i>	item) 3. onItemClick(Adapter<?> parent, View view, int position, long id)

Tabel 39. Keterangan Class Setting

Class Setting				
No	Nama Attribute	Tipe	Deskripsi	Nama Method
1.	alertActivation	Boolean	Untuk menampung aktivasi <i>alert</i>	onCreate(Bundle savedInstanceState)
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	defaltAlertActivation	Boolean	Untuk menampung nilai default untuk aktivasi <i>alert</i>	
6	defaltAlertTime	String	Untuk menampung nilai default untuk waktu ketika menampilkan <i>alert</i>	
7	defaltAlertTone	Uri	Untuk menampung nilai default untuk nada dering yang akan digunakan saat <i>alert</i> ditampilkan	
8	defaltNotifTone	Uri	Untuk menampung nilai default untuk nada dering yang akan digunakan saat <i>notification</i> ditampilkan	

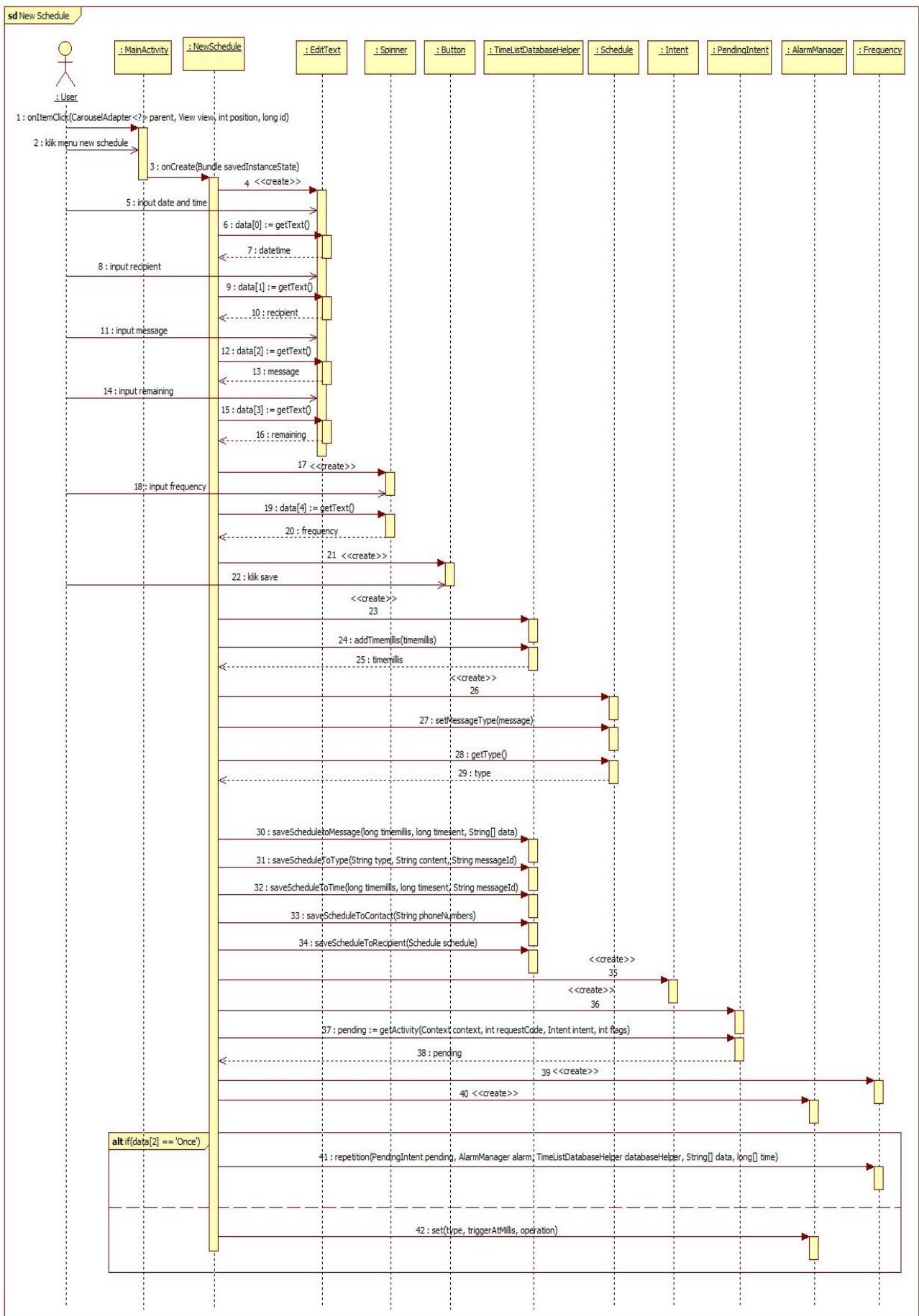
Tabel 40. Keterangan Class DefaultSetting

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

4.2.2 Perancangan Proses Aplikasi SMS *Scheduler*

Tujuan perancangan proses adalah untuk menggambarkan interaksi atau komunikasi yang terjadi diantara objek-objek dalam suatu proses pada aplikasi SMS *Scheduler*. Perancangan proses tersebut digambarkan dalam *sequence diagram*.

4.2.2.1 *Sequence Diagram* Membuat *Schedule*



Gambar 14 . Sequence diagram New Schedule

Keterangan dari *sequence diagram New Schedule* pada gambar 14, dijelaskan pada tabel 41 - 50.

Tabel 41. Keterangan *sequence diagram New Schedule*

No.	SD - 01
Requirement	[REQ F - 01], [REQ F - 01.1], [REQ CRT - 01], [REQ CRT - 02], [REQ CRT - 04], [REQ CRT - 07], [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	MainActivity, NewSchedule, EditText, Spinner, Button, TimeListDatabaseHelper, Schedule, Intent, PendingIntent, Frequency, AlarmManager
Method yang terkait	onItemClick(), onCreate(), getText(), addTimemillis(), setMessageType(), getType(), saveScheduleToMessage(), saveScheduleToType(), saveScheduleToTime(), saveScheduleToContact(), saveScheduleToRecipient(), getActivity(), repetition(), set()
Logika Proses	<pre> Var databaseHelper : TimeListDatabaseHelper timemillis, timesent : Long recipient, datetime, content, freqTime : EditText Widget btn_save, btn_cancel : Button Widget frequency : Spinner Widget freq = {"Once", "hourly", "daily", "weekly", "monthly", "yearly", "2 hourly", "4 hourly", "6 hourly", "8 hourly", "12 hourly", "2 weekly", "3 weekly", "2monthly", "4 monthly", "6 monthly"} : Array of String date_schedule : DatePicker time_schedule : TimePicker data : Array of String check : boolean freqtimes : Frequency schedule : Schedule intent : Intent pending : PendingIntent alarm : AlarmManager time : Array of long begin recipient <- new EditText(); datetime <- new EditText(); content <- new EditText(); freqTime <- new EditText(); frequency <- new Spinner(freq); data[0] <- datetime.getText(); data[1] <- recipient.getText(); data[2] <- content.getText(); data[3] <- freqTime.getText(); data[4] <- frequency.getText(); if (btn_save.clicked) then timesent <- timemillis; //melakukan pengecekan, apakah terdapat timemillis dengan value atau nilai yang sama timemillis <- databaseHelper.addTimemillis(timemillis); schedule <- new Schedule(timemillis, timesent, data); //cek tipe dari content schedule.setMessageType(schedule.contentMessages); data[7] = schedule.getType(); //melakukan insert ke database ke dalam table message databaseHelper.saveScheduleToMessage(schedule.getTimemillis(), schedule.getTimesent(), data); //melakukan insert ke database ke dalam table type databaseHelper.saveScheduleToType(data[7], data[2], schedule.getScheduleId()); //melakukan insert ke database ke dalam table time databaseHelper.saveScheduleToTime(timemillis, timesent, schedule.getScheduleId()); //melakukan insert ke database ke dalam table contact databaseHelper.saveScheduleToContact(schedule.getRecipientNumbers()); //melakukan insert ke database ke dalam table recipient intent <- new Intent(); intent.putExtra("timemillis", timesent); pending <- pending.getActivity(); alarm <- getSystemService(ALARM_SERVICE); if (data[4] == "Once") then dataRepetition[0] <- data[3]; dataRepetition[1] <- schedule.getScheduleId(); </pre>

No.	SD - 01
	<pre> dataRepetition[2] <- data[4]; time[0] <- timemillis; time[1] <- timesent; freqtimes.repetition(pending, alarm, databaseHelper, dataRepetition, time); else alarm.set(AlarmManager.RTC_WAKEUP, timesent, pending); endif endif end </pre>

Tabel 42. Spesifikasi proses addTimemillis

No.	SP - 01
Nama Method	addTimemillis
Nama Class	TimeListDatabaseHelper
Requirement	[REQ CRT - 09]
Deskripsi	<i>Method ini digunakan untuk menyimpan timemillis (yang merupakan waktu schedule yang telah di-convert ke millisecond) ke tabel time di database.</i>
Return	-
Parameter	timemillis : long
Method yang terkait	getReadableDatabase(), rawQuery()
Logika Proses	<pre> procedure addTimemillis(timemillis : long) var openHelper : TimeListDatabaseHelper //sebagai penampung inisiasi class TimeListDatabaseHelper databaseReadable : SQLiteDatabase //sebagai penampung inisiasi class SQLiteDatabase cursor : Cursor //sebagai penampung dari data yang berhasil diambil dari database begin //sebagai inisiasi method untuk melakukan query read (baca) dari database databaseReadable <- openHelper.getReadableDatabase(); //mendapatkan semua timemillis cursor <- databaseReadable.rawQuery("select time_timemillis from time"); while(cursor.moveToFirst() = true) do //bila terdapat timemillis yang sama, maka terjadi penambahan timemills if (cursor.getLong(cursor.getColumnIndex("time_timemillis")) = timemillis) then timemillis <- timemillis + 1; endif endwhile end endprocedure </pre>

Tabel 43. Spesifikasi proses setMessageType

No.	SP - 02
Nama Method	setMessageType
Nama Class	Schedule
Requirement	[REQ CRT - 10]
Deskripsi	<i>Method ini digunakan untuk mengeset tipe pesan</i>
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.contain("%")) then this.type <- "typical"; else this.type <- "normal"; endif end endprocedure </pre>

Tabel 44. Spesifikasi proses getType

No.	SP - 03
Nama Method	getType
Nama Class	Schedule
Requirement	[REQ CRT - 10]
Deskripsi	Method ini digunakan untuk mengambil tipe pesan
Return	type : String
Parameter	-
Method yang terkait	-
Logika Proses	<pre>function getType() begin return type; end endfunction</pre>

Tabel 45. Spesifikasi proses saveScheduleToMessage

No.	SP - 04
Nama Method	saveScheduleToMessage
Nama Class	TimeListDatabaseHelper
Requirement	[REQ CRT - 10]
Deskripsi	Method ini digunakan untuk menyimpan <i>schedule</i> ke tabel message di <i>database</i>
Return	-
Parameter	timemillis : long, data : Array of String
Method yang terkait	getWritableDatabase(), rawQuery(), put(), insert(), clear()
Logika Proses	<pre>procedure saveScheduleToMessage(timemillis : long, data : Array of String) var openHelper : TimeListDatabaseHelper //sebagai penampung inisiasi class TimeListDatabaseHelper databaseWriteable : SQLiteDatabase //sebagai penampung inisiasi class SQLiteDatabase contentValues : ContentValues //sebagai penampung inisiasi terhadap class ContentValues begin //sebagai inisiasi method untuk melakukan query write (tulis) ke database databaseWriteable <- openHelper.getWritableDatabase(); contentValues.put("message_timedate", data[0]); contentValues.put("message_message", data[2]); contentValues.put("message_status", data[5]); contentValues.put("message_timemillis", timemillis); contentValues.put("message_type", data[7]); contentValues.put("message_frequency", data[3]); databaseWriteable.insert("message", null, contentValues); contentValues.clear(); end endprocedure</pre>

Tabel 46. Spesifikasi proses saveScheduleToType

No.	SP - 05
Nama Method	saveScheduleToType
Nama Class	TimeListDatabaseHelper
Requirement	[REQ CRT - 10]
Deskripsi	Method ini digunakan untuk menyimpan tipe pesan dengan <i>id message</i> tertentu ke tabel message di <i>database</i>
Return	-
Parameter	type, content, messageId : string
Method yang terkait	getWritableDatabase(), rawQuery(), put(), insert(), clear()
Logika Proses	<pre>procedure saveScheduleToType(type, content, messageId : String) var //sebagai penampung inisiasi class TimeListDatabaseHelper databaseWriteable : SQLiteDatabase //sebagai penampung inisiasi class SQLiteDatabase</pre>

No.	SP - 05
Nama Method	saveScheduleToType
<pre> contentValues : ContentValues //sebagai penampung inisiasi terhadap class ContentValues begin //sebagai inisiasi method untuk melakukan query write (tulis) ke database databaseWriteable <- openHelper.getWritableDatabase(); contentValues.put("message_timedate", data[0]); contentValues.put("message_message", data[2]); contentValues.put("message_status", data[5]); contentValues.put("message_timenillis", timemillis); contentValues.put("message_type", data[7]); contentValues.put("message_frequency", data[3]); databaseWriteable.insert("message", null, contentValues); contentValues.clear(); end endprocedure </pre>	

Tabel 47. Spesifikasi proses saveScheduleToTime

No.	SP - 06
Nama Method	saveScheduleToTime
Nama Class	TimeListDatabaseHelper
Requirement	[REQ CRT - 10]
Deskripsi	Method ini digunakan untuk menyimpan waktu <i>schedule</i> dengan <i>id message</i> tertentu ke tabel time di <i>database</i>
Return	-
Parameter	timemillis, timesent : long, messageId : String
Method yang terkait	getWritableDatabase(), rawQuery(), put(), insert(), clear()
Logika Proses	<pre> procedure saveScheduleToTime(timemillis, timesent : long, messageId : String) var openHelper : TimeListDatabaseHelper//sebagai penampung inisiasi class TimeListDatabaseHelper databaseWriteable : SQLiteDatabase //sebagai penampung inisiasi class SQLiteDatabase contentValues : ContentValues //sebagai penampung inisiasi terhadap class ContentValues begin //sebagai inisiasi method untuk melakukan query write (tulis) ke database databaseWriteable <- openHelper.getWritableDatabase(); contentValues.put("time_timenillis", timemillis); contentValues.put("time_message_id", messageId); contentValues.put("time_timesent", timesent); databaseWriteable.insert("time", null, contentValues); contentValues.clear(); end endprocedure </pre>

Tabel 48. Spesifikasi proses saveScheduleToContact

No.	SP - 07
Nama Method	saveScheduleToContact
Nama Class	TimeListDatabaseHelper
Requirement	[REQ CRT - 10]
Deskripsi	Method ini digunakan untuk menyimpan <i>recipient schedule</i> ke tabel nomor di <i>database</i> .
Return	-
Parameter	phoneNumber : String
Method yang terkait	getWritableDatabase(), getReadableDatabase(), rawQuery(), put(), insert(), clear(), nextElement()
Logika Proses	<pre> procedure saveScheduleToContact(phoneNumbers : String) var openHelper : TimeListDatabaseHelper//sebagai penampung inisiasi class TimeListDatabaseHelper databaseWriteable : SQLiteDatabase //sebagai penampung inisiasi class SQLiteDatabase databaseReadable : SQLiteDatabase //sebagai penampung inisiasi class SQLiteDatabase contentValues : ContentValues //sebagai penampung inisiasi terhadap class ContentValues st : StringTokenizer //sebagai penampung inisiasi terhadap class StringTokenizer tempPhoneNumber : String //sebagai penampung nomor recipient begin </pre>

No.	SP - 07
Nama Method	<pre> databaseReadable <- openhelper.getReadableDatabase(); //sebagai inisiasi method untuk melakukan query read (baca) dari database databaseWriteable <- openHelper.getWritableDatabase(); //sebagai inisiasi method untuk melakukan query write (tulis) ke database st <- new StringTokenizer(phoneNumbers, ";"); while(st.hasMoreElements()) do tempPhoneNumber <- (String) st.nextElement(); cursor <- databaseReadable.rawQuery("select * from contact where contact_number='tempPhoneNumber'"); if(cursor.moveToFirst() == false) then contentValues.put("contact_number", tempPhoneNumber); databaseWriteable.insert("contact", null, contentValues); endif endwhile contentValues.clear(); end endprocedure </pre>

Tabel 49. Spesifikasi proses saveScheduleToRecipient

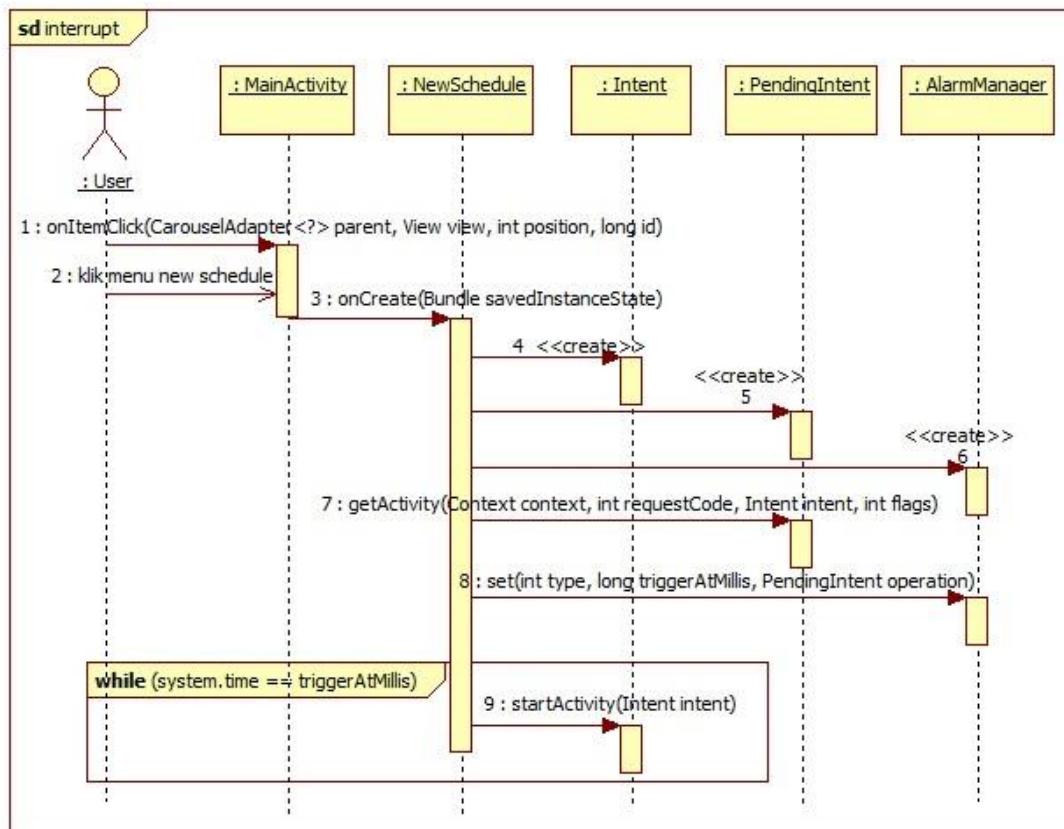
No.	SP - 08
Nama Method	saveScheduleToRecipient
Nama Class	TimeListDatabaseHelper
Requirement	[REQ CRT - 10]
Deskripsi	Method ini digunakan untuk menyimpan recipient schedule dengan id message tertentu ke tabel recipient di database
Return	-
Parameter	recipients, messageId : string
Method yang terkait	getWritableDatabase(), rawQuery(), put(), insert(), clear(), nextElement()
Logika Proses	<pre> procedure saveScheduleToRecipient(recipients, messageId : String) var openHelper : TimeListDatabaseHelper//sebagai penampung inisiasi class TimeListDatabaseHelper databaseWriteable : SQLiteDatabase //sebagai penampung inisiasi class SQLiteDatabase contentValues : ContentValues //sebagai penampung inisiasi terhadap class ContentValues st : StringTokenizer //sebagai penampung inisiasi terhadap class StringTokenizer tempPhoneNumber : String //sebagai penampung nomor recipient begin databaseWriteable <- openHelper.getWritableDatabase(); //sebagai inisiasi method untuk melakukan query write (tulis) ke database StringTokenizer st <- new StringTokenizer(recipients, ";"); while (st.hasMoreelements()) do tempPhoneNumber <- (String) st.nextElement(); contentValues.put("recipient_number", tempPhoneNumber); contentValues.put("recipient_message_id", messageId); databaseWriteable.insert("recipient", null, contentValues); contentValues.clear(); endwhile contentValues.clear(); end endprocedure </pre>

Tabel 50. Spesifikasi proses repetition

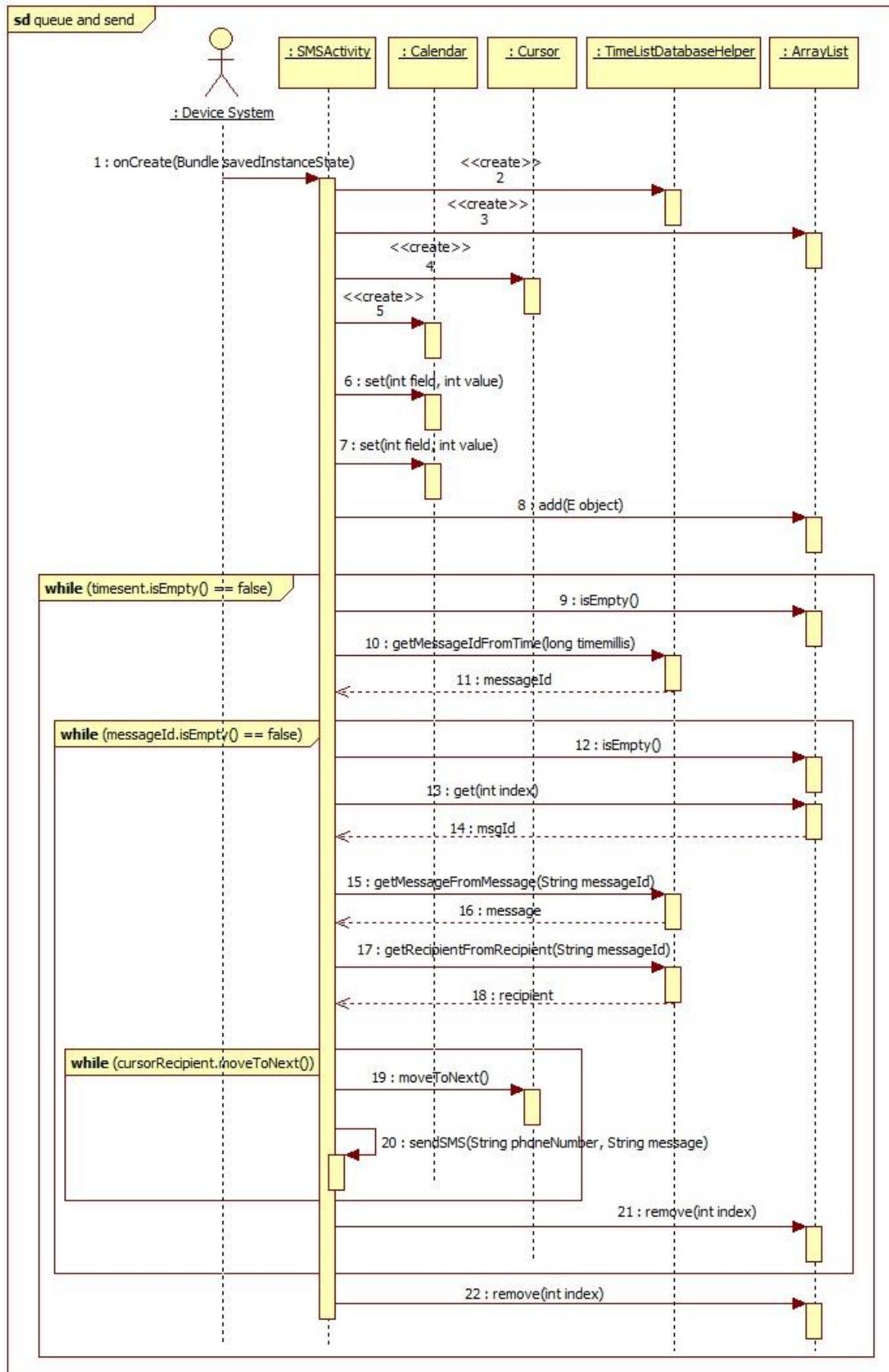
No.	SP - 09
Nama Method	repetition
Nama Class	Frequency
Requirement	[REQ CRT - 07]
Deskripsi	Method ini digunakan untuk menangani pengulangan pengiriman schedule
Return	-
Parameter	pending : PendingIntent, alarm : AlarmManager, databaseHelper : TimeListDatabaseHelper, data : Array of String, time : Array of Long
Method yang terkait	setRepeating(), parseInt(), add(), saveScheduleToTime()

No.	SP - 09
Nama Method	repetition
Logika Proses	
<pre> procedure repetition(pending : PendingIntent, alarm : AlarmManager, databaseHelper : TimeListDatabaseHelper, data : Array of String, time : Array of Long) var timemillis : long //sebagai penampung data timemillis timesent : long //sebagai penampung data timesent frequency : String //sebagai penampung data frequency messageId : String //sebagai penampung data messageId interval : long //sebagai penampung data interval begin timemillis <- time[0]; timesent <- time[1]; frequency <- data[0]; messageId <- data[1]; //memanggil method interval untuk menghitung interval waktu sesuai dengan freqencynya interval <- interval(frequency); //set ke dalam AlarmManager sebagai interrupt alarm.setRepeating(AlarmManager.RTC_WAKEUP, timesent, interval, pending); //insert ke dalam database sesuai dengan remaining yang diinputkan for(int remaining <- Integer.parseInt(data[2]) to remaining > 1) do //melakukan penambahan timemillis bila ditemukan terdapat timemillis yang sama di //database timemillis <- databaseHelper.addTimemillisFromTime(timemillis); //melakukan insert ke database ke table time databaseHelper.saveScheduleToTime(timemillis, timesent, messageId); endfor end endprocedure </pre>	

4.2.2.1.1 Sequence Diagram Send Message



Gambar 15 . Sequence diagram interrupt



Gambar 16 . Sequence diagram queue dan send message

Keterangan dari *sequence diagram queue* pada gambar 15 dan 16, dijelaskan pada tabel 51 - 56.

Tabel 51. Keterangan *sequence diagram interrupt*

No.	SD - 02
Requirement	[REQ F - 02], [REQ SND - 01]
Deskripsi	Sequence ini digunakan untuk menggambarkan proses interrupt
Class yang terkait	MainActivity, NewSchedule, Intent, PendingIntent, AlarmManager
Method yang terkait	onItemClick, onCreate(), getActivity(), set(), startActivity()
<pre> var intent : Intent //sebagai penampung inisiasi dari class Intent pendingIntent : PendingIntent //sebagai penampung inisiasi dari class PendingIntent alarm : new AlarmManager() //sebagai penampung inisiasi dari class AlarmManager timemillis : long //sebagai penampung waktu yang sudah berbentuk millisecond timesent : long //sebagai penampung waktu pengiriman yang sudah berbentuk millisecond begin intent <- new Intent(); //sebagai inisiasi dari class Intent pendingIntent <- new PendingIntent() //sebagai inisiasi dari class PendingIntent alarm <- new AlarmManager(); //sebagai inisiasi dari class AlarmManager //mendapatkan izin untuk dapat melakukan activity terhadap class tertentu intent.getActivity(this, (int) timemillis, PendingIntent.FLAG_UPDATE_CURRENT); //melakukan set terhadap class AlarmManager yang berfungsi sebagai interrupt dengan waktu interrupt adalah timesent alarm.set(AlarmManager.RTC_WAKEUP, timesent, pending); //bila waktu sistem dengan waktu timesent sama while(system.time = timesent) do //melakukan aktivitas/activity terhadap class intent lain startActivity(intent); endwhile end </pre>	

Tabel 52. Keterangan *sequence diagram queue* dan *send message*

No.	SD - 03
Requirement	[REQ F - 02], [REQ SND - 01]
Deskripsi	Sequence ini digunakan untuk menggambarkan pembuatan <i>queue</i> dan pelaksanaan <i>queue</i> tersebut
Class yang terkait	SMSActivity, TimeListDatabaseHelper, 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 timesent : Array of long //sebagai penampung waktu dalam bentuk millisecond messageId : Array of String //sebagai penampung messageId -> plural databaseHelper : TimeListDatabaseHelper //sebagai penampung inisiasi terhadap class TimeListDatabaseHelper 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(Calendar.SECOND, 0); //mengeset nilai dari millisecond calendar.set(Calendar.MILLISECOND, 0); //mengisi variabel timesent dengan waktu yang sudah berbentuk millisecond timesent.add(calendar.getTimeInMillis()); //loop sampai isi array timesent tidak ada lagi while(timesent.isEmpty() = false) do //mengambil messageId dari table time di database messageId <- databaseHelper.getMessageIdFromTime(timesent); 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); </pre>	

No.	SD - 03
	<pre>//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 timesent dengan index 0 timesent.remove(0); endwhile end</pre>

Tabel 53. Spesifikasi proses getMessageIdFromTime

No.	SP - 10
Nama Method	getMessageIdFromTime
Nama Class	TimeListDatabaseHelper
Requirement	[REQ F - 02], [REQ SND - 01]
Deskripsi	<i>Method</i> 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	timemillis : long
Method yang terkait	getReadableDatabase(), rawQuery(), add()
Logika Proses	<pre>function getMessageIdFromTime(timemillis : long) var openHelper : TimeListDatabaseHelper//sebagai penampung inisiasi class TimeListDatabaseHelper databaseReadable : SQLiteDatabase //sebagai penampung inisiasi class SQLiteDatabase cursor : Cursor, //penampung data yang diambil dari database messageId : ArrayList of String //penampung data id message yang didapat dari database begin //sebagai inisiasi method untuk melakukan query read (baca) dari database databaseReadable <- openhelper.getReadableDatabase(); //mendapatkan messageId dari table time dengan timesent tertentu cursor <- databaseReadable.rawQuery("select time_message_id from time where time_timesent='timemillis'"); while(cursor.moveToNext()) do messageId.add(cursor.getString(cursor.getColumnIndex("time_message_id"))); endwhile return messageId; end end function</pre>

Tabel 54. Spesifikasi proses getMessageFromMessage

No.	SP - 11
Nama Method	getMessageFromMessage
Nama Class	TimeListDatabaseHelper
Requirement	[REQ F - 02], [REQ SND - 01]
Deskripsi	<i>Method</i> ini digunakan untuk mengambil isi pesan dengan <i>id message</i> tertentu.
Return	message : String
Parameter	msgId : String
Method yang terkait	getReadableDatabase(), rawQuery(), moveToNext(), getString(), getColumnIndex()
Logika Proses	<pre>function getMessageFromMessage(msgId : String) var openHelper : TimeListDatabaseHelper//sebagai penampung inisiasi class TimeListDatabaseHelper databaseReadable : SQLiteDatabase //sebagai penampung inisiasi class SQLiteDatabase message : String //penampung data message atau content yang didapat dari database cursor : Cursor //penampung data yang diambil dari database</pre>

No.	SP - 11
Nama Method	getMessageFromMessage
<pre> begin message <- null; //sebagai inisiasi method untuk melakukan query read (baca) dari database databaseReadable <- openhelper.getReadableDatabase(); //mendapatkan message dari table message dengan messageId tertentu cursor <- databaseReadable.rawQuery("select message_message from message where message_id='msgId'"); if(cursor.moveToFirst() = true) then message <- cursor.getString(cursor.getColumnIndex("message_message")); endif return message; end endfunction </pre>	

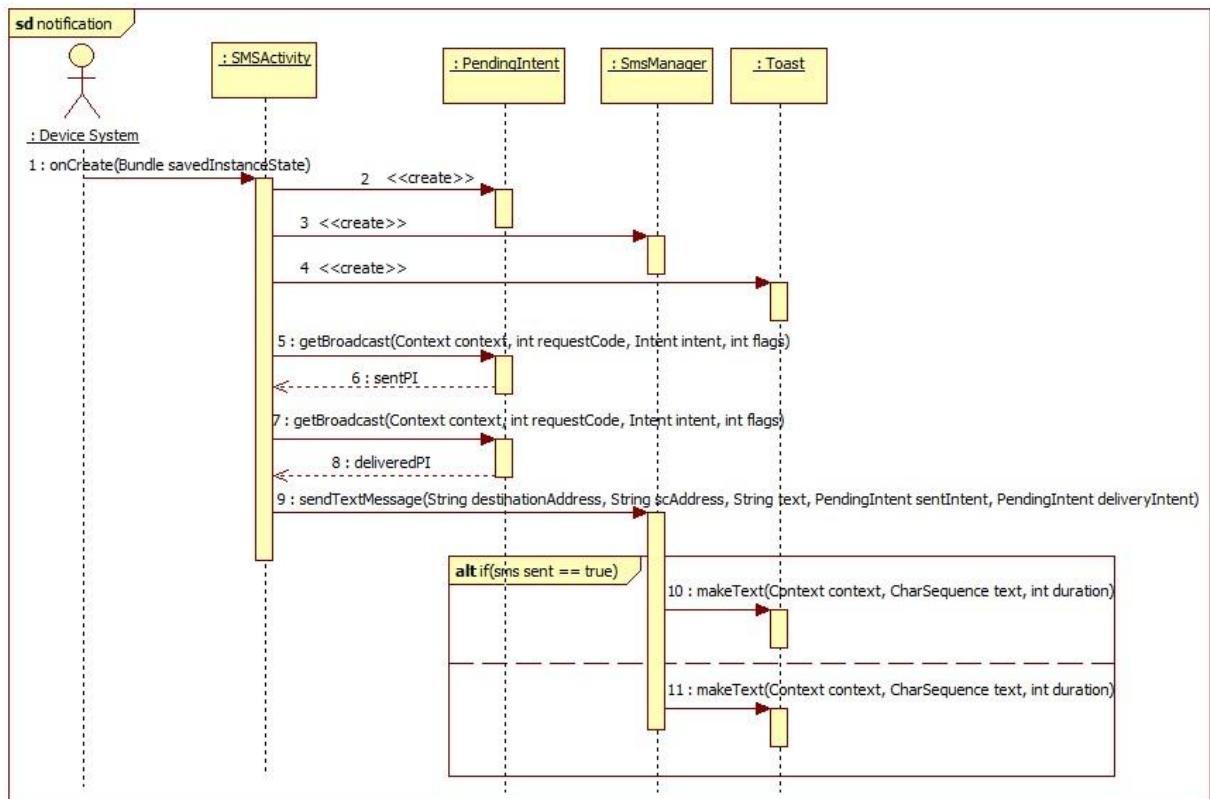
Tabel 55. Spesifikasi proses getRecipientFromRecipient

No.	SP - 12
Nama Method	getRecipientFromRecipient
Nama Class	TimeListDatabaseHelper
Requirement	[REQ F - 02], [REQ SND - 01]
Deskripsi	Method ini digunakan untuk mengambil nomor recipient dengan id message tertentu
Return	cursorRecipient : Cursor
Parameter	msgId : String
Method yang terkait	getReadableDatabase(), rawQuery()
Logika Proses <pre> function getRecipientFromRecipient(msgId : String) var openHelper : TimeListDatabaseHelper//sebagai penampung inisiasi class TimeListDatabaseHelper databaseReadable : SQLiteDatabase //sebagai penampung inisiasi class SQLiteDatabase begin //sebagai inisiasi method untuk melakukan query read (baca) dari database databaseReadable <- openhelper.getReadableDatabase(); //mendapatkan semua record recipient yang memiliki messageId tertentu return databaseReadable.rawQuery("select * from recipient_number from recipient where recipient_message_id='msgId'"; end endfunction </pre>	

Tabel 56. Spesifikasi proses sendSMS

No.	SP - 13
Nama Method	sendSMS
Nama Class	SMSActivity
Requirement	[REQ F - 02], [REQ SND - 01]
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 : PendingIntent //penampung bila sms berhasil dikirimkan deliveredPI : PendingIntent //penampung bila sms gagal dikirimkan begin //izin untuk dapat menggunakan fitur sms yang telah tersedia SmsManager sms <- SmsManager.getDefault(); //proses pengiriman sms oleh class yang telah disediakansms.sendTextMessage(phoneNumber, null, message, sentPI, deliveredPI); end endprocedure </pre>	

4.2.2.1.2 Sequence Diagram Notification



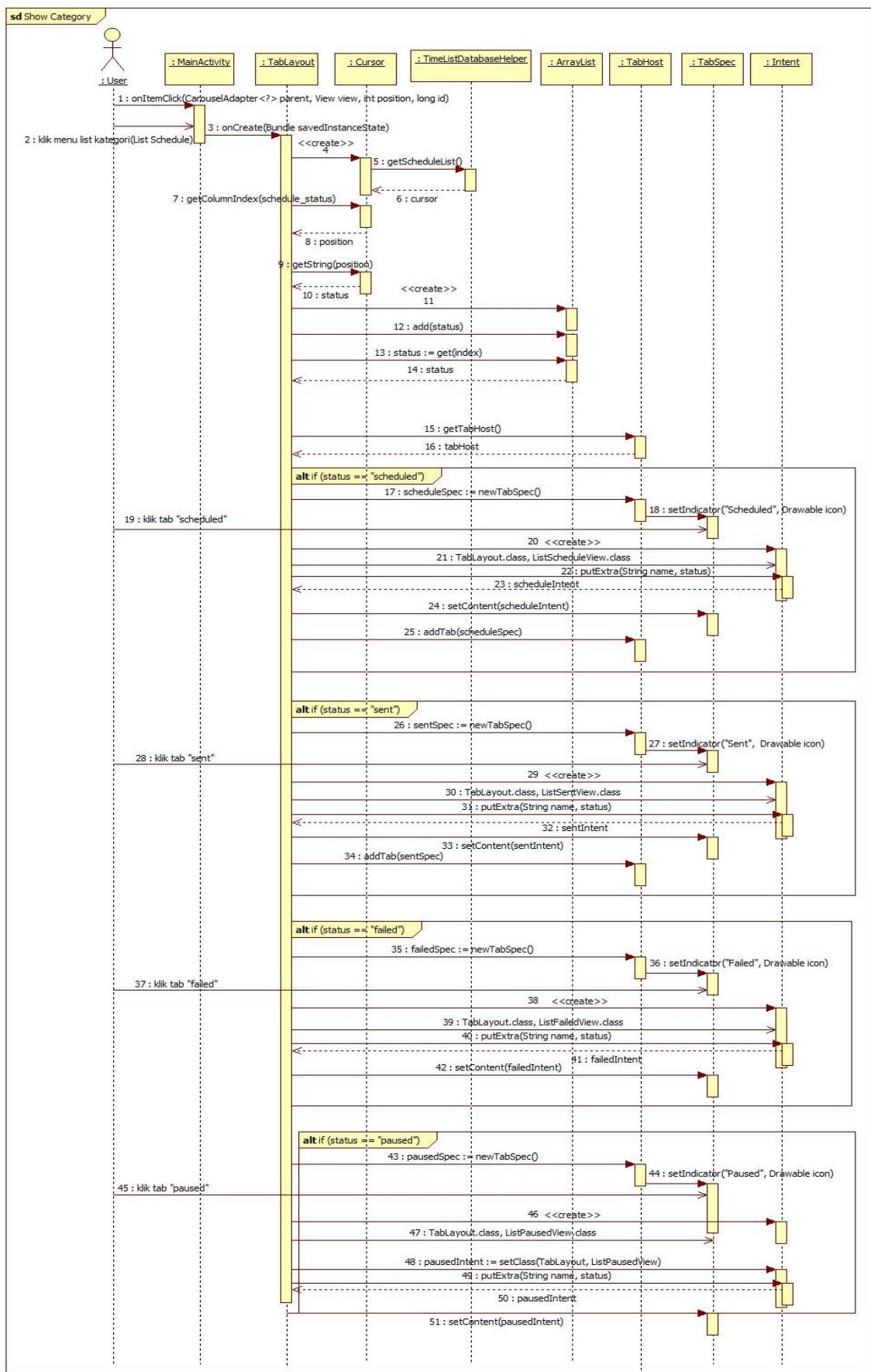
Gambar 17 . Sequence diagram notification

Keterangan dari *sequence diagram notification* pada gambar 17, dijelaskan pada tabel 57.

Tabel 57. Keterangan *sequence diagram notification*

No.	SD - 04
Requirement	[REQ F - 04] , [REQ NTF - 01], [REQ NTF - 02]
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, PendingIntent, SmsManager, Toast
Method yang terkait	getBroadcast(), sendTextMessage(), makeText()
<pre> var sentPI : PendingIntent //sebagai penampung inisiasi dari class PendingIntent deliveredPI : PendingIntent //sebagai penampung inisiasi dari class PendingIntent 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 <- PendingIntent.getBroadcast(this, 0, new Intent(SENT)); //sebagai inisiasi dari class PendingIntent deliveredPI <- PendingIntent.getBroadcast(this, 0, new Intent(DELIVERED), 0); //sebagai inisiasi dari class PendingIntent 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", Toast.LENGTH_SHORT); else toast.makeText(this, "SMS Failed", Toast.LENGTH_SHORT); endif end </pre>	

4.2.2.1.3 Sequence Diagram Menampilkan Kategori Pesan



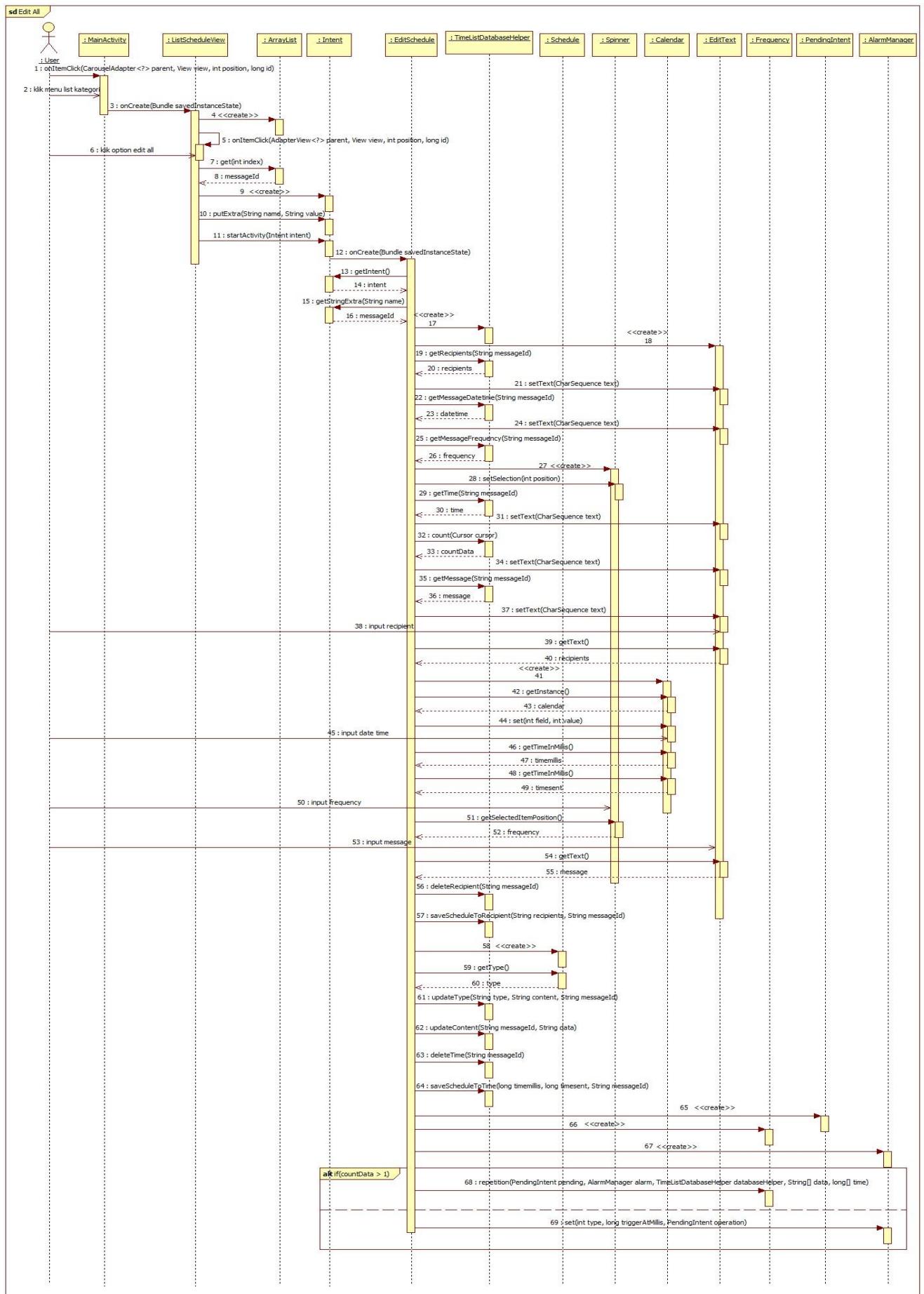
Gambar 18 . Sequence diagram Menampilkan Kategori Pesan

Keterangan dari *sequence diagram* Menampilkan Kategori Pesan pada gambar 18, dijelaskan pada tabel 58.

Tabel 58. Keterangan *sequence diagram* Menampilkan Kategori Pesan

No.	SD - 05
Requirement	[REQ F - 01.5] , [REQ FOL - 01], [REQ FOL - 02], [REQ FOL - 03], [REQ FOL - 04]
Deskripsi	<i>Sequence</i> ini digunakan untuk menggambarkan proses untuk menampilkan kelompok pesan sesuai dengan kategorinya
Class yang terkait	MainActivity, TabLayout, Cursor, TimeListDatabaseHelper, ArrayList, TabHost, TabSpec, Intent
Method yang terkait	onItemClick(), onCreate(), getScheduleList(), getColumnIndex(), getString(), add(), get(), getTabHost(), newTabSpec(), setIndicator(), putExtra(), setContent(), addTab()
Var	<pre> cursor : Cursor //sebagai penampung inisiasi class Cursor openHelper : TimeListDatabaseHelper //sebagai penampung inisiasi class TimeListDatabaseHelper schedule : ArrayList //sebagai penampung inisiasi class ArrayList position : Integer //sebagai penampung posisi schedule status : String //sebagai penampung status schedule </pre>
begin	<pre> cursor <- openHelper.getScheduleList(); position <- cursor.getColumnIndex("schedule_status"); status <- cursor.getString(position); schedule <- new ArrayList; if (cursor.moveToFirst())then do schedule.add(status); while (cursor.moveToNext()) endif status <- schedule.get(index); tabHost <- getTabHost(); CASE OF status CASE "scheduled": scheduleSpec <- tabHost.newTabSpec(); scheduleSpec.setIndicator("scheduled", drawable icon); //icon for schedule list scheduleIntent <- new Intent(TabLayout.class, ListScheduleView.class); scheduleIntent.putExtra("scheduled", status); scheduleSpec.setContent(scheduleIntent); tabHost.addTab(scheduleSpec); CASE "sent": sentSpec <- tabHost.newTabSpec(); sentSpec.setIndicator("sent", drawable icon); //icon for schedule list sentIntent <- new Intent(TabLayout.class, ListSentView.class); sentIntent.putExtra("sent", status); sentSpec.setContent(sentIntent); tabHost.addTab(sentSpec); CASE "failed": failedSpec <- tabHost.newTabSpec(); failedSpec.setIndicator("failed", drawable icon); //icon for schedule list failedIntent <- new Intent(TabLayout.class, ListFailedView.class); failedIntent.putExtra("failed", status); failedSpec.setContent(failedIntent); tabHost.addTab(failedSpec); CASE "paused": pausedSpec <- tabHost.newTabSpec(); pausedSpec.setIndicator("paused", drawable icon); //icon for schedule list pausedIntent <- new Intent(TabLayout.class, ListPausedView.class); pausedIntent.putExtra("paused", status); pausedSpec.setContent(pausedIntent); tabHost.addTab(pausedSpec); ENDCASE end </pre>

4.2.2.1.4 Sequence Diagram Edit All Schedule



Gambar 19 . Sequence diagram edit all

Keterangan dari *sequence diagram edit all schedule* pada gambar 19, dijelaskan pada tabel 59 - 72.

Tabel 59. Keterangan *sequence diagram edit all schedule*

No.	SD - 06
Requirement	[REQ F - 01.3], [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	Sequence ini menggambarkan proses untuk melakukan perubahan <i>schedule</i> secara keseluruhan (<i>edit all content schedule</i>)
Class yang terkait	MainActivity, ListScheduleView, ArrayList, Intent, EditSchedule, TimeListDatabaseHelper, Cursor, EditText, Spinner, Calendar, Schedule, Frequency, AlarmManager, PendingIntent
Method yang terkait	onItemClick(), onCreate(), get(), putExtra(), startActivity(), getIntent(), 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 : TimeListDatabaseHelper freqtimes : Frequency schedule : ArrayList messageId, type, content : String intent : Intent pending : PendingIntent lview : ListScheduleView calendar : Calendar alarm : AlarmManager recipients, datetime, message, countData, frequency, time : String timemillis, timesent : Long rcp, dt, msg, freq, cntdata : EditText Widget freq_spinner : Spinner Widget position : Integer data : Array of String //penampung untuk data-data yang akan disimpan di database schedule time : Array of Long </pre> <pre> begin lview.onItemClick(); schedule <- new ArrayList(); //mengambil nilai id dari schedule sesuai dari schedule yang dipilih messageId <- schedule.get(index); intent <- new Intent(); //menyimpan nilai terhadap intent baru intent.putExtra(messageId,messageId); startActivity(intent); intent <- intent.getIntent(); //mengambil nilai simpanan yang ada pada intent messageId <- intent.getStringExtra(); database <- new TimeListdatabaseHelper(); //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 : Cursor); cntdata.setText(datetime); //melakukan pengecekan apakah nilai frequency sama dengan yang di database frequency <- databaseHelper.getMessageFrequency(messageId); freq_spinner.setSelection(position : Integer); rcp.getText(); //mengambil nilai baru pada calendar calendar <- Calendar.getInstance(); </pre>

No.	SD - 06
	<pre> Calendar.set(field : Integer , value : Integer); timemillis <- getTimeInMillis(); timesent <- timemillis; //mengambil nilai baru pada frequency frequency <- freq_spinner.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(); //memperbarui data pada database databaseHelper.updateType(type, content, messageId); databaseHelper.updateContent(messageId, data); databaseHelper.deleteTime(messageId); //menyimpan ulang waktu pada schedule databaseHelper.saveScheduleToTime(timemillis, timesent, messageId); intent <- new Intent(); intent.putExtra("timemillis", timesent); pending <- pending.getActivity(); //melakukan penyimpanan terhadap schedule yang siap dikirim alarm <- getSystemService(ALARM_SERVICE); freqtimes <- new Frequency(); alarm <- new AlarmManager(); //pengecekan terhadap jumlah schedule yang sama yang akan dikirim if(countData > 1) then freqtimes <- repetition(pending, alarm, databaseHelper, data, time); else alarm.set(); endif en </pre>

Tabel 60. Spesifikasi proses getRecipient

No.	SP - 14
Nama Method	getRecipient
Nama Class	TimeListDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 02]
Deskripsi	Method ini digunakan untuk mengambil nomor-nomor recipient dengan id message tertentu
Return	recipient : String
Parameter	messageId : String
Method yang terkait	moveToNext(), 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)); 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 61. Spesifikasi proses getMessageDatetime

No.	SP - 15
Nama Method	getMessageDatetime
Nama Class	TimeListDatabaseHelper
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	getReadableDatabase(), rawQuery(), moveToNext(), getString(), getColumnIndex()
Logika Proses	<pre> function getMessageDatetime(messageId : String) var datetime : String //sebagai penampung data datetime hasil fetch dari database openHelper : TimeListDatabaseHelper databaseReadable : SQLiteDatabase cursor : Cursor //sebagai penampung dari data yang berhasil diambil dari database begin //sebagai inisiasi method untuk melakukan query read (baca) dari database databaseReadable <- openHelper.getReadableDatabase(); cursor <- databaseReadable.rawQuery("select message_timedate from table message where message_id='messageId'"); if(cursor.moveToFirst() = true) then datetime <- cursor.getString(cursor.getColumnIndex("message_timedate")); endif return datetime; end endfunction </pre>

Tabel 62. Spesifikasi proses getMessageFrequency

No.	SP - 16
Nama Method	getMessageFrequency
Nama Class	TimeListDatabaseHelper
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	getReadableDatabase(), rawQuery(), moveToNext(), getString(), getColumnIndex()
Logika Proses	<pre> function getMessageFrequency(messageId : String) var openHelper : TimeListDatabaseHelper databaseReadable : SQLiteDatabase frequency : String //sebagai penampung data frequency hasil fetch dari database cursor : Cursor //sebagai penampung dari data yang berhasil diambil dari database begin //sebagai inisiasi method untuk melakukan query read (baca) dari database databaseReadable <- openHelper.getReadableDatabase();cursor <- databaseReadable.rawQuery("select message_frequency from table message where message_id='messageId'"); if(cursor.moveToFirst() = true) then frequency <- cursor.getString(cursor.getColumnIndex("message_frequency")); endif return frequency; end endfunction </pre>

Tabel 63. Spesifikasi proses getTime

No.	SP - 17
Nama Method	getTime
Nama Class	TimeListDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 02]
Deskripsi	Method ini digunakan untuk mendapatkan semua <i>time</i> dengan <i>id message</i> tertentu

No.	SP - 17
Nama Method	getTime
Return	cursor : Cursor
Parameter	messageId : String
Method yang terkait	getReadableDatabase(), rawQuery()
Logika Proses	<pre> function getTime(messageId : String) var openHelper : TimeListDatabaseHelper databaseReadable : SQLiteDatabase cursor : Cursor //sebagai penampung dari data yang berhasil diambil dari database begin databaseReadable <- openhelper.getReadableDatabase(); //sebagai inisiasi method untuk melakukan query read (baca) dari database cursor <- databaseReadable.rawQuery("select * from table time where time_message_id='messageId'"); return cursor; end endfunction </pre>

Tabel 64. Spesifikasi proses getMessage

No.	SP - 18
Nama Method	getMessage
Nama Class	TimeListDatabaseHelper
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	getReadableDatabase(), rawQuery(), moveToNext(), getString(), getColumnIndex()
Logika Proses	<pre> function getMessage(messageId : String) var message : String, //sebagai penampung data message hasil fetch dari database openHelper : TimeListDatabaseHelper, databaseReadable : SQLiteDatabase cursor : Cursor, //sebagai penampung dari data yang berhasil diambil dari database begin //sebagai inisiasi method untuk melakukan query read (baca) dari database databaseReadable <- openhelper.getReadableDatabase(); cursor <- databaseReadable.rawQuery("select message_message from table message where message_id='messageId'"); if(cursor.moveToFirst() = true) then message <- cursor.getString(cursor.getColumnIndex("message_message")); endif return message; end endfunction </pre>

Tabel 65. Spesifikasi proses deleteRecipient

No.	SP - 19
Nama Method	deleteRecipient
Nama Class	TimeListDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 11]
Deskripsi	Method ini digunakan untuk menghapus nomor <i>recipient</i> dengan <i>id message</i> tertentu dari <i>database</i>
Return	-
Parameter	messageId : String
Method yang terkait	getWritableDatabase(), rawQuery()
Logika Proses	<pre> procedure deleteRecipient(messageId : String) var openHelper : TimeListDatabaseHelper, databaseWriteable : SQLiteDatabase </pre>

No.	SP - 19
Nama Method	deleteRecipient
Logika Proses	<pre> begin databaseWriteable <- openHelper.getWritableDatabase(); //sebagai inisiasi method untuk melakukan query write (tulis) ke database databaseWriteable.rawQuery("delete from recipient where recipient_message_id='messageId'"); end endprocedure </pre>

Tabel 66. Spesifikasi proses saveScheduleToRecipient

No.	SP - 20
Nama Method	saveScheduleToRecipient
Nama Class	TimeListDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 11]
Deskripsi	Method ini digunakan untuk menyimpan <i>schedule</i> yang diubah ke tabel <i>recipient</i> di <i>database</i>
Return	-
Parameter	recipients, messageId : String
Method yang terkait	getWritableDatabase(), put(), insert(), clear(), hasMoreelements()
Logika Proses	<pre> procedure saveScheduleToRecipient(recipients, messageId : String) var openHelper : TimeListDatabaseHelper databaseWriteable : SQLiteDatabase st : StringTokenizer //sebagai inisiasi terhadap class StringTokenizer tempPhoneNumber : String //sebagai penampung data nomor recipient contentValues : ContentValues //sebagai inisiasi terhadap class ContentValues begin //sebagai inisiasi method untuk melakukan query write (tulis) ke database databaseWriteable <- openHelper.getWritableDatabase(); StringTokenizer st <- new StringTokenizer(recipients, ","); while (st.hasMoreElements()) do tempPhoneNumber <- (String) st.nextElement(); contentValues.put("recipient_number", tempPhoneNumber); contentValues.put("recipient_message_id", messageId); databaseWriteable.insert("recipient", null, contentValues); contentValues.clear(); endwhile contentValues.clear(); end endprocedure </pre>

Tabel 67. Spesifikasi proses getType

No.	SP - 21
Nama Method	getType
Nama Class	Schedule
Requirement	[REQ UPD - 01], [REQ UPD - 12]
Deskripsi	Method ini digunakan untuk mengambil tipe pesan
Return	type : String
Parameter	-
Method yang terkait	-
Logika Proses	<pre> function getType() begin return type; end endfunction </pre>

Tabel 68. Spesifikasi proses updateType

No.	SP - 22
Nama Method	updateType
Nama Class	TimeListDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 12]

No.	SP - 22
Nama Method	updateType
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	getWritableDatabase(), rawQuery()
Logika Proses	<pre> procedure updateType(type, content, messageId : String) Var openHelper : TimeListDatabaseHelper //sebagai inisiasi class TimeListDatabaseHelper databaseWriteable : SQLiteDatabase //sebagai inisiasi class SQLiteDatabase begin //sebagai inisiasi method untuk melakukan query write (tulis) ke database databaseWriteable <- openHelper.getWritableDatabase(); //melakukan pengecekan kondisi untuk tipe pesan if(type = "normal") then databaseWriteable.rawQuery("update normal_message set nm_message='content' where nm_message_id='messageId'"); else databaseWriteable.rawQuery("update typical_message set tm_message='content' where tm_message_id='messageId'"); endif end endprocedure </pre>

Tabel 69. Spesifikasi proses updateContent

No.	SP - 23
Nama Method	updateContent
Nama Class	TimeListDatabaseHelper
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	getWritableDatabase(), rawQuery()
Logika Proses	<pre> procedure updateContent(data, messageId : String) var openHelper : TimeListDatabaseHelper //sebagai inisiasi class TimeListDatabaseHelper databaseWriteable : SQLiteDatabase //sebagai inisiasi class SQLiteDatabase begin //sebagai inisiasi method untuk melakukan query write (tulis) ke database databaseWriteable <- openHelper.getWritableDatabase(); //mengupdate isi pesan sesuai dengan yang diinput user databaseWriteable.rawQuery("update message set message_message='data' where message_id='messageId'"); end endprocedure </pre>

Tabel 70. Spesifikasi proses deleteTime

No.	SP - 24
Nama Method	deleteTime
Nama Class	TimeListDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 11]
Deskripsi	Method ini digunakan untuk menghapus time dengan id message tertentu
Return	-
Parameter	messageId : String
Method yang terkait	getWritableDatabase(), rawQuery()
Logika Proses	<pre> procedure deleteTime(messageId : String) var openHelper : TimeListDatabaseHelper //sebagai inisiasi class TimeListDatabaseHelper databaseWriteable : SQLiteDatabase //sebagai inisiasi class SQLiteDatabase begin </pre>

No.	SP - 24
Nama Method	deleteTime
	//sebagai inisiasi method untuk melakukan query write (tulis) ke database databaseWriteable <- openHelper.getWritableDatabase(); databaseWriteable.rawQuery("delete from time where time_message_id='messageId'"); end endprocedure

Tabel 71. Spesifikasi proses saveScheduleToTime

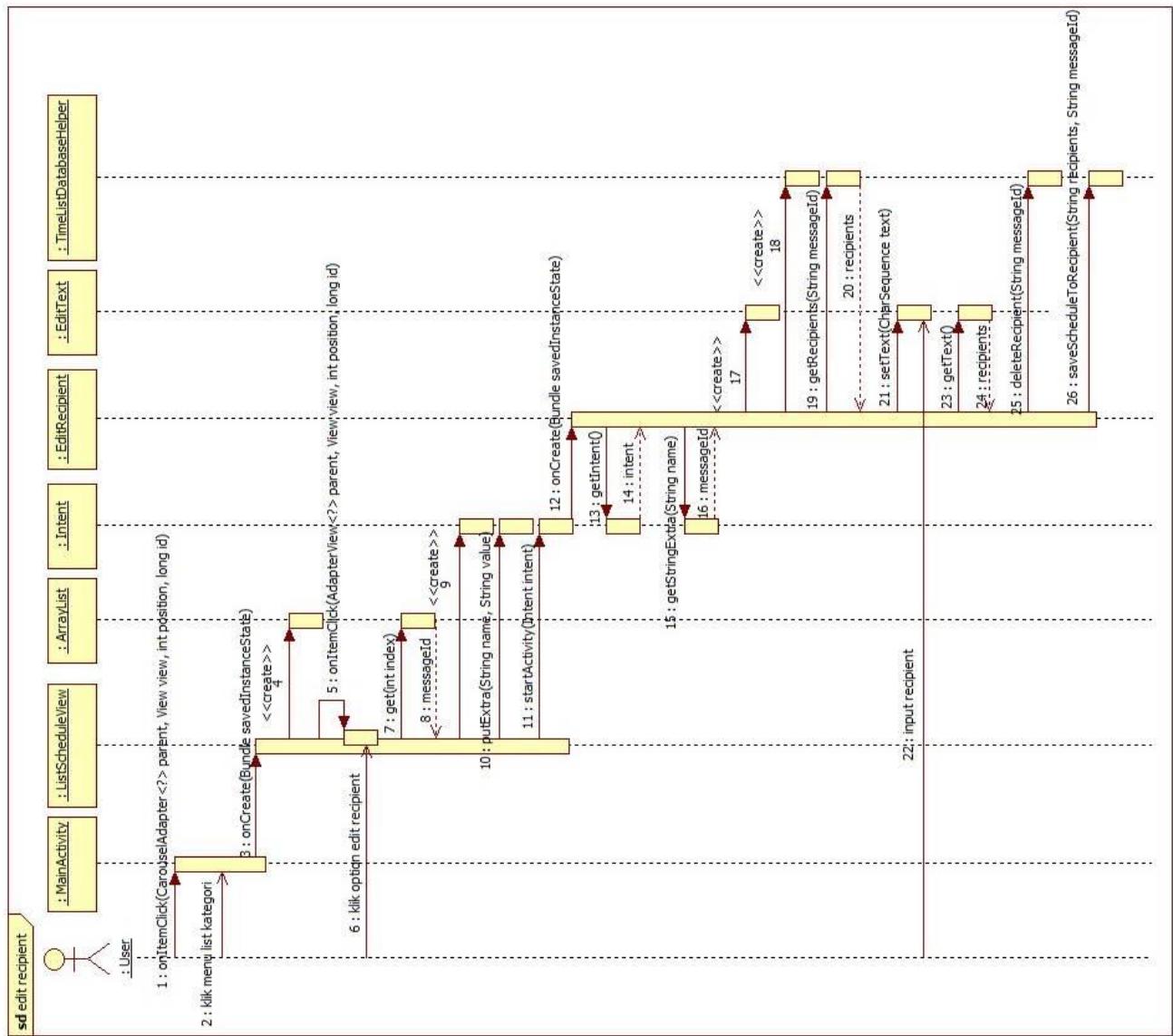
No.	SP - 25
Nama Method	saveScheduleToTime
Nama Class	TimeListDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 11]
Deskripsi	Method ini digunakan untuk menyimpan waktu schedule ke tabel time di database
Return	-
Parameter	timemillis, timesent : long messageId : String
Method yang terkait	getWritableDatabase(), put(), insert(), clear()
Logika Proses	<pre>procedure saveScheduleToTime(timemillis, timesent : long, messageId : String) var openHelper : TimeListDatabaseHelper //sebagai inisiasi class TimeListDatabaseHelper databaseWriteable : SQLiteDatabase //sebagai inisiasi class SQLiteDatabase contentValues : ContentValues //sebagai inisiasi terhadap class ContentValues begin //sebagai inisiasi method untuk melakukan query write (tulis) ke database databaseWriteable <- openHelper.getWritableDatabase(); contentValues.put("time_timemillis", timemillis); contentValues.put("time_message_id", messageId); contentValues.put("time_timesent", timesent); databaseWriteable.insert("time", null, contentValues); contentValues.clear(); end endprocedure</pre>

Tabel 72. Spesifikasi proses repetition

No.	SP - 26
Nama Method	repetition
Nama Class	Frequency
Requirement	[REQ UPD - 01], [REQ UPD - 09]
Deskripsi	Method ini digunakan untuk menangani pengulangan pengiriman schedule
Return	-
Parameter	pending : PendingIntent, alarm : AlarmManager, databaseHelper : TimeListDatabaseHelper, data : Array of String, time : Array of Long
Method yang terkait	setRepeating(), parseInt(), add(), saveScheduleToTime()
Logika Proses	<pre>procedure repetition(pending : PendingIntent, alarm : AlarmManager, databaseHelper : TimeListDatabaseHelper, data : Array of String, time : Array of Long) var timemillis : long //sebagai penampung data timemillis timesent : long //sebagai penampung data timesent frequency : String //sebagai penampung data frequency messageId : String //sebagai penampung data messageId interval : long //sebagai penampung data interval begin timemillis <- time[0]; timesent <- time[1]; frequency <- data[0]; messageId <- data[1]; //memanggil method interval untuk menghitung interval waktu sesuai dengan frequencynya interval <- interval(frequency); //set ke dalam AlarmManager sebagai interrupt alarm.setRepeating(AlarmManager.RTC_WAKEUP, timesent, interval, pending);</pre>

No.	SP - 26
Nama Method	repetition
	//insert ke dalam database sesuai dengan remaining yang diinputkan for(int remaining <- Integer.parseInt(data[2]) to remaining > 1) do //melakukan penambahan timemillis bila ditemukan terdapat timemillis yang sama di database timemillis <- databaseHelper.addTimemillisFromTime(timemillis); //melakukan insert ke database ke table time databaseHelper.saveScheduleToTime(timemillis, timesent, messageId); endfor end endprocedure

4.2.2.1.5 Sequence Diagram Edit Recipient Schedule



Gambar 20 . Sequence diagram edit recipient

Keterangan dari *sequence diagram edit recipient* pada gambar 20, dijelaskan pada tabel 73 - 76.

Tabel 73. Keterangan *sequence diagram edit recipient*

No.	SD - 07
Requirement	[REQ F - 01.3], [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	MainActivity, ListScheduleView, ArrayList, Intent, EditRecipient, TimeListDatabaseHelper, EditText
Method yang terkait	onItemClick(), onCreate(), get(), putExtra(), startActivity(), getIntent(), getStringExtra(), getRecipients(), setText(), getText(), deleteRecipients(), saveScheduleToRecipient()
<pre> var messageId : String, //sebagai penampung messageId intent Intent //sebagai inisiasi terhadap class Intent recipients : String, //sebagai penampung recipient hasil fetch dari database recipientsText : EditText, //sebagai inisiasi dari class EditText databaseHelper : TimeListDatabaseHelper, //sebagai inisisasi terhadap class TimeListDatabaseHelper begin //sebagai inisiasi terhadap class Intent intent <- new Intent(ListScheduleView.this, EditContent.class); if(menu.clicked = "Edit Recipient") then //mendapatkan index message id dari list schedule yang dipilih messageId <- get(index(schedule.clicked)); //memasukkan hasil messageId agar dapat diakses intent lain intent.putExtra("messageId", messageId); //memulai activity EditRecipient startActivity(intent); endif //bila activity mulai if(activity.started) then //fungsi untuk mendapatkan value atau nilai yang di pass oleh intent lain intent <- getIntent(); //mendapatkan nilai yang di pass oleh intent lain messageId <- intent.getStringExtra("messageId"); //mendapatkan record recipient dari table recipient dengan messageId tertentu recipients <- getRecipients(messageId); //set recipientsText recipientsText.setText(recipients); //mendapatkan nilai dari recipients yang telah diinputkan oleh user recipients <- getText(); //menghapus recipient dengan messageId tertentu databaseHelper.deleteRecipient(messageId); //insert data recipient ke dalam database databaseHelper.saveScheduleToRecipients(recipient, messageId); endif end </pre>	

Tabel 74. Spesifikasi proses *getRecipient*

No.	SP - 27
Nama Method	getRecipient
Nama Class	TimeListDatabaseHelper
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	moveToNext(), 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 //add recipient ke dalam array while(recipient.moveToNext()) do contact.add(recipient.getString(recipient.getColumnIndex("recipient number"))); </pre>	

No.	SP - 27
Nama Method	getRecipient
<pre> 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(); return recipients; end endfunction </pre>	

Tabel 75. Spesifikasi proses deleteRecipient

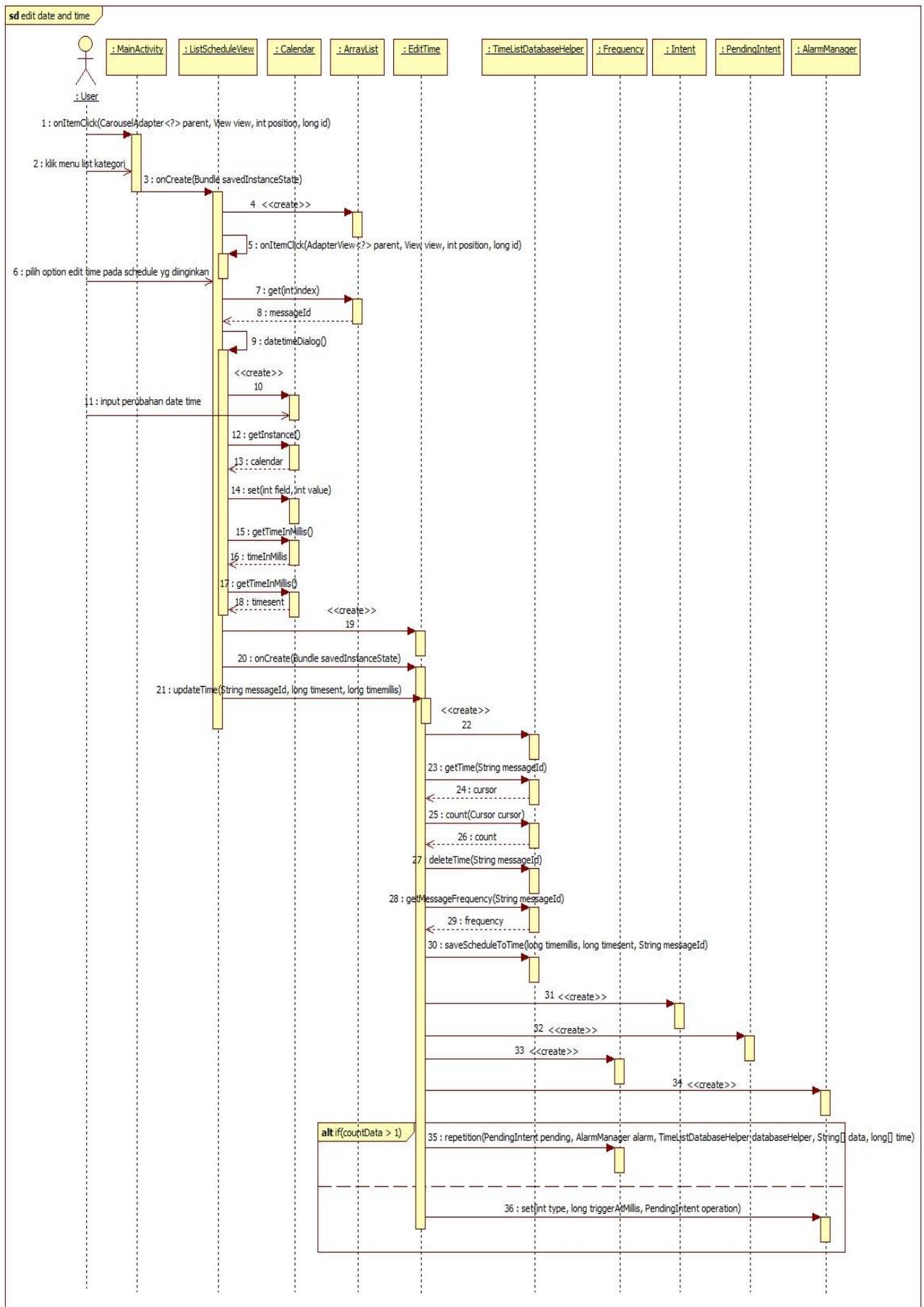
No.	SP - 28
Nama Method	deleteRecipient
Nama Class	TimeListDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 11]
Deskripsi	Method ini digunakan untuk menghapus nomor recipient dengan id message tertentu dari database
Return	-
Parameter	messageId : String
Method yang terkait	getWritableDatabase(), rawQuery()
Logika Proses	<pre> procedure deleteRecipient(messageId : String) var openHelper : TimeListDatabaseHelper //sebagai inisiasi class TimeListDatabaseHelper databaseWriteable : SQLiteDatabase //sebagai inisiasi class SQLiteDatabase begin //sebagai inisiasi method untuk melakukan query write (tulis) ke database databaseWriteable <- openHelper.getWritableDatabase(); //melakukan penghapusan recipient yang memiliki messageId tertentu databaseWriteable.rawQuery("delete from recipient where recipient_message_id='messageId'"); end endprocedure </pre>

Tabel 76. Spesifikasi proses saveScheduleToRecipient

No.	SP - 29
Nama Method	saveScheduleToRecipient
Nama Class	TimeListDatabaseHelper
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	getWritableDatabase(), put(), insert(), clear(), hasMoreelements()
Logika Proses	<pre> procedure saveScheduleToRecipient(recipients, messageId : String) var openHelper : TimeListDatabaseHelper //sebagai inisiasi class TimeListDatabaseHelper databaseWriteable : SQLiteDatabase //sebagai inisiasi class SQLiteDatabase st : StringTokenizer //sebagai inisiasi terhadap class StringTokenizer tempPhoneNumber : String //sebagai penampung data nomor recipient contentValues : ContentValues //sebagai inisiasi terhadap class ContentValues begin //sebagai inisiasi method untuk melakukan query write (tulis) ke database databaseWriteable <- openHelper.getWritableDatabase(); StringTokenizer st <- new StringTokenizer(recipients, ";"); while (st.hasMoreElements()) do tempPhoneNumber <- (String) st.nextElement(); contentValues.put("recipient number", tempPhoneNumber); end endprocedure </pre>

No.	SP - 29
Nama Method	<pre>saveScheduleToRecipient contentValues.put("recipient_message_id", messageId); databaseWriteable.insert("recipient", null, contentValues); contentValues.clear(); endwhile contentValues.clear(); end endprocedure</pre>

4.2.2.1.6 Sequence Diagram Edit Date and Time Schedule



Gambar 21 . Sequence diagram edit date and time schedule

Keterangan dari *sequence diagram edit date and time schedule* pada gambar 21, dijelaskan pada tabel 77 - 84.

Tabel 77. Keterangan *sequence diagram edit date and time schedule*

No.	SD - 08
Requirement	[REQ F - 01.3], [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	MainActivity, ListScheduleView, ArrayList, Calendar, EditTime, TimeListDatabaseHelper, Intent, PendingIntent, Frequency, AlarmManager
Method yang terkait	onItemClick(), onCreate(), get(), datetimeDialog(), getInstance(), set(), getTimeInMillis(), updateTime(), getTime(), count(), deleteTime(), getMessageFrequency(), saveScheduleToTime(), repetition(), set()

```

Logika Proses
Var
messageId : String, //sebagai penampung messageId
calendar : Calendar, //sebagai inisiasi terhadap class Calendar
timemillis : long, //sebagai penampung waktu dalam bentuk millisecond
timesent : long, //sebagai penampung waktu dalam bentuk millisecond
cursor : Cursor, //sebagai penampung hasil fetch dari database
databaseHelper : TimeListDatabaseHelper, //sebagai inisisasi terhadap class
TimeListDatabaseHelper
countData : int, //sebagai penampung banyaknya record suatu data yang di fetch dari database
frequency : String, //sebagai penampung data frequency yang didapat dari database
intent : Intent //sebagai inisiasi terhadap class Intent
pending : PendingIntent, //sebagai inisiasi class PendingIntent
dataRepetition : Array of String, //sebagai penampung data
time : Array of long, //sebagai penampung time
freqtime : Frequency, //sebagai inisiasi terhadap class Frequency
alarm : AlarmManager, //sebagai inisiasi terhadap class AlarmManager
begin
    //sebagai inisiasi terhadap class Intent
    intent <- new Intent(EditTime.this, SMSActivity.class);
    //sebagai inisiasi class PendingIntent
    pending : PendingIntent.getActivity(EditTime.this, (int), intent,
    PendingIntent.FLAG_UPDATE_CURRENT);
    if(menu.clicked = "Edit Time") then
        //mendapatkan index messageId dari list schedule yang dipilih
        messageId <- get(index(schedule.clicked));
        //bila membuka Date and Time Picker
        if(dateTimeDialog().open) then
            //mendapatkan instance dari class Calendar
            calendar <- getInstance();
            //mengeset nilai dari second
            calendar.set(Calendar.SECOND, 0);
            //mengeset nilai dari millisecond
            calendar.set(Calendar.MILLISECOND, 0);
            //mendapatkan waktu dalam bentuk millisecond
            timemillis <- getTimeInMillis();
            //mendapatkan waktu dalam bentuk millisecond
            timesent <- getTimeInMillis();
        endif
        //melakukan update time terhadap class EditTime
        updateTime(messageId, timesent, timemillis);
        //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(timemillis, timesent, 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;

```

No.	SD - 08
<pre> //menampung waktu time[0] <- timemillis; time[1] <- timesent; //memanggil method terhadap class Frequency freqtimes.repetition(pending, alarm, databaseHelper, dataRepetition, time); else //melakukan set terhadap class AlarmManager yang berfungsi untuk interrupt alarm.set(AlarmManager.RTC_WAKEUP, timesent, pending); endif endif end </pre>	

Tabel 78. Spesifikasi proses updateTime

No.	SP - 30
Nama Method	updateTime
Nama Class	EditTime
Requirement	[REQ UPD - 01], [REQ UPD - 11]
Deskripsi	Method ini digunakan untuk meng-update time dengan waktu schedule dengan id message tertentu
Return	-
Parameter	messageId : String, timesent, timemillis : Long
Method yang terkait	getWritableDatabase(), rawQuery()
Logika Proses	<pre> procedure updateTime(messageId : String, timesent, timemillis : long) var messageId : String, //sebagai penampung messageId calendar : Calendar, //sebagai inisiasi terhadap class Calendar timemillis : long, //sebagai penampung waktu dalam bentuk millisecond timesent : long, //sebagai penampung waktu dalam bentuk millisecond cursor : Cursor, //sebagai penampung hasil fetch dari database databaseHelper : TimeListDatabaseHelper, //sebagai inisisasi terhadap class TimeListDatabaseHelper countData : int, //sebagai penampung banyaknya record suatu data yang di fetch dari database frequency : String, //sebagai penampung data frequency yang didapat dari database intent : Intent, //sebagai inisiasi terhadap class Intent pending : PendingIntent, //sebagai inisiasi 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 inisiasi terhadap class AlarmManager begin //sebagai inisiasi terhadap class Intent intent <- new Intent(EditTime.this, SMSActivity.class); //sebagai inisiasi class PendingIntent pending <- PendingIntent.getActivity(EditTime.this, (int), intent, PendingIntent.FLAG_UPDATE_CURRENT); //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(timemillis, timesent, 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] <- timemillis; time[1] <- timesent; //memanggil method terhadap class Frequency </pre>

No.	SP - 30
Nama Method	updateTime
	<pre> freqtimes.repetition(pending, alarm, databaseHelper, dataRepetition, time); else //melakukan set terhadap class AlarmManager yang berfungsi untuk interrupt alarm.set(AlarmManager.RTC_WAKEUP, timesent, pending); endif end endprocedure </pre>

Tabel 79. Spesifikasi proses getTime

No.	SP - 31
Nama Method	getTime
Nama Class	TimeListDatabaseHelper
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	getReadableDatabase(), rawQuery()
Logika Proses	<pre> function getTime(messageId : String) var openHelper : TimeListDatabaseHelper //sebagai inisiasi class TimeListDatabaseHelper databaseWritableDatabase : SQLiteDatabase //sebagai inisiasi class SQLiteDatabase cursor : Cursor //sebagai penampung dari data yang berhasil diambil dari database begin //sebagai inisiasi method untuk melakukan query read (baca) dari database databaseReadable <- openHelper.getReadableDatabase(); //mendapatkan semua record time dengan messageId tertentu cursor <- databaseReadable.rawQuery("select * from table time where time_message_id='messageId'"); return cursor; end endfunction </pre>

Tabel 80. Spesifikasi proses count

No.	SP - 32
Nama Method	count
Nama Class	TimeListDatabaseHelper
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 81. Spesifikasi proses deleteTime

No.	SP - 33
Nama Method	deleteTime
Nama Class	TimeListDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 11]

No.	SP - 33
Nama Method	deleteTime
Deskripsi	Method ini digunakan untuk menghapus <i>time</i> dengan <i>id message</i> tertentu
Return	-
Parameter	messageId : String
Method yang terkait	getWritableDatabase(), rawQuery()
Logika Proses	<pre> procedure deleteTime(messageId : String) var openHelper : TimeListDatabaseHelper //sebagai inisiasi class TimeListDatabaseHelper databaseWriteable : SQLiteDatabase //sebagai inisiasi class SQLiteDatabase begin //sebagai inisiasi method untuk melakukan query write (tulis) ke database databaseWriteable <- openHelper.getWritableDatabase(); //melakukan delete time dengan messageId tertentu databaseWriteable.rawQuery("delete from time where time_message_id='messageId'"); end endprocedure </pre>

Tabel 82. Spesifikasi proses getMessageFrequency

No.	SP - 34
Nama Method	getMessageFrequency
Nama Class	TimeListDatabaseHelper
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	getReadableDatabase(), rawQuery(), moveToNext(), getString(), getColumnIndex()
Logika Proses	<pre> function getMessageFrequency(messageId : String) var openHelper : TimeListDatabaseHelper //sebagai inisiasi class TimeListDatabaseHelper databaseReadable : SQLiteDatabase //sebagai inisiasi class SQLiteDatabase frequency : String //sebagai penampung data frequency hasil fetch dari database cursor : Cursor //sebagai penampung dari data yang berhasil diambil dari database begin //sebagai inisiasi method untuk melakukan query read (baca) dari database databaseReadable <- openHelper.getReadableDatabase(); //mendapatkan frequency dari suatu schedule dengan messageId tertentu cursor <- databaseReadable.rawQuery("select message_frequency from table message where message_id='messageId'"); //jika cursor terisi if(cursor.moveToFirst() = true) then //mengambil frequency dari cursor dengan atribut "message_frequency" frequency <- cursor.getString(cursor.getColumnIndex("message_frequency")); endif return frequency; end endfunction </pre>

Tabel 83. Spesifikasi proses saveScheduleToTime

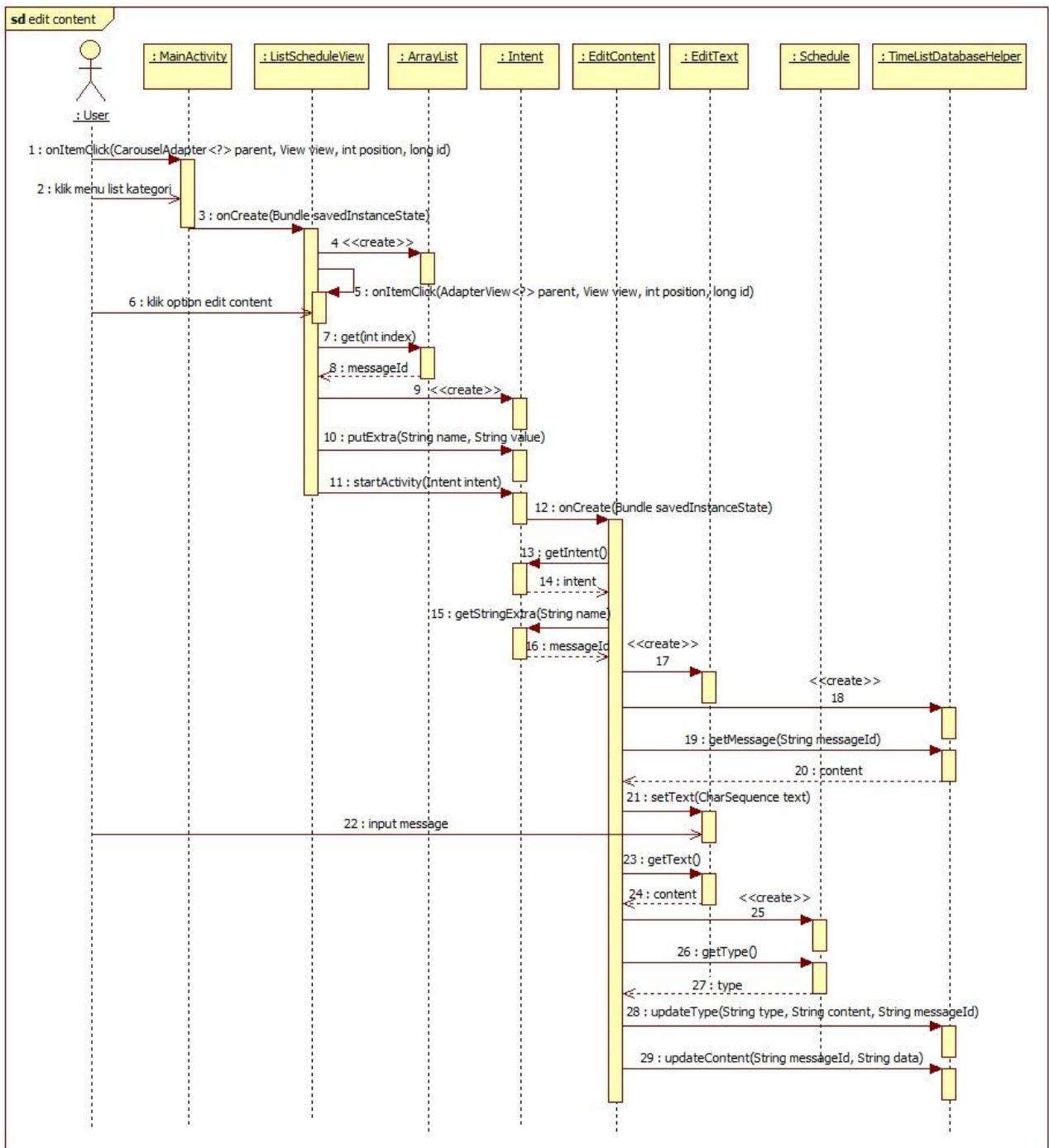
No.	SP - 35
Nama Method	saveScheduleToTime
Nama Class	TimeListDatabaseHelper
Requirement	[REQ UPD - 01], [REQ UPD - 11]
Deskripsi	Method ini digunakan untuk menyimpan waktu <i>schedule</i> ke tabel <i>time</i> di <i>database</i>
Return	-
Parameter	timemillis, timesent : long, messageId : String
Method yang terkait	getWritableDatabase(), put(), insert(), clear()
Logika Proses	<pre> procedure saveScheduleToTime(timemillis, timesent : long, messageId : String) </pre>

No.	SP - 35
Nama Method	saveScheduleToTime
<pre> var openHelper : TimeListDatabaseHelper //sebagai inisiasi class TimeListDatabaseHelper databaseWriteable : SQLiteDatabase //sebagai inisiasi class SQLiteDatabase contentValues : ContentValues, //sebagai inisiasi terhadap class ContentValues begin //sebagai inisiasi method untuk melakukan query write (tulis) ke database databaseWriteable <- openHelper.getWritableDatabase(); //menyimpan data yang akan diinsertkan ke table time ke dalam contentValues contentValues.put("time_timenillis", timenillis); contentValues.put("time_message_id", messageId); contentValues.put("time_timesent", timesent); databaseWriteable.insert("time", null, contentValues); contentValues.clear(); end endprocedure </pre>	

Tabel 84. Spesifikasi proses repetition

No.	SP - 36
Nama Method	repetition
Nama Class	Frequency
Requirement	[REQ UPD - 01], [REQ UPD - 09]
Deskripsi	Method ini digunakan untuk menangani pengulangan pengiriman schedule
Return	-
Parameter	pending : PendingIntent, alarm : AlarmManager, databaseHelper : TimeListDatabaseHelper, data : Array of String, time : Array of Long
Method yang terkait	setRepeating(), parseInt(), add(), saveScheduleToTime()
Logika Proses	<pre> procedure repetition(pending : PendingIntent, alarm : AlarmManager, databaseHelper : TimeListDatabaseHelper, data : Array of String, time : Array of Long) var timenillis : long //sebagai penampung data timenillis timesent : long //sebagai penampung data timesent frequency : String //sebagai penampung data frequency messageId : String //sebagai penampung data messageId interval : long //sebagai penampung data interval begin timenillis <- time[0]; timesent <- time[1]; frequency <- data[0]; messageId <- data[1]; //memanggil method interval untuk menghitung interval waktu sesuai dengan frequencynya interval <- interval(frequency); //set ke dalam AlarmManager sebagai interrupt alarm.setRepeating(AlarmManager.RTC_WAKEUP, timesent, interval, pending); //insert ke dalam database sesuai dengan remaining yang diinputkan for(int remaining <- Integer.parseInt(data[2]) to remaining > 1) do //melakukan penambahan timenillis bila ditemukan terdapat timenillis yang sama di //database timenillis <- databaseHelper.addTimenillisFromTime(timenillis); //melakukan insert ke database ke table time databaseHelper.saveScheduleToTime(timenillis, timesent, messageId); endfor end endprocedure </pre>

4.2.2.1.7 Sequence Diagram Edit Content Schedule



Gambar 22 . Sequence diagram edit content schedule

Keterangan dari *sequence diagram edit content schedule* pada gambar 22, dijelaskan pada tabel 85 - 88.

Tabel 85. Keterangan *sequence diagram edit content schedule*

No.	SD - 09
Requirement	[REQ F - 01.3], [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	MainActivity, ListScheduleView, ArrayList, Calendar, EditText, Schedule, TimeListDatabaseHelper, Intent

No.	SD - 09
Method yang terkait	<pre> onItemClick(), onCreate(), get(), putExtra(), startActivity(), getIntent(), getStringExtra(), getMessage(), setText(), getText(), getType(), updateType(), updateContent() var messageId : String, //sebagai penampung messageId intent : new Intent(ListScheduleView.this, EditContent.class), //sebagai inisiasi terhadap class Intent content : String, //sebagai penampung pesan atau content yang didapatkan dari database databaseHelper : TimeListDatabaseHelper, //sebagai inisiasi terhadap class TimeListDatabaseHelper contentText : EditText, //sebagai inisiasi terhadap class EditText dan penampung terhadap var content untuk ditampilkan type : String, //sebagai penampung tipe dari pesan schedule : Schedule, //sebagai inisiasi terhadap class Schedule begin //bila menu Edit Content dipilih if(menu.clicked = "Edit Content") then //mendapatkan index message id dari list schedule yang dipilih messageId <- get(index(schedule.clicked)); //memasukkan hasil messageId agar dapat diakses intent lain intent.putExtra("messageId", messageId); //memulai activity EditContent startActivity(intent); endif //bila activity mulai if(activity.started) then //fungsi untuk mendapatkan value atau nilai yang di pass oleh intent lain intent <- getIntent(); //mendapatkan nilai yang di pass oleh intent lain messageId <- intent.getStringExtra("messageId"); //mendapatkan isi pesan dengan messageId tertentu dari database content <- databaseHelper.getMessage(messageId); //mengeset isi contentText sesuai dengan content contentText.setText(content); //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); endif end </pre>

Tabel 86. Spesifikasi proses getMessage

No.	SP - 37
Nama Method	getMessage
Nama Class	TimeListDatabaseHelper
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	getReadableDatabase(), rawQuery(), moveToNext(), getString(), getColumnIndex()
Logika Proses	<pre> function getMessage(messageId : String) var message : String //sebagai penampung data message hasil fetch dari database openHelper : TimeListDatabaseHelper //sebagai inisiasi class TimeListDatabaseHelper databaseReadable : SQLiteDatabase //sebagai inisiasi class SQLiteDatabase cursor : Cursor //sebagai penampung dari data yang berhasil diambil dari database begin //sebagai inisiasi method untuk melakukan query read (baca) dari database databaseReadable <- openHelper.getReadableDatabase(); //mendapatkan seluruh record message_message dengan messageId tertentu cursor <- databaseReadable.rawQuery("select message_message from table message where message_id='messageId'"); //bila terdapat record yang tersimpan dalam cursor if(cursor.moveToFirst() = true) then //menyimpan isi cursor dengan atribut message_message ke dalam message message <- cursor.getString(cursor.getColumnIndex("message_message")); </pre>

No.	SP - 37
Nama Method	getMessage
	<pre> endif return message; end endfunction </pre>

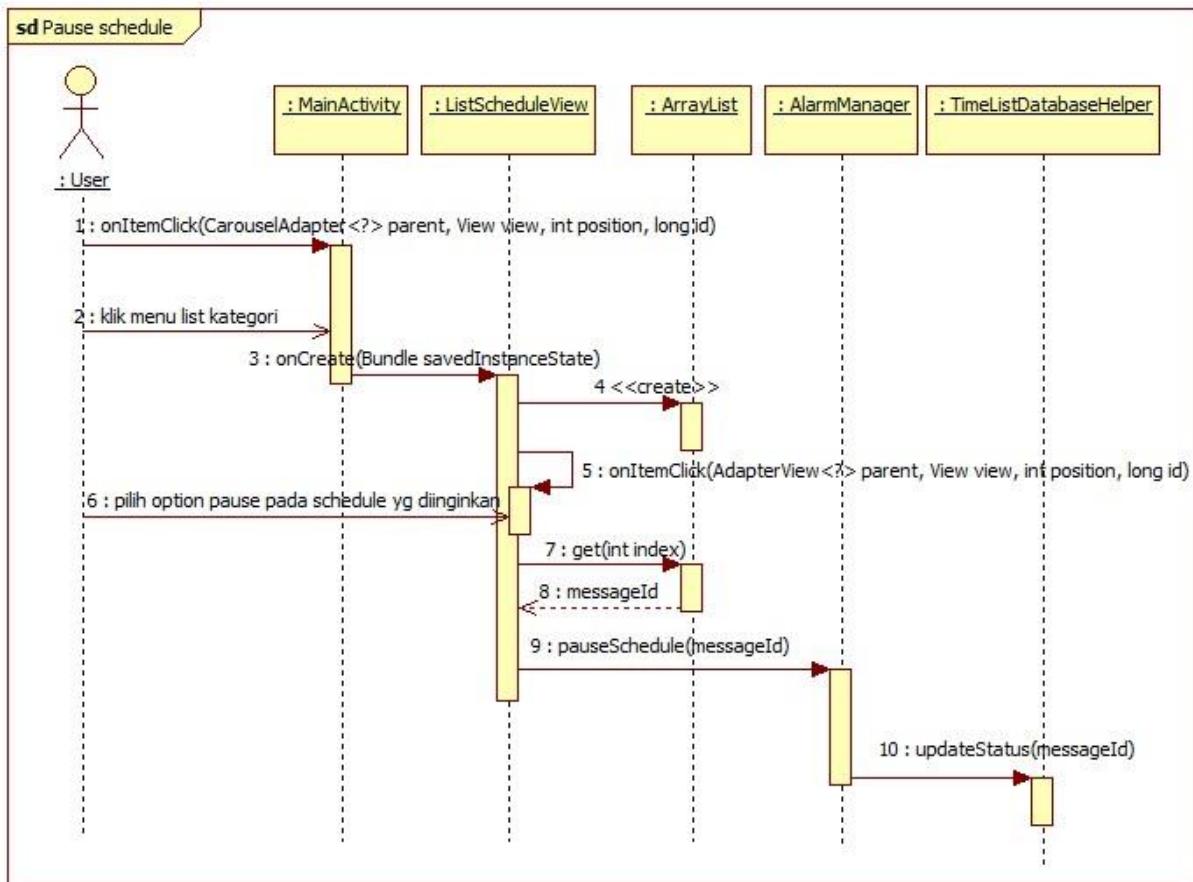
Tabel 87. Spesifikasi proses updateType

No.	SP - 38
Nama Method	updateType
Nama Class	TimeListDatabaseHelper
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	getWritableDatabase(), rawQuery()
Logika Proses	<pre> procedure updateType(type, content, messageId : String) Var openHelper : TimeListDatabaseHelper //sebagai inisiasi class TimeListDatabaseHelper databaseWriteable : SQLiteDatabase //sebagai inisiasi class SQLiteDatabase begin //sebagai inisiasi method untuk melakukan query write (tulis) ke database databaseWriteable <- openHelper.getWritableDatabase(); //bila type dari content bertipe normal if(type == "normal") then //melakukan update terhadap table normal_message dengan messageId tertentu databaseWriteable.rawQuery("update normal_message set nm_message='content' where nm_message_id='messageId'"); //melakukan update terhadap table typical_message dengan messageId tertentu else databaseWriteable.rawQuery("update typical_message set tm_message='content' where tm_message_id='messageId'"); endif end endprocedure </pre>

Tabel 88. Spesifikasi proses updateContent

No.	SP - 39
Nama Method	updateContent
Nama Class	TimeListDatabaseHelper
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	getWritableDatabase(), rawQuery()
Logika Proses	<pre> procedure updateContent(data, messageId : String) var openHelper : TimeListDatabaseHelper //sebagai inisiasi class TimeListDatabaseHelper databaseWriteable : SQLiteDatabase //sebagai inisiasi class SQLiteDatabase begin //sebagai inisiasi method untuk melakukan query write (tulis) ke database databaseWriteable <- openHelper.getWritableDatabase(); //melakukan update terhadap table message dengan messageId tertentu databaseWriteable.rawQuery("update message set message_message='data' where message_id='messageId'"); end endprocedure </pre>

4.2.2.1.8 Sequence Diagram Pause Schedule



Gambar 23 . Sequence diagram pause schedule

Keterangan dari *sequence diagram pause schedule* pada gambar 23, dijelaskan pada tabel 89 - 91.

Tabel 89. Keterangan *sequence diagram pause schedule*

No.	SD - 10
Requirement	[REQ F - 01.4], [REQ PSD - 01]
Deskripsi	Sequence ini menggambarkan proses untuk menghentikan sementara pengiriman suatu schedule yang sebelumnya telah dibuat (<i>pause schedule</i>)
Class yang terkait	MainActivity, ListScheduleView, ArrayList, AlarmManager, TimeListDatabaseHelper
Method yang terkait	onItemClick(), onCreate(), get(), pauseSchedule(), updateStatus()
Logika Proses	<pre> Var openHelper : TimeListDatabaseHelper //sebagai inisiasi class TimeListDatabaseHelper cursor : Cursor //sebagai inisiasi class Cursor scheduleId : ArrayList of Integer //sebagai penampung seluruh id schedule yang ada di database messageId, index : Integer //sebagai penampung id message dan index pesan alarm : AlarmManager //sebagai inisiasi class AlarmManager status : String //sebagai penampung status schedule begin //Jika user memilih opsi pause schedule if (onContextItemSelected('paused')) then messageId <- scheduleId.get(index); alarm.pauseSchedule(messageId); //remove schedule on alarm manager not delete from database //mengupdate status schedule dari scheduled menjadi paused openHelper.updateStatus(status, messageId); endif end </pre>

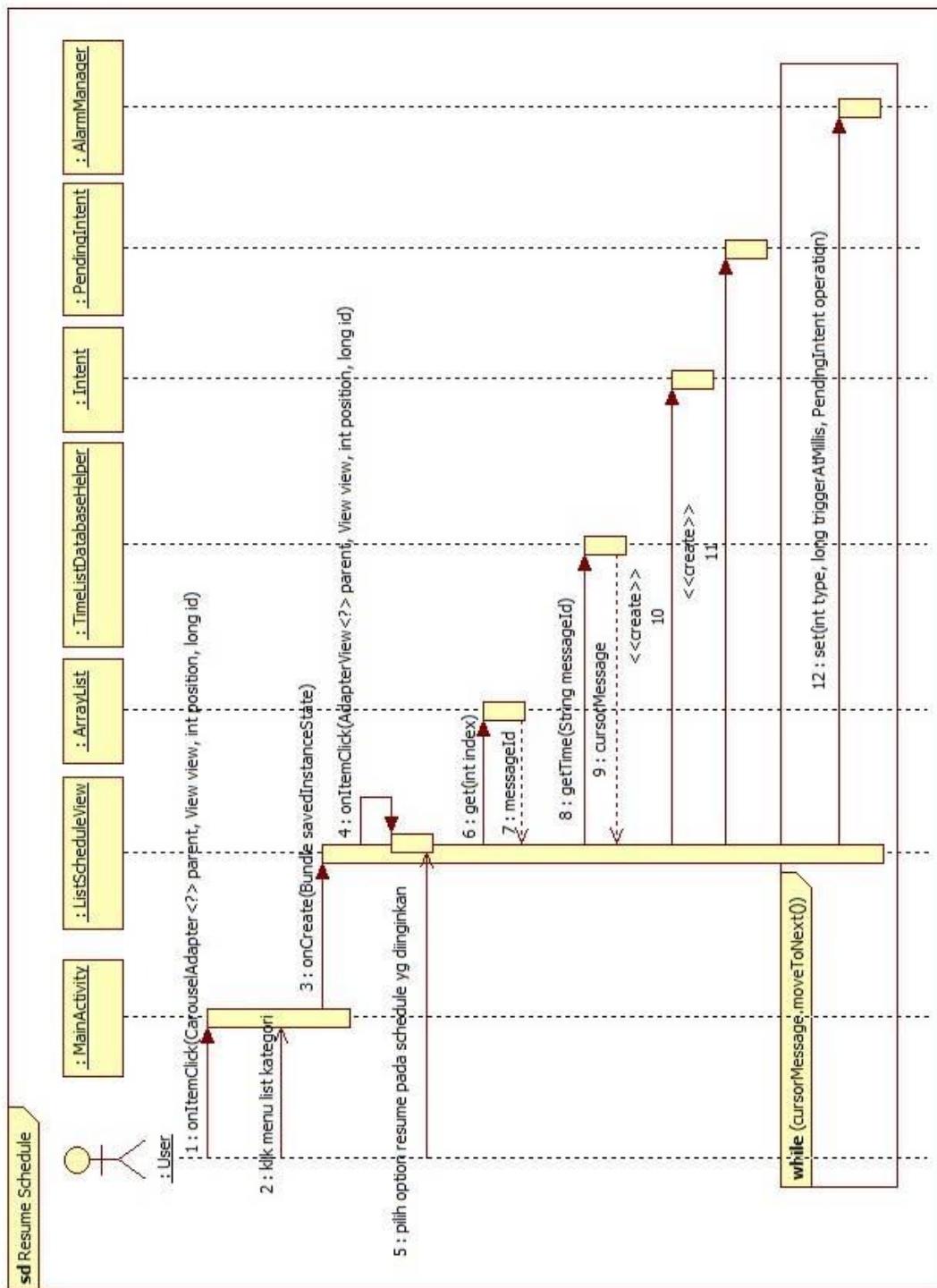
Tabel 90. Spesifikasi proses pauseSchedule

No.	SP - 40
Nama Method	pauseSchedule
Nama Class	AlarmManager
Requirement	[REQ PSD - 01]
Deskripsi	<i>Method ini digunakan untuk menghapus schedule dengan id message tertentu dari alarm manager</i>
Return	Boolean
Parameter	messageId : String
Method yang terkait	size(), remove(), get()
Logika Proses	<pre>function pauseSchedule(messageId : String) cursor : Cursor //sebagai inisiasi class Cursor scheduleId : ArrayList of Integer //sebagai penampung seluruh id schedule yang ada di database alarm : AlarmManager //sebagai inisiasi class AlarmManager //assumption : already have scheduleId arraylist which fill with id for each 'paused' status begin for(integer i = 0 to scheduleId.size()) if(scheduleId.get(i) = messageId) then scheduleId.remove(i); return true; else return false; endif endfor end endfunction</pre>

Tabel 91. Spesifikasi proses updateStatus

No.	SP - 41
Nama Method	updateStatus
Nama Class	Time
Requirement	[REQ PSD - 01]
Deskripsi	<i>Method ini digunakan untuk mengupdate status schedule di database</i>
Return	Boolean
Parameter	messageId, status : String
Method yang terkait	put(), update()
Logika Proses	<pre>function updateStatus(status : String , messageId : String) database : SQLiteDatabase //sebagai inisiasi class SQLiteDatabase contentValues : ContentValues //sebagai inisiasi begin contentValues <- new ContentValues(); try contentValues.put(SCHEDULE_COLUMN_STATUS, status); database.update(TABLE_SCHEDULE, contentValues, SCHEDULE_COLUMN_ID + "=" + messageId, null); return true; endtry catch(SQLException sqe) return false; endcatch end endfunction</pre>

4.2.2.1.9 Sequence Diagram Resume Schedule



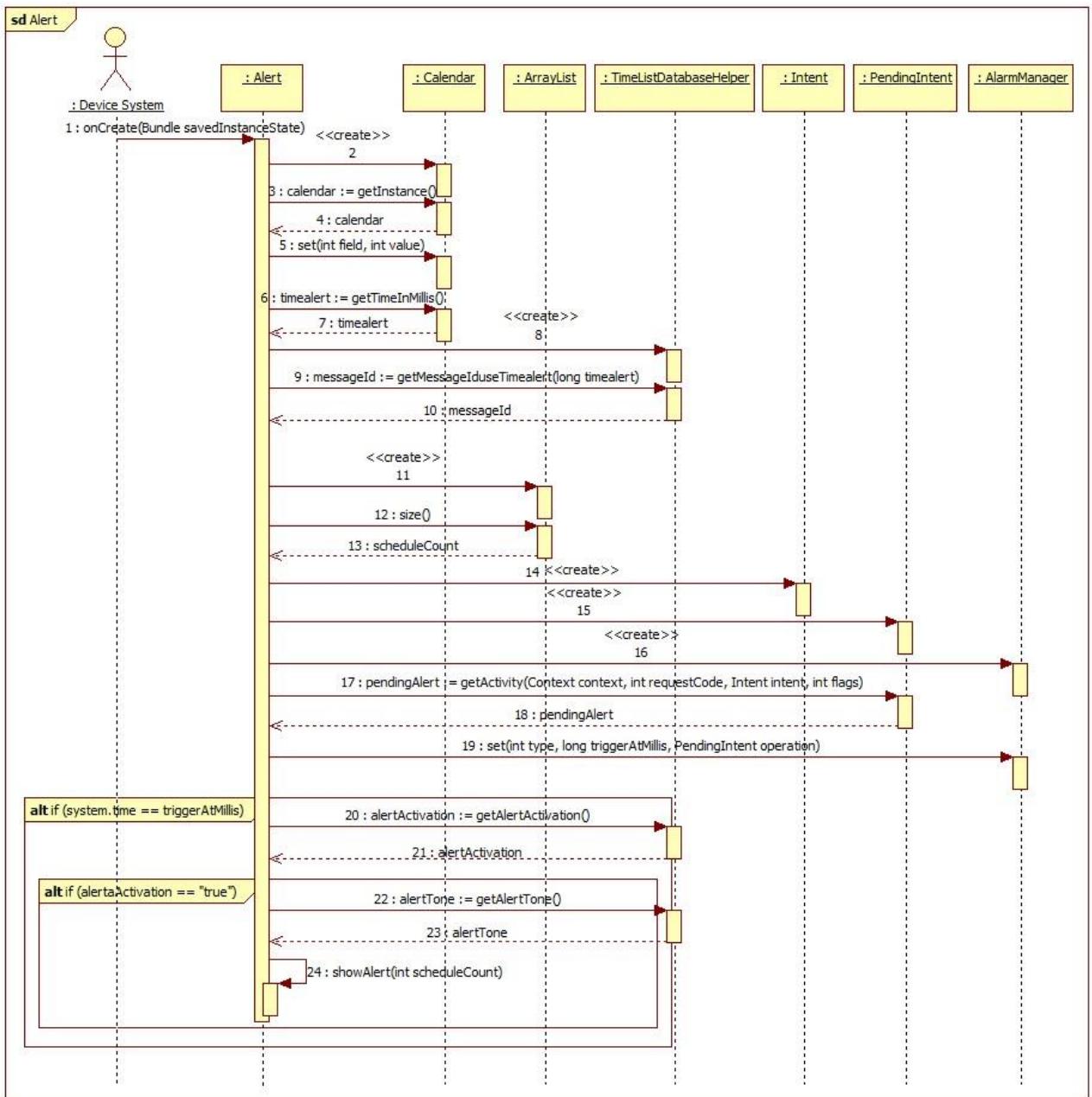
Gambar 24 . *Sequence diagram resume schedule*

Keterangan dari *sequence diagram resume schedule* pada gambar 24, dijelaskan pada tabel 92.

Tabel 92. Keterangan *sequence diagram resume schedule*

No.	SD - 11
Requirement	[REQ F - 01.4], [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	MainActivity, ListScheduleView, ArrayList, TimeListDatabaseHelper, Intent, PendingIntent, AlarmManager
Method yang terkait	onItemClick(), onCreate(), get(), getTime(), set()
Logika Proses	<pre> Var messageId : String //sebagai penampung id message alarm : AlarmManager //sebagai inisiasi dari class AlarmManager database : TimeListDatabaseHelper //sebagai inisiasi dari class TimeListDatabaseHelper array_list : ArrayList //sebagai penampung inisiasi class ArrayList cursorMessage : Cursor //sebagai penampung inisiasi class Cursor intentResume : Intent //sebagai penampung inisiasi class Intent pendingResume : PendingIntent //sebagai penampung inisiasi class PendingIntent menu : MenuItem //sebagai penampung inisiasi class MenuItem index : Integer //sebagai penampung index message begin array_list <- new ArrayList(); if (onContextItemSelected('paused')) then messageId <- array_list.get(index); cursorMessage <- database.getTime(messageId); Intent intent <- new Intent(); intent.putExtra("CursorMessage", cursorMessage); PendingIntent p_intent <- new PendingIntent(); pendingResume <- p_intent.getActivity(); while(cursorMessage.moveToNext()) do alarm.set(AlarmManager.RTC_WAKEUP, cursorMessage, pendingResume); endif end </pre>

4.2.2.1.10 Sequence Diagram Menampilkan Alert



Gambar 25 . Sequence diagram menampilkan alert

Keterangan dari *sequence diagram* menampilkan *alert* pada gambar 25, dijelaskan pada tabel 93 - 94.

Tabel 93. Keterangan *sequence diagram* menampilkan *alert*

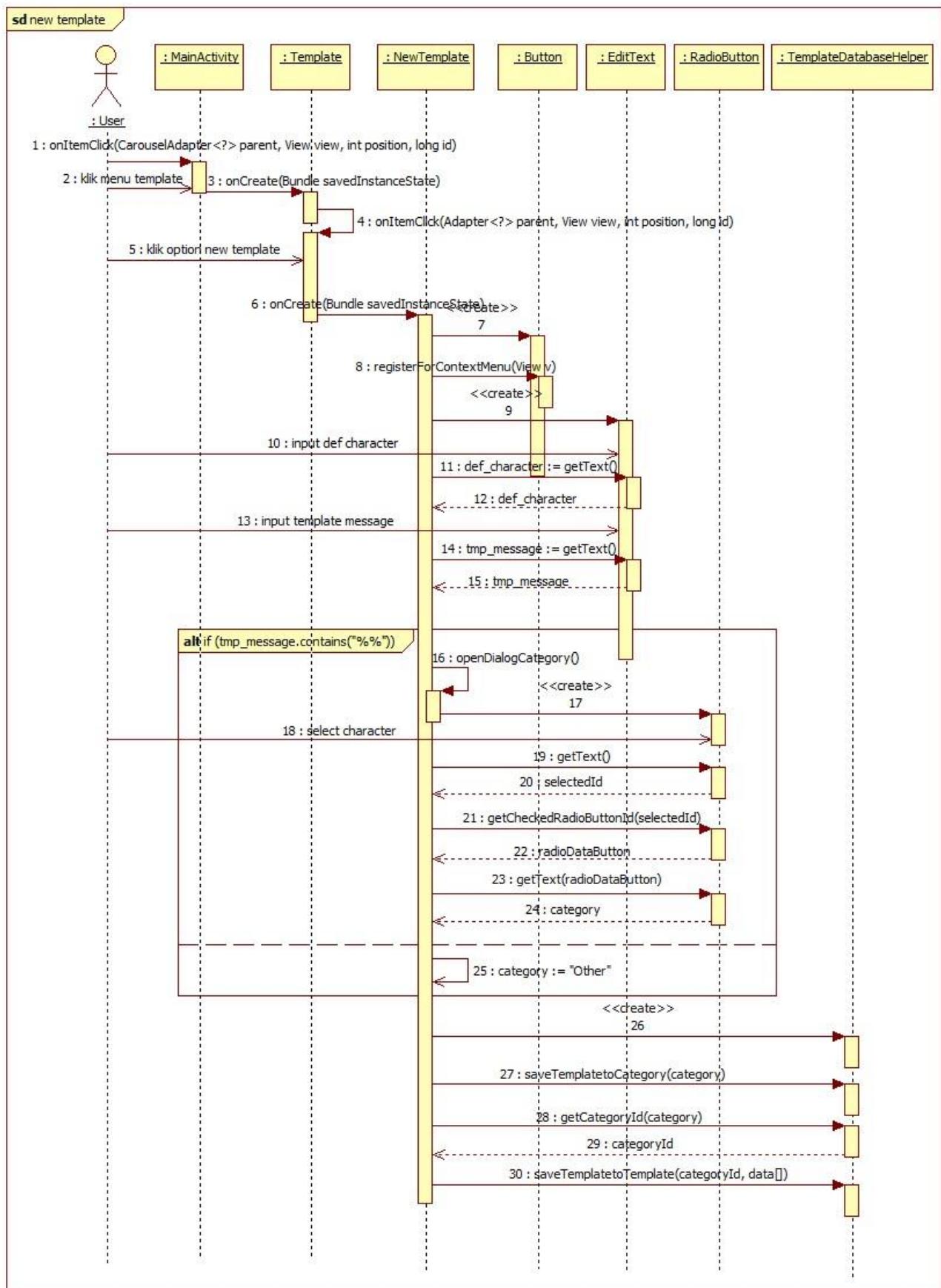
No.	SD - 12
Requirement	[REQ F - 03], [REQ ALR - 01], [REQ ALR - 02], [REQ ALR - 03], [REQ ALR - 04], [REQ ALR - 05]
Class yang terkait	MainActivity,
Method yang terkait	onItemClick(), onCreate(),
Deskripsi	<i>Sequence</i> ini digunakan untuk menjalankan fungsi alert sesuai dengan pengaturan yang telah disimpan pada database.
Logika Proses	Var

No.	SD - 12
	<pre> 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 : PendingIntent //sebagai inisiasi dari class PendingIntent alertActivation : Boolean //sebagai penampung status aktivasi alert alertTone : Uri //sebagai penampung nada dering alert begin Calendar cal <- new Calendar(); calendar <- cal.getInstance(); cal.set(field, value); timealert <- cal.getTimeInMillis(); TimeListDatabaseHelper database <- new TimeListDatabaseHelper(); messageId <- database.getMessageIduseTimealer(timealert); ArrayList array_list <- new ArrayList(); scheduleCount <- array_list.size(); Intent intentAlert <- new Intent(); intent.putExtra("ScheduleCount", scheduleCount); PendingIntent p_intent <- new PendingIntent(); pendingAlert <- p_intent.getActivity(); alarm.set(AlarmManager.RTC_WAKEUP, timealert, pendingAlert); long triggerAtMillis <- timealert; if(system.time == triggerAtMillis) then alertActivation <- database.getAlertActivation(); if(alertActivation == true) then alertTone <- database.getAlertTone(); showAlert(scheduleCount); endif endif end </pre>

Tabel 94. Spesifikasi proses showAlert

No.	SP - 42
Nama Method	showAlert
Nama Class	Alert
Requirement	[REQ F - 03]
Deskripsi	Method ini digunakan untuk menampilkan <i>alert</i> sesuai dengan <i>schedule</i> yang akan dilaksanakan
Return	-
Parameter	scheduleCount : Integer
Method yang terkait	setTitle(), setPositiveButton(), setNegativeButton(), show()
Logika Proses	<pre> procedure showAlert(scheduleCount : Integer) var alert : AlertDialog.Builder //sebagai penampung inisiasi class AlertDialog detail_schedule : Intent //sebagai penampung inisiasi class Intent begin alert <- new Builder(); //mengeset informasi yang akan ditampilkan pada alert alert.setTitle(scheduleCount + "schedule(s) will be sent"); //mengeset button yang akan ditampilkan pada alert alert.setPositiveButton("Yes"); alert.setNeutralButton("See Detail"); alert.setNegativeButton("No"); alert.show(); end endprocedure </pre>

4.2.2.1.11 Sequence Diagram Membuat Template



Gambar 26 . Sequence diagram membuat template

Tabel 95. Keterangan *sequence diagram* membuat *template*

No.	SD - 13
Requirement	[REQ TMP - 06], [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	MainActivity, Template, NewTemplate, Button, EditText, RadioButton, TemplateDatabaseHelper
Method yang terkait	onItemClick(), onCreate(), registerForContextMenu(), getText(), openDialogCategory(), getCheckedRadioButtonId(), saveTemplateToCategory(), getCategoryID(), saveTemplateToTemplate()
Logika Proses	<pre> databaseHelper : TemplateDatabaseHelper //sebagai penampung inisiasi class TemplateDatabaseHelper 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 create new tmp_message; create new tmp_name; create new data_count; //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 96. Spesifikasi proses saveTemplateToCategory

No.	SP - 43
Nama Method	saveTemplateToCategory
Nama Class	TemplateDatabaseHelper
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	getWritableDatabase(), put(), insert()
Logika Proses	<pre> procedure saveTemplateToCategory(data : String) openHelper : TemplateDatabaseHelper //sebagai penampung inisiasi class TemplateDatabaseHelper database : SQLiteDatabase //sebagai penampung inisiasi class SQLiteDatabase contentValues : ContentValues //sebagai penampung inisiasi class ContentValues begin //sebagai inisiasi method untuk melakukan query write (tulis) ke database database <- openHelper.getWritableDatabase(); contentValues <- new ContentValues; contentValues.put(CATEGORY_COLUMN_TYPE, data); database.insert(TABLE_CATEGORY, null, contentValues); end endprocedure </pre>

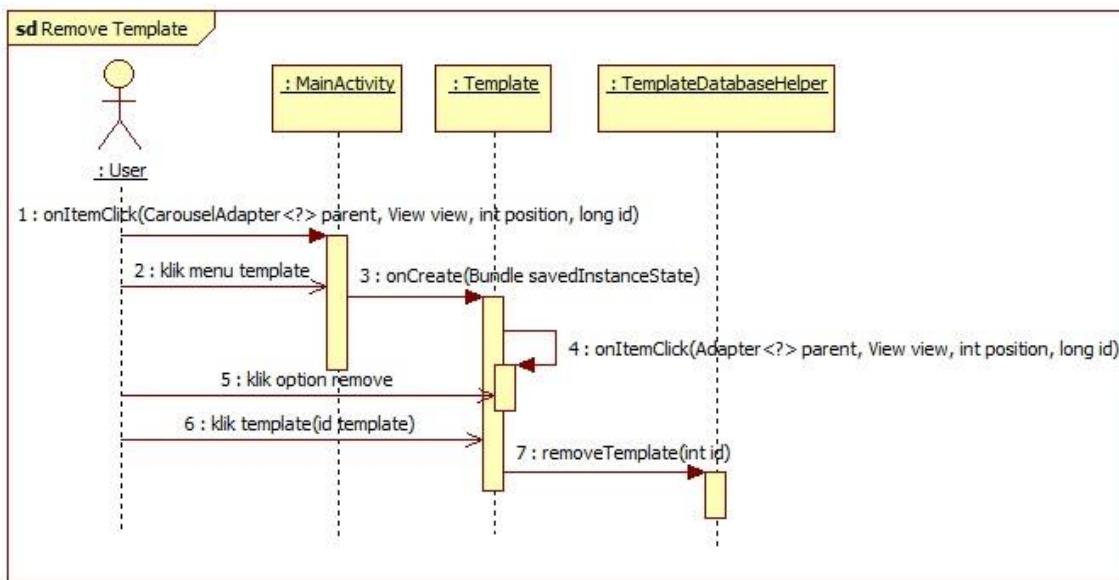
Tabel 97. Spesifikasi proses getCategoryId

No.	SP - 44
Nama Method	getCategoryId
Nama Class	TemplateDatabaseHelper
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	getWritableDatabase(), put(), insert()
Logika Proses	<pre> procedure saveTemplatetoCategory(data : String) openHelper : TemplateDatabaseHelper //sebagai penampung inisiasi class TemplateDatabaseHelper database : SQLiteDatabase //sebagai penampung inisiasi class SQLiteDatabase contentValues : ContentValues //sebagai penampung inisiasi class ContentValues begin //sebagai inisiasi method untuk melakukan query write (tulis) ke database database <- openHelper.getWritableDatabase(); contentValues <- new ContentValues; contentValues.put(CATEGORY_COLUMN_TYPE, data); database.insert(TABLE_CATEGORY, null, contentValues); end endprocedure </pre>

Tabel 98. Spesifikasi proses saveTemplatetoTemplate

No.	SP - 45
Nama Method	saveTemplatetoTemplate
Nama Class	TemplateDatabaseHelper
Requirement	[REQ TMP - 04]
Deskripsi	Method ini digunakan untuk <i>template</i> yang dibuat ke tabel Template 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) openHelper : TemplateDatabaseHelper //sebagai penampung inisiasi class TemplateDatabaseHelper database : SQLiteDatabase //sebagai penampung inisiasi class SQLiteDatabase contentValues : ContentValues //sebagai penampung inisiasi class ContentValues begin TemplateDatabaseHelper openHelper <- new TemplateDatabaseHelper(); SQLiteDatabase database <- new SQLiteDatabase(); ContentValues contentValues <- ContentValues(); //sebagai inisiasi method untuk melakukan query write (tulis) ke database database <- openHelper.getWritableDatabase(); contentValues <- new ContentValues(); contentValues.put(TEMPLATE_COLUMN_CATEGORY_ID, categoryID); contentValues.put(TEMPLATE_COLUMN_MESSAGE, data[1]); contentValues.put(CATEGORY_COLUMN_TYPE, data[2]); contentValues.put(TEMPLATE_COLUMN_NAME, data[3]); database.insert(TABLE_TEMPLATE, null, contentValues); end endprocedure </pre>

4.2.2.1.12 Sequence Diagram Menghapus Template



Gambar 27 . Sequence diagram menghapus template

Keterangan dari sequence diagram menghapus template pada gambar 27, dijelaskan pada tabel 99 - 100.

Tabel 99. Keterangan sequence diagram menghapus template

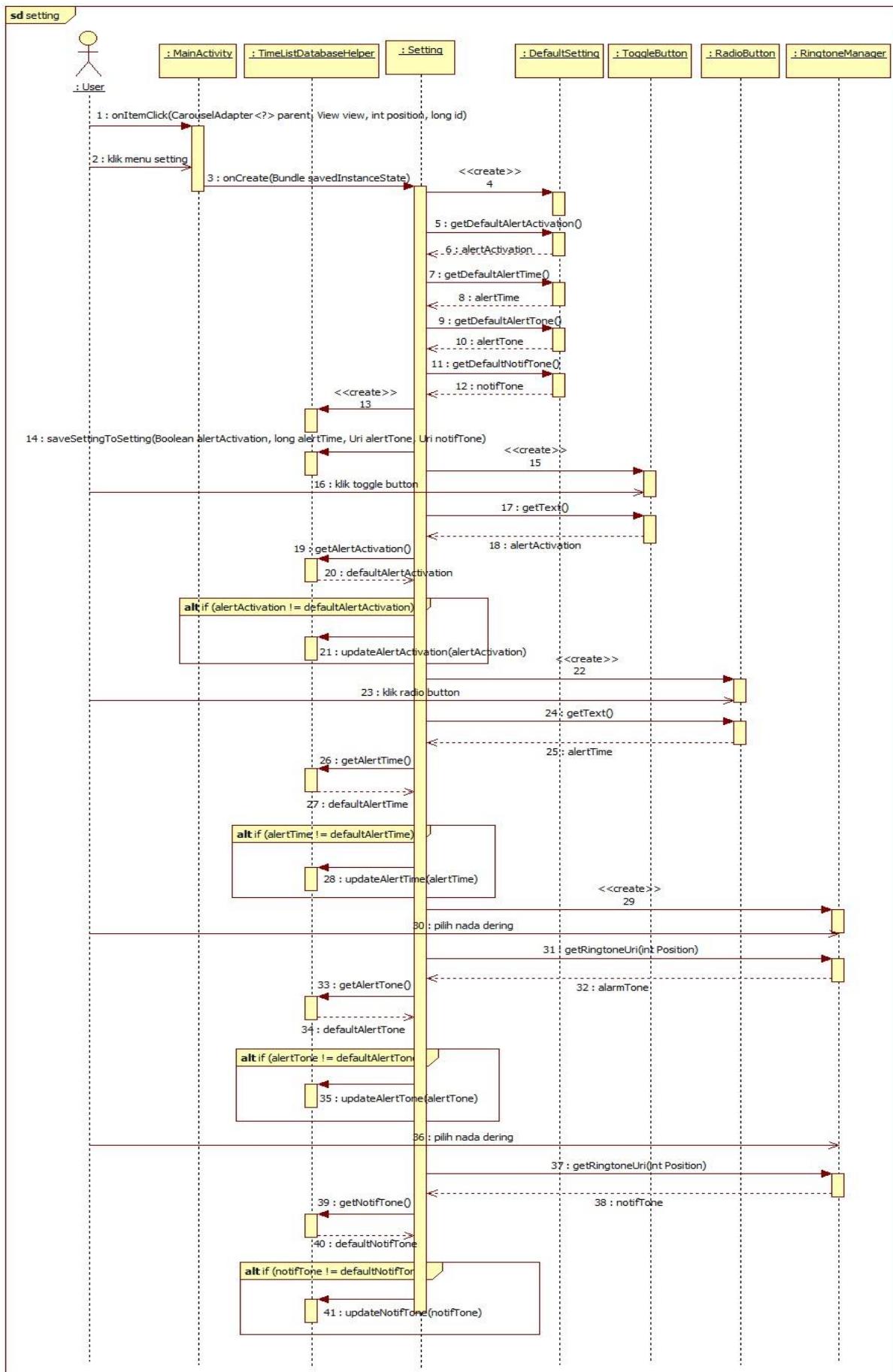
No.	SD - 14
Requirement	[REQ TMP - 06], [REQ TMP - 07]
Deskripsi	Sequence ini menggambarkan proses untuk menghapus template yang sebelumnya telah dibuat
Class yang terkait	MainActivity, Template, TemplateDatabaseHelper
Method yang terkait	onItemClick(), onCreate(), removeTemplate()
Logika Proses	<pre> openHelper : TimeListDatabaseHelper //sebagai penampung inisiasi class TimeListDatabaseHelper cursor : Cursor //sebagai penampung inisiasi class Cursor templateId : ArrayList of Integer //sebagai penampung id template yang sudah tersimpan di database id : Integer //sebagai penampung id template yang akan dihapus begin TimeListDatabaseHelper openHelper <- new TimeListDatabaseHelper(); Cursor cursor <- new Cursor(); onContextItemSelected(item); cursor <- openHelper.getTemplateList(); if(cursor.moveToFirst()) then do templateId.add(cursor.getColumnIndex(TimeListDatabaseHelper.TEMPLATE_COLUMN_ID)); while (cursor.moveToNext()) endif id <- templateId.get(position); removeTemplate(id); end </pre>

Tabel 100. Spesifikasi proses removeTemplate

No.	SP - 46
Nama Method	removeTemplate
Nama Class	TemplateDatabaseHelper
Requirement	[REQ TMP - 07]
Deskripsi	Method ini digunakan untuk menghapus template dengan id template tertentu dari database

No.	SP - 46
Nama Method	removeTemplate
Return	Boolean
Parameter	id : Integer
Method yang terkait	delete()
Logika Proses	
<pre> function removeTemplate(id : Integer) database : SQLiteDatabase //sebagai penampung inisiasi class SQLiteDatabase begin try database.delete(TABLE_TEMPLATE, TEMPLATE_COLUMN_ID + "=" + id, null); database.delete(TABLE_CATEGORY, CATEGORY_COLUMN_ID + "=" + TEMPLATE_COLUMN_CATEGORY_ID, null); return true; end try catch(SQLException sqe) return false; endcatch end endfunction </pre>	

4.2.2.1.13 Sequence Diagram Setting



Gambar 28 . Sequence diagram setting

Keterangan dari *sequence diagram setting* pada gambar 28, dijelaskan pada tabel 101 - 115.

Tabel 101. Keterangan *sequence diagram setting*

No.	SD - 15
Requirement	[REQ F - 05], [REQ F - 05.1], [REQ F - 05.2], [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	MainActivity, DefaultSetting, SettingDatabaseHelper, Setting, ToggleButton, RingtoneManager, Spinner
Method yang terkait	onItemClick(), onCreate(), getDefaultUri(), add(), saveToSetting(), getText(), 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 : TimeListDatabaseHelper //sebagai penampung inisiasi class TimeListDatabaseHelper toggle_button : ToggleButton //sebagai penampung inisiasi class ToggleButton radio_button : RadioButton //sebagai penampung inisiasi class RadioButton ringtone_manager : RingtoneManager //sebagai penampung inisiasi class RingtoneManager 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 TimeListDatabaseHelper(); helper.saveSettingToSetting(alertActivation, alertTime, alertTone, notifTone); //mengambil aktivasi alert yang diinputkan user toggle_button <- new ToggleButton(); alertActivation <- toggle_button.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 radio_button <- new RadioButton(); alertTime <- radio_button.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 ringtone_manager <- new RingtoneManager(); alertTone <- ringtone_manager.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 <- ringtone_manager.getRingtoneUri(position); //mengambil nada dering notification yang sebelumnya sudah tersimpan di database defaultNotifTone <- helper.getDefaultNotifTone(); </pre>

No.	SD - 15
	<pre>//membandingkan nada dering notification if(defaultNotifTone != notifTone) then helper.updateNotifTone(notifTone, defaultNotifTone); end end</pre>

Tabel 102. Spesifikasi proses getDefaultAlertActivation

No.	SP - 47
Nama Method	getDefaultAlertActivation
Nama Class	DefaultSetting
Deskripsi	Method ini digunakan untuk mengambil <i>default setting</i> untuk aktivasi <i>alert</i> dari class DefaultSetting
Requirement	[REQ SA - 01]
Return	alertActivation : Boolean
Parameter	-
Method yang terkait	-
Logika Proses	<pre>function getDefaultAlertActivation() begin return alertActivation; end endfunction</pre>

Tabel 103. Spesifikasi proses getDefaultAlertTime

No.	SP - 48
Nama Method	getDefaultAlertTime
Nama Class	DefaultSetting
Deskripsi	Method ini digunakan untuk mengambil <i>default setting</i> untuk waktu ketika menampilkan <i>alert</i> dari class DefaultSetting
Requirement	[REQ SA - 02]
Return	alertTime : long
Parameter	-
Method yang terkait	-
Logika Proses	<pre>function getDefaultAlertTime() begin return alertTime; end endfunction</pre>

Tabel 104. Spesifikasi proses getDefaultAlertTone

No.	SP - 49
Nama Method	getDefaultAlertTone
Nama Class	DefaultSetting
Deskripsi	Method ini digunakan untuk mengambil <i>default setting</i> untuk nada dering <i>alert</i> dari class DefaultSetting
Requirement	[REQ SA - 03]
Return	alertTone : Uri
Parameter	-
Method yang terkait	-
Logika Proses	<pre>function getDefaultAlertTone() begin return alertTone; end endfunction</pre>

Tabel 105. Spesifikasi proses getDefaultNotifTone

No.	SP - 50
Nama Method	getDefaultNotifTone
Nama Class	DefaultSetting
Deskripsi	Method ini digunakan untuk mengambil <i>default setting</i> untuk aktivasi <i>notification</i> dari class DefaultSetting
Requirement	[REQ SN - 01]
Return	notifTone : Uri
Parameter	-
Method yang terkait	-
Logika Proses	<pre>function getDefaultNotifTone() begin return notifTone; end endfunction</pre>

Tabel 106. Spesifikasi proses saveSettingToSetting

No.	SP - 51
Nama Method	saveSettingToSetting
Nama Class	TimeListDatabaseHelper
Deskripsi	Method ini digunakan untuk menyimpan <i>default setting</i> yang didapatkan dari class DefaultSetting. Tabel yang terkait dengan method ini adalah tabel Setting.
Requirement	[REQ SA - 05]
Return	-
Parameter	alertActivation : Boolean, alertTime : long, alertTone, notifTone : Uri
Method yang terkait	getWritableDatabase() rawQuery()
Logika Proses	<pre>procedure saveSettingToSetting(alertActivation : Boolean, alertTime : Long, alertTone : Uri, notifTone : Uri) var openHelper : DatabaseOpenHelper databaseWriteable : TimeListDatabaseHelper contentValues : ContentValues, //sebagai inisiasi terhadap class ContentValues begin databaseWriteable <- openHelper.getWritableDatabase(); //sebagai inisiasi method untuk melakukan query write (tulis) ke database contentValues.put("alert_activation", alertActivation); contentValues.put("alert_time", alertTime); contentValues.put("alert_tone", alertTone); contentValues.put("notif_tone", notifTone); //menyimpan setting ke tabel setting di database databaseWriteable.insert("setting", null, contentValues); end endprocedure</pre>

Tabel 107. Spesifikasi proses getAlertActivation

No.	SP - 52
Nama Method	getAlertActivation
Nama Class	TimeListDatabaseHelper
Deskripsi	Method ini digunakan untuk mengambil aktivasi <i>alert</i> yang sudah tersimpan di <i>database</i> . Tabel yang terkait dengan method ini adalah tabel Setting.
Requirement	[REQ SA - 05]
Return	defaultAlertActivation : boolean
Parameter	-
Method yang terkait	-
Logika Proses	<pre>function getAlertActivation() begin</pre>

No.	SP – 52
Nama Method	getAlertActivation
	<pre> return defaultAlertActivation; end endfunction</pre>

Tabel 108. Spesifikasi proses updateAlertActivation

No.	SP - 53
Nama Method	updateAlertActivation
Nama Class	TimeListDatabaseHelper
Deskripsi	Method ini digunakan untuk menyimpan perubahan aktivasi <i>alert</i> ke <i>database</i> . Tabel yang terkait dengan <i>method</i> ini adalah tabel Setting.
Requirement	[REQ SA - 05]
Return	-
Parameter	alertActivation : boolean
Method yang terkait	getWritableDatabase() rawQuery()
Logika Proses	<pre>procedure updateAlertActivation(alertActivation : Boolean) var openHelper : DatabaseOpenHelper databaseWriteable : TimeListDatabaseHelper begin //sebagai inisiasi method untuk melakukan query write (tulis) ke database databaseWriteable <- openHelper.getWritableDatabase() //melakukan update aktivasi alert sesuai inputan user ke tabel setting databaseWriteable.rawQuery("update setting set alert_activation='alertActivation'"); end endprocedure</pre>

Tabel 109. Spesifikasi proses getAlertTime

No.	SP - 54
Nama Method	getAlertTime
Nama Class	TimeListDatabaseHelper
Deskripsi	Method ini digunakan untuk mengambil waktu untuk menampilkan <i>alert</i> yang sudah tersimpan di <i>database</i> . Tabel yang terkait dengan <i>method</i> ini adalah tabel Setting.
Requirement	[REQ SA - 05]
Return	defaultAlertTime : long
Parameter	-
Method yang terkait	-
Logika Proses	<pre>function getAlertTime() begin return defaultAlertTime; end endfunction</pre>

Tabel 110. Spesifikasi proses updateAlertTime

No.	SP - 55
Nama Method	updateAlertTime
Nama Class	TimeListDatabaseHelper
Deskripsi	Method ini digunakan untuk menyimpan perubahan waktu untuk menampilkan <i>alert</i> ke <i>database</i> . Tabel yang terkait dengan <i>method</i> ini adalah tabel Setting.
Requirement	[REQ SA - 05]
Return	-
Parameter	alertTime : long
Method yang terkait	getWritableDatabase() rawQuery()

No.	SP - 55
Nama Method	updateAlertTime
Logika Proses	
<pre> procedure updateAlertTime(alertTime : long) var openHelper : DatabaseOpenHelper databaseWriteable : TimeListDatabaseHelper begin //sebagai inisiasi method untuk melakukan query write (tulis) ke database databaseWriteable <- openHelper.getWritableDatabase() //melakukan update waktu alert sesuai inputan user ke tabel setting databaseWriteable.rawQuery("update setting set alert_time='alertTime'"); end endprocedure </pre>	

Tabel 111. Spesifikasi proses getAlertTone

No.	SP - 56
Nama Method	getAlertTone
Nama Class	TimeListDatabaseHelper
Deskripsi	Method ini digunakan untuk mengambil nada dering <i>alert</i> yang sudah tersimpan di <i>database</i> . Tabel yang terkait dengan <i>method</i> ini adalah tabel Setting.
Requirement	[REQ SA - 05]
Return	defaultAlertTone : uri
Parameter	-
Method yang terkait	-
Logika Proses	
<pre> function getAlertTone() begin return defaultAlertTone; end endfunction </pre>	

Tabel 112. Spesifikasi proses updateAlertTone

No.	SP - 57
Nama Method	updateAlertTone
Nama Class	TimeListDatabaseHelper
Deskripsi	Method ini digunakan untuk menyimpan perubahan nada dering <i>alert</i> ke <i>database</i> . Tabel yang terkait dengan <i>method</i> ini adalah tabel Setting.
Requirement	[REQ SA - 05]
Return	-
Parameter	alertTone : Uri
Method yang terkait	getWritableDatabase() rawQuery()
Logika Proses	
<pre> procedure updateAlertTone(alertTone : Uri) var openHelper : DatabaseOpenHelper databaseWriteable : TimeListDatabaseHelper begin //sebagai inisiasi method untuk melakukan query write (tulis) ke database databaseWriteable <- openHelper.getWritableDatabase(); //melakukan update nada dering alert sesuai inputan user ke tabel setting databaseWriteable.rawQuery("update setting set alert_tone='alertTone'"); end endprocedure </pre>	

Tabel 113. Spesifikasi proses getNotifTone

No.	SP - 58
Nama Method	getNotifTone
Nama Class	TimeListDatabaseHelper
Deskripsi	Method ini digunakan untuk mengambil nada dering notification yang sudah tersimpan di database. Tabel yang terkait dengan method ini adalah tabel Setting.
Requirement	[REQ SA - 05]
Return	defaultNotifTone : uri
Parameter	-
Method yang terkait	-
Logika Proses	<pre>function getNotifTone() begin return defaultNotifTone; end endfunction</pre>

Tabel 114. Spesifikasi proses updateNotifTone

No.	SP - 59
Nama Method	updateNotifTone
Nama Class	TimeListDatabaseHelper
Deskripsi	Method ini digunakan untuk menyimpan perubahan nada dering notification ke database. Tabel yang terkait dengan method ini adalah tabel Setting.
Requirement	[REQ SA - 05]
Return	-
Parameter	notifTone : Uri
Method yang terkait	getWritableDatabase() rawQuery()
Logika Proses	<pre>procedure updateNotifTone(notifTone : Uri) var openHelper : DatabaseOpenHelper //sebagai penampung inisiasi class DatabaseOpenHelper databaseWriteable : TimeListDatabaseHelper //sebagai penampung inisiasi class TimeListDatabaseHelper begin //sebagai inisiasi method untuk melakukan query write (tulis) ke database databaseWriteable <- openHelper.getWritableDatabase(); //melakukan update nada dering notification sesuai inputan user ke tabel setting databaseWriteable.rawQuery("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 36 sampai dengan tabel 156.

Tabel 115. 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>UI - 01.1 : UI New Schedule untuk platform Android</p>
UI - 01.2 : UI New Schedule untuk platform Blackberry	

Tabel 116. Rancangan UI halaman *New Schedule (Form New Schedule)* (Lanjutan)

No.	UI – 01
Nama	<i>New Schedule (Form New Schedule)</i>
Gambar	
UI - 01.3 : UI <i>New Schedule</i> untuk platform iOS	
	<p>Jika user menginputkan string khusus, yaitu birthday's date atau anniversary's date</p>
UI - 01.4 : UI <i>New Schedule</i> untuk platform Java/Symbian	
	<p>Jika user menginputkan string khusus, yaitu birthday's date atau anniversary's date</p>

Tabel 117. Rancangan UI halaman *New Schedule (Form New Schedule)* (Lanjutan)

No. UI – 01

Nama New Schedule (Form New Schedule)

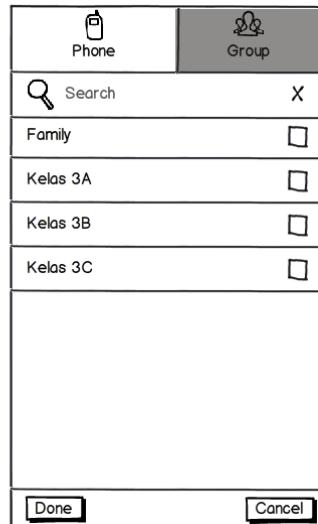
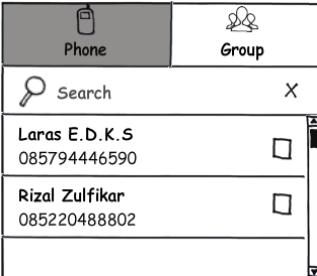
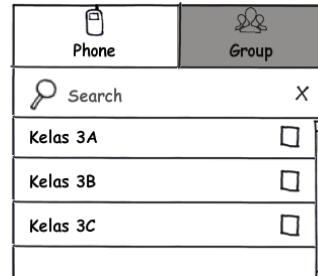
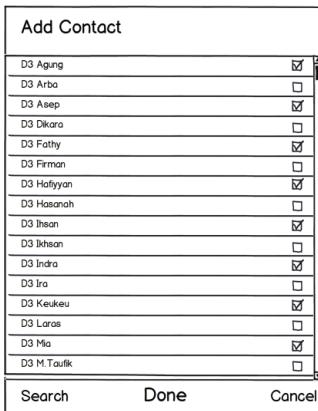
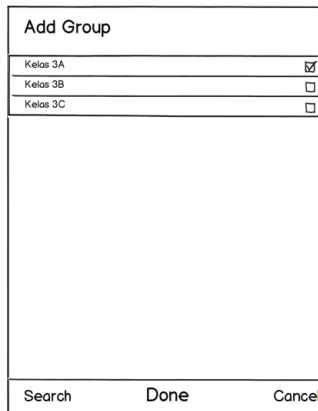
Keterangan :

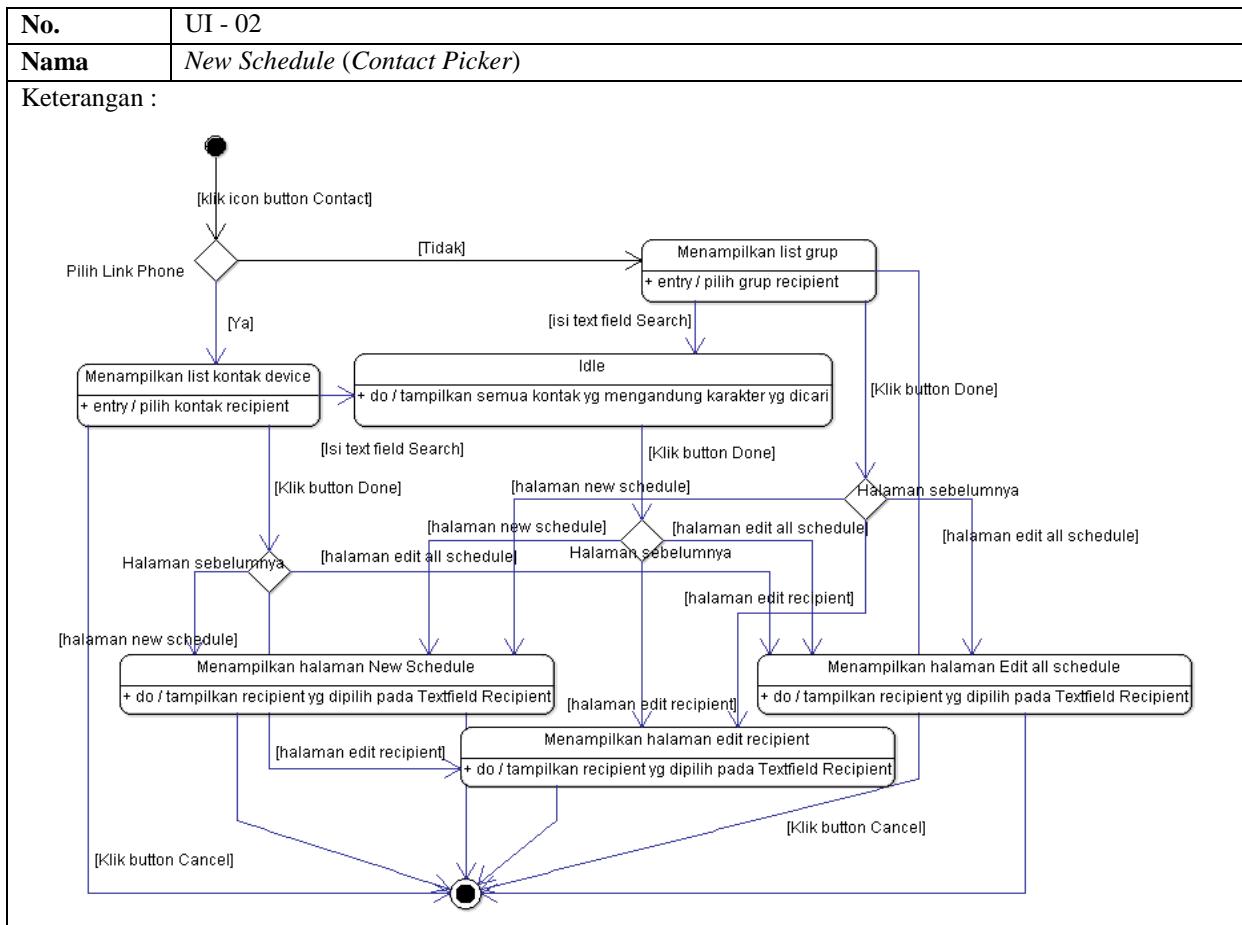
```

graph TD
    Start(( )) -- "[Klik menu New Schedule]" --> Form[Menampilkan form new schedule  
+ do / menampilkan form]
    Form -- "[Klik button Ok]" --> InvalidSchedule[Menampilkan halaman invalid schedule  
+ do / menampilkan alert box invalid schedule]
    InvalidSchedule -- "Tidak" --> MengisiForm[Mengisi form  
+ entry / kontak recipient  
+ entry / tanggal dan waktu pengiriman  
+ entry / frequency pengiriman  
+ entry / content message]
    MengisiForm -- "[Klik button Ok]" --> ScheduleTimeExpired[Menampilkan halaman schedule time expired  
+ do / menampilkan alert box time schedule expired]
    ScheduleTimeExpired -- "Tidak" --> Save{waktu schedule valid}
    Save -- Ya --> Simpan[Simpan schedule]
    Simpan --> ListScheduled[Menampilkan halaman list scheduled]
    ListScheduled -- "[klik button Cancel]" --> End(( ))
    Save -- Tidak --> MengisiForm
    MengisiForm -- "[klik icon button Contact]" --> ContactPicker[Menampilkan halaman Contact Picker  
+ entry / memilih recipient]
    ContactPicker -- "[klik button Done]" --> MengisiRecipient[Mengisi kontak recipient  
+ entry / menulis recipient secara manual  
+ entry / mengambil recipient dari list kontak]
    MengisiRecipient -- "[klik button Done]" --> MengisiForm
    MengisiForm -- "[klik text field Message]" --> ContentMessage[Mengisi content message  
+ entry / menulis pesan secara manual  
+ entry / mengambil pesan dari template]
    ContentMessage -- "[klik icon button Set]" --> DateTimePicker[Menampilkan halaman DateTime Picker  
+ entry / pilih tanggal dan waktu pengiriman]
    DateTimePicker -- "[klik button Done]" --> MengisiForm
    MengisiForm -- "[klik combo box Frequency]" --> Frequency[Menampilkan frequency  
+ entry / pilih frequency pengiriman]
    Frequency -- "[klik template atau button Done]" --> Template[Menampilkan halaman template  
+ entry / pilih template]
    Template -- "[klik icon button Template]" --> ContentMessage
  
```

The flowchart details the process of creating a new schedule. It starts with clicking the 'New Schedule' menu, which displays the 'New Schedule' form. If the form is incomplete (indicated by 'form terisi lengkap' as 'Tidak'), an invalid schedule page is shown. If the form is complete ('Ya'), it checks if the scheduled time is valid. If valid, the schedule is saved, leading to a page showing the list of scheduled tasks. If the scheduled time is invalid, a page showing an expired schedule is displayed. The process then moves to filling the form. This involves selecting a recipient (using a contact picker), entering a message (with options for manual entry or template selection), setting a frequency (using a date-time picker), and choosing a delivery time. Finally, a template selection screen is shown before returning to the message entry step.

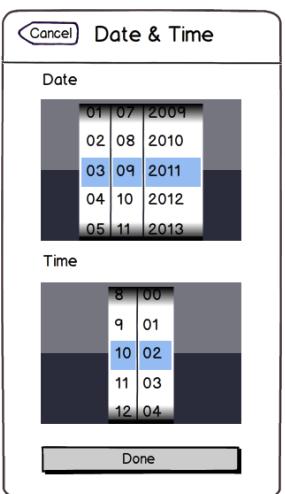
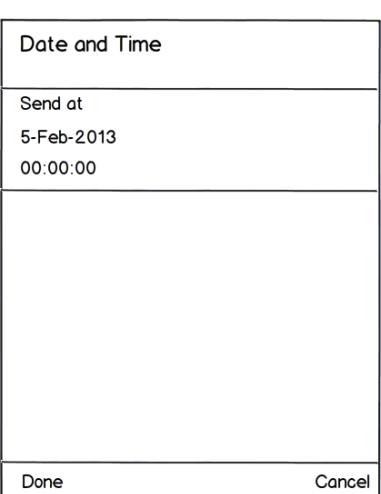
Tabel 118. Rancangan UI halaman *New Schedule (Contact Picker)*

No.	UI - 02	
Nama	<i>New Schedule (Contact Picker)</i>	
Deskripsi	Halaman untuk memilih recipient dari kontak yang terintegrasi dengan device	
Gambar	 	UI - 02.1 : UI <i>New Schedule (Contact Picker)</i> untuk platform Android dan iOS
	  A BlackBerry key	UI - 02.2 : UI <i>New Schedule (Contact Picker)</i> untuk platform BlackBerry
	 	UI - 02.3 : UI <i>New Schedule (Contact Picker)</i> untuk platform Java/Symbian

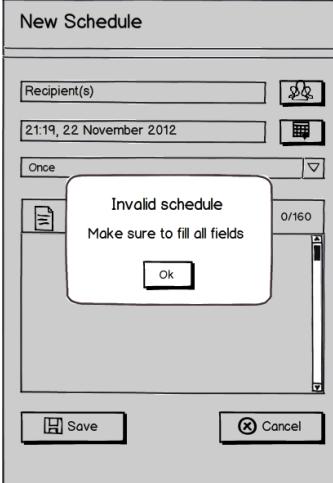
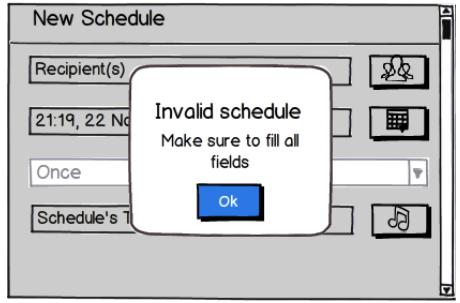
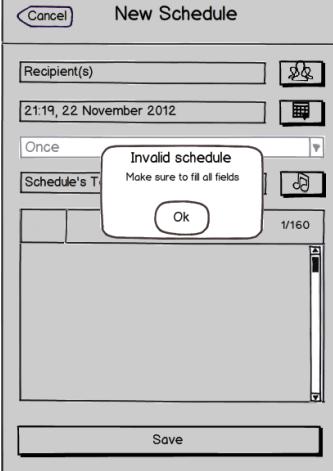
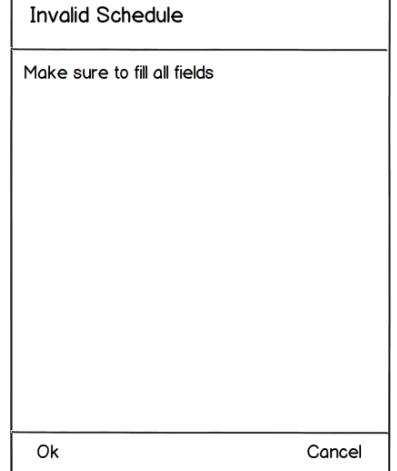
Tabel 119. Rancangan UI halaman *New Schedule (Contact Picker)* (Lanjutan)Tabel 120. Rancangan UI halaman *New Schedule* (Menampilkan Pengaturan Tanggal dan Waktu)

No.	UI - 03
Nama	<i>New Schedule</i> (Menampilkan pengaturan tanggal dan waktu)
Deskripsi	Halaman untuk mengatur waktu dan tanggal pengiriman pesan.
Gambar	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>UI - 03.1 : UI Date and Time picker untuk platform Android</p> </div> <div style="text-align: center;"> <p>UI - 03.2 : UI Date and Time picker untuk platform BlackBerry</p> </div> </div>

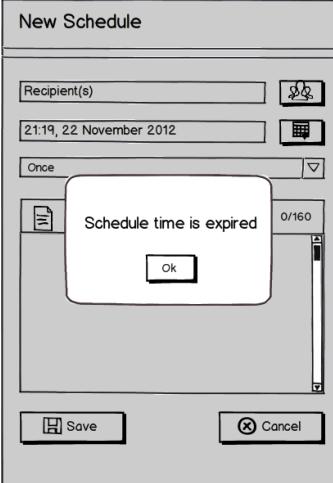
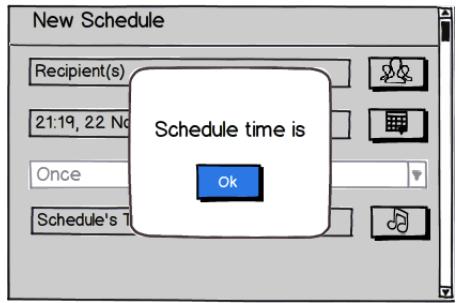
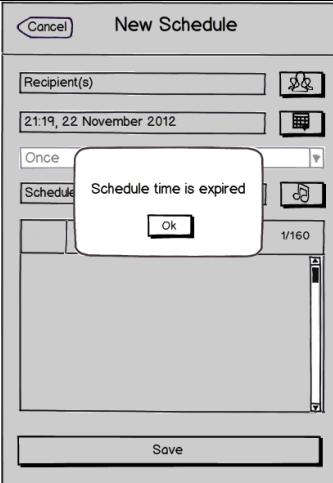
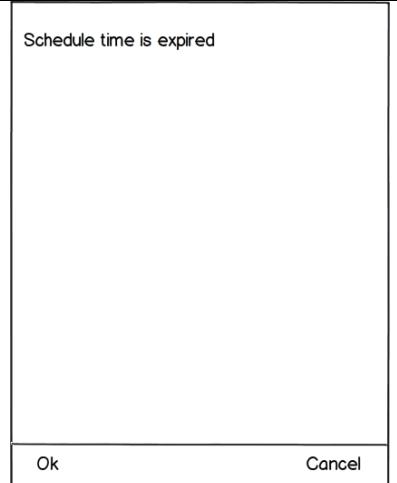
Tabel 121. Rancangan UI halaman *New Schedule* (Menampilkan Pengaturan Tanggal dan Waktu) (Lanjutan)

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</p>  <p>UI - 03.4 : UI Date and Time picker untuk platform Java/Symbian</p>
Keterangan :	<pre> graph TD Start(()) -- "klik icon button Set" --> DT_Picker[Menampilkan Date and Time Picker + entry / tanggal pengiriman pesan + entry / waktu pengiriman pesan] DT_Picker -- "[klik button Cancel]" --> End(()) DT_Picker -- "[klik button Done]" --> Decision{ } Decision -- "[menu edit date time schedule]" --> EditAll[Menampilkan halaman Edit All Schedule + do / tampilan tanggal dan waktu pengiriman pada Text field DateTime] Decision -- "[halaman edit all schedule]" --> ScheduledList[Menampilkan halaman list scheduled] EditAll -- "[halaman new schedule]" --> NewSchedule[Menampilkan halaman New Schedule + do / tampilan tanggal dan waktu pengiriman pada Text field DateTime] </pre>

Tabel 122. 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 user belum mengisi semua <i>field</i> pada <i>form new schedule</i>
Gambar	 <p>UI - 04.1 : UI warning invalid schedule untuk platform Android</p>  <p>UI - 04.2 : UI warning invalid schedule untuk platform BlackBerry</p>  <p>UI - 04.1 : UI warning invalid schedule untuk platform iOS</p>  <p>UI - 04.1 : UI warning invalid schedule untuk platform Java/Symbian</p>
Keterangan	<pre> graph TD Start(()) -- "[form schedule belum lengkap]" --> Alert[Menampilkan halaman invalid schedule
+ do / menampilkan alert box invalid schedule] Alert -- "[klik button Ok]" --> HalamanSebelumnya[Halaman sebelumnya] HalamanSebelumnya -- "[halaman new schedule]" --> FormNewSchedule[Menampilkan form new schedule
+ do / menampilkan form] HalamanSebelumnya -- "[halaman edit all schedule]" --> FormEditAllSchedule[Menampilkan form edit all schedule
+ do / menampilkan form yang berisi data schedule] </pre>

Tabel 123. 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 user salah memasukkan tanggal atau waktu schedule
Gambar	 <p>UI - 05.1 : UI warning schedule time expired untuk platform Android</p>
	 <p>UI - 05.2 : UI warning schedule time expired untuk platform Blackberry</p>
	 <p>UI - 05.1 : UI warning schedule time expired untuk platform iOS</p>
	 <p>UI - 05.1 : UI warning schedule time expired untuk platform Java/Symbian</p>
Keterangan	<pre> graph TD Start(()) -- "[waktu schedule tidak valid]" --> Alert[Menampilkan schedule time expired
+ do / menampilkan alert box time schedule expired] Alert -- "[klik button Ok]" --> Home(()) Alert -- "[halaman new schedule]" --> NewSchedule[Menampilkan form new schedule
+ do / menampilkan form] Alert -- "[halaman edit all schedule]" --> EditAllSchedule[Menampilkan form edit all schedule
+ do / menampilkan form yang berisi data schedule] NewSchedule --> Home EditAllSchedule --> Home </pre>

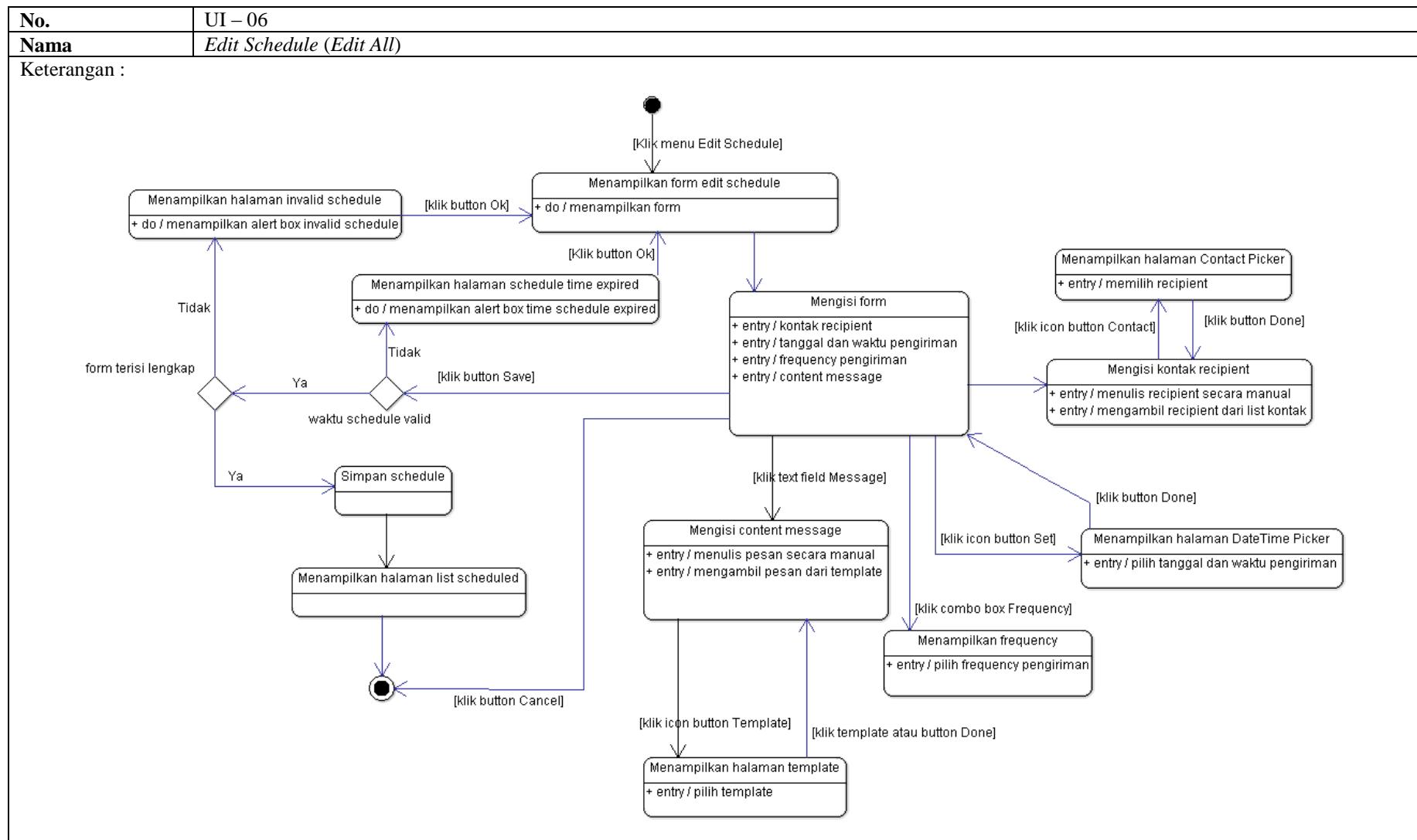
Tabel 124. Rancangan UI halaman *Edit Schedule (Edit All)*

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	

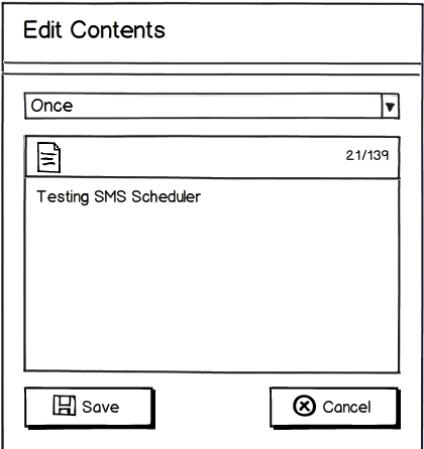
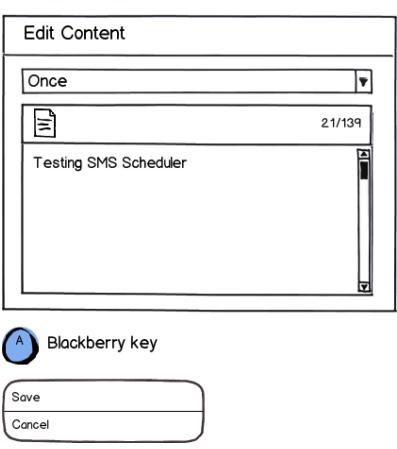
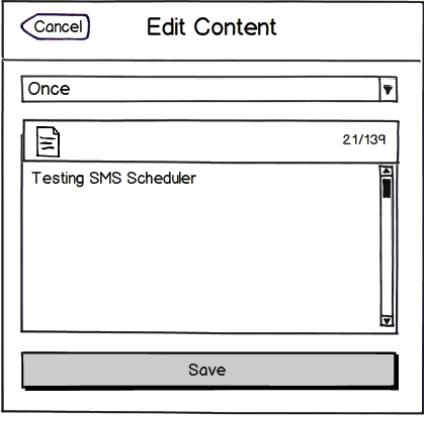
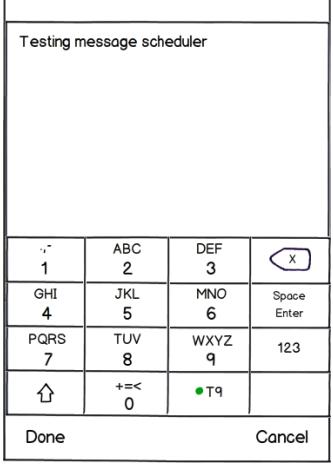
UI - 06.1 : UI Edit All Schedule untuk platform Android
UI - 06.2 : UI Edit All Schedule untuk platform Blackberry

Tabel 125. Rancangan UI halaman *Edit Schedule (Edit All)* (lanjutan)

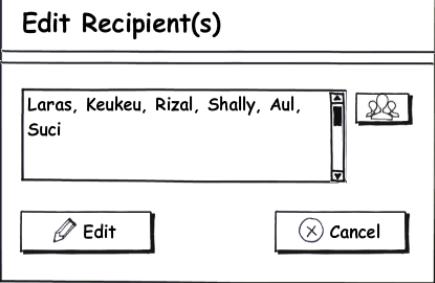
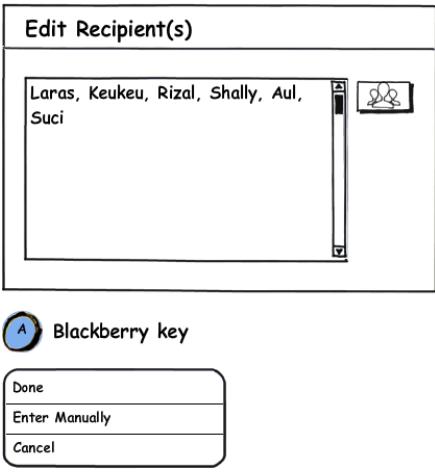
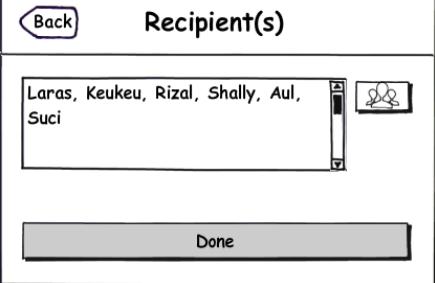
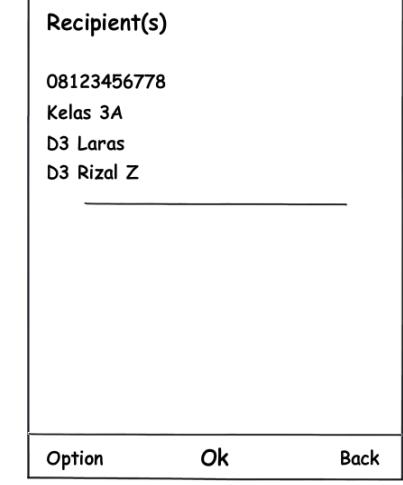
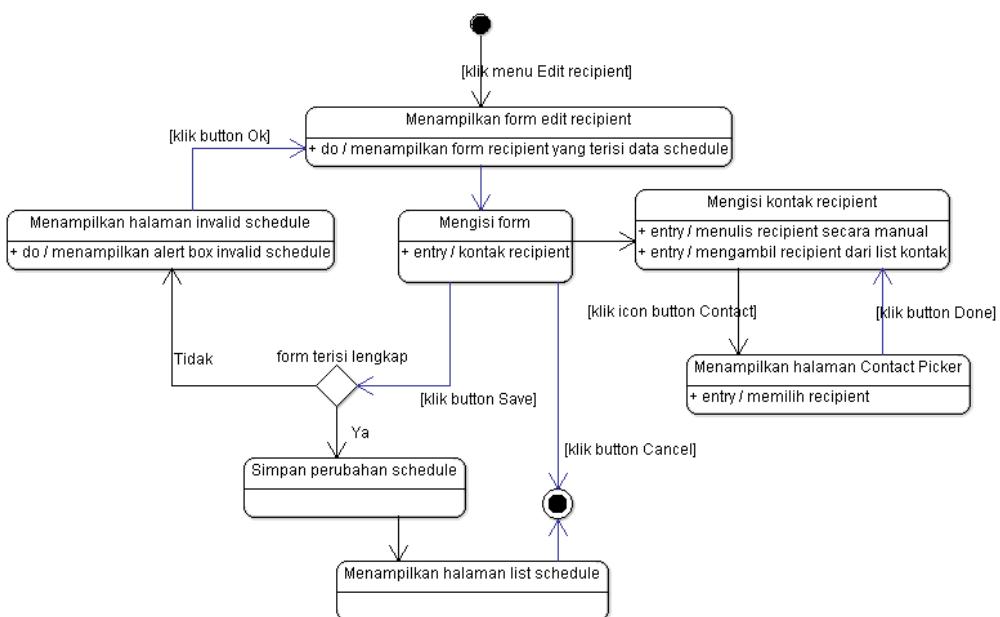
No.	UI - 06	
Nama	<i>Edit Schedule (Edit All)</i>	
Gambar	<p style="text-align: center;">UI - 06.3 : UI <i>Edit All Schedule</i> untuk platform iOS</p>	
<p style="text-align: center;">UI - 06.4 : UI <i>Edit All Schedule</i> untuk platform Java/Symbian</p>		

Tabel 126. Rancangan UI halaman *Edit Schedule (Edit All)* (Lanjutan)

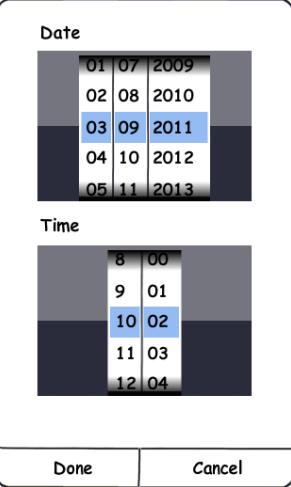
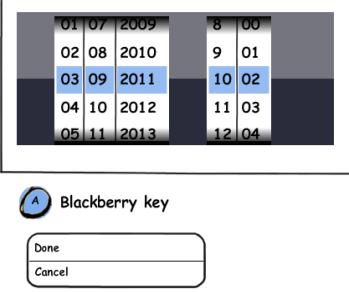
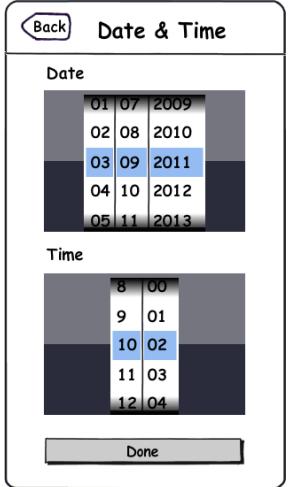
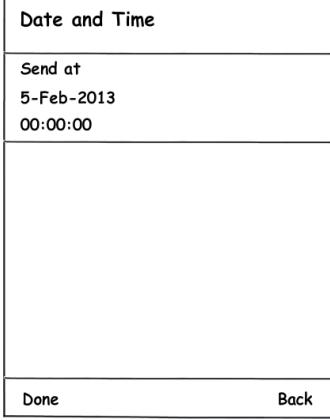
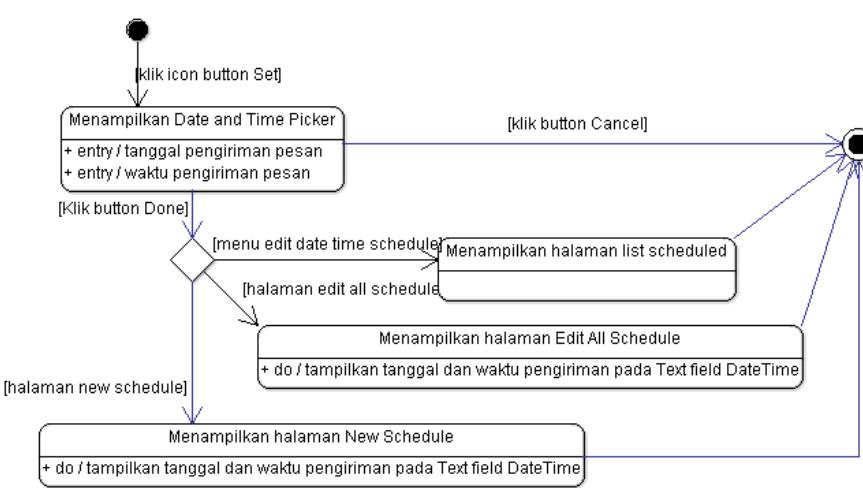
Tabel 127. Rancangan UI halaman *Edit Schedule (Edit Content)*

No.	UI - 07
Nama	<i>Edit Schedule (Edit Content)</i>
Deskripsi	Halaman untuk melakukan perubahan isi pesan dari jadwal yang sebelumnya telah dibuat.
Gambar	 <p>UI - 07.1 : UI <i>Edit Content</i> untuk platform Android</p>  <p>UI - 07.2 : UI <i>Edit Content</i> untuk platform Blackberry</p>
	 <p>UI - 07.3 : UI <i>Edit Content</i> untuk platform iOS</p>  <p>UI - 07.4 : UI <i>Edit Content</i> untuk platform Java/Symbian</p>
Keterangan	<pre> graph TD Start([Klik menu Edit content]) --> Form[Menampilkan form edit content] Form --> Mengisi[Mengisi form
+ entry / ringtone schedule
+ entry / frequency pengiriman
+ entry / content message] Mengisi --> Simpan[Simpan perubahan schedule] Simpan --> List[Menampilkan halaman list scheduled] Simpan -- "form terisi lengkap
Tidak" --> Invalid[Menampilkan halaman invalid schedule
+ do / menampilkan alert box invalid schedule] Invalid --> List Mengisi --> Done[Done] Done --> Template[Menampilkan halaman template
+ entry / pilih template] Mengisi --> Frequency[Menampilkan frequency
+ entry / pilih frequency pengiriman] Frequency --> Done </pre>

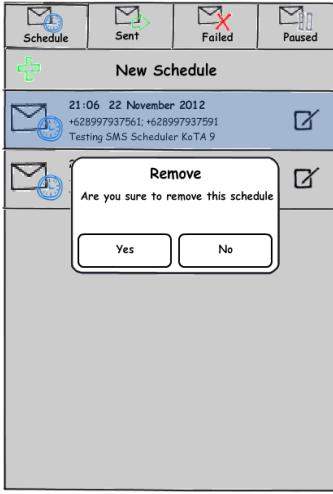
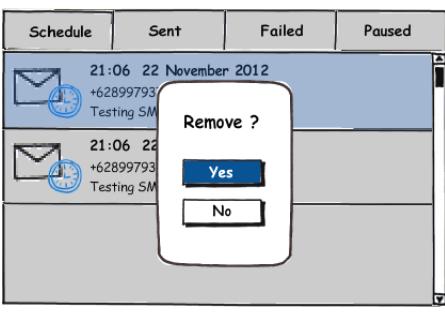
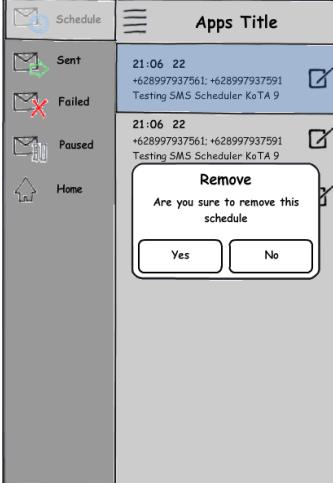
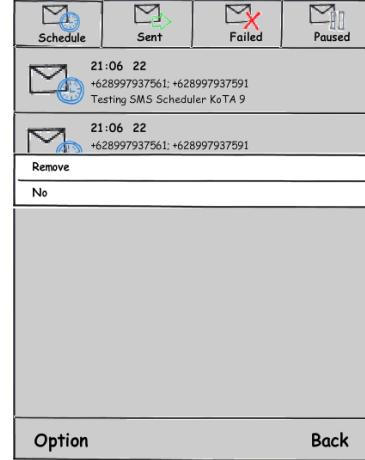
Tabel 128. Rancangan UI halaman *Edit Schedule (Edit Recipient)*

No.	UI - 08
Nama	Edit Schedule (<i>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 platform Android</p>
	 <p>UI - 08.2 : UI <i>Edit Recipient</i> untuk platform BlackBerry</p>
	 <p>UI - 08.3 : UI <i>Edit Recipient</i> untuk platform iOS</p>
	 <p>UI - 08.4 : UI <i>Edit Recipient</i> untuk platform Java/Symbian</p>
Keterangan	 <pre> graph TD Start(()) --> DisplayForm[Menampilkan form edit recipient] DisplayForm --> OK[+ klik button OK] OK --> InvalidSchedule[Menampilkan halaman invalid schedule + do / menampilkan alert box invalid schedule] OK --> Save[+ klik button Save] Save --> IncompleteForm{form terisi lengkap} IncompleteForm -- Tidak --> InvalidSchedule IncompleteForm -- Ya --> Simpan[Simpan perubahan schedule] Simpan --> ListSchedule[Menampilkan halaman list schedule] Contact[+ klik icon button Contact] --> ContactPicker[Menampilkan halaman Contact Picker + entry / memilih recipient] Done[+ klik button Done] --> ListSchedule </pre>

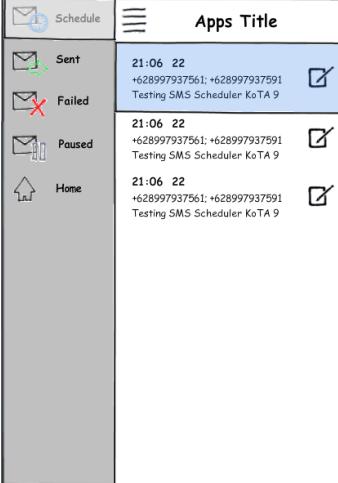
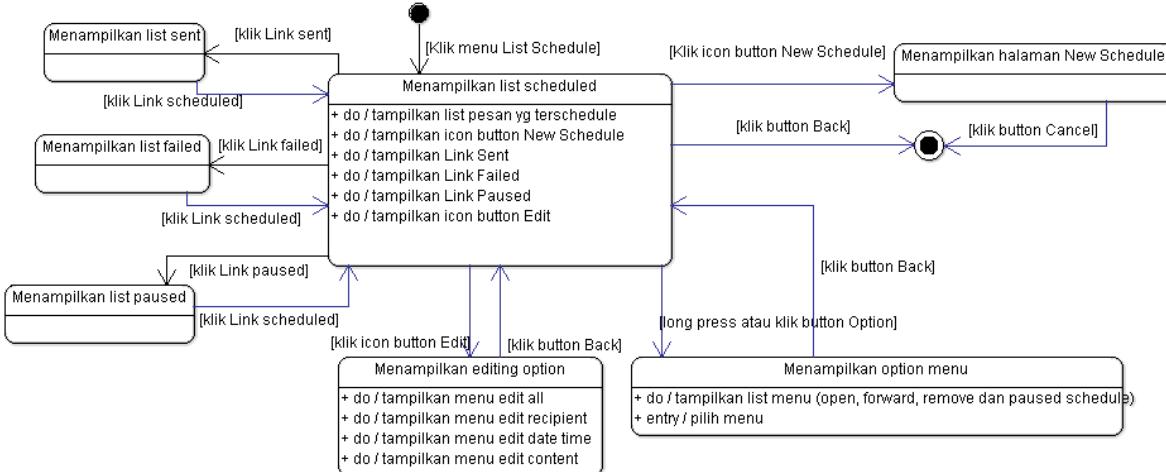
Tabel 129. Rancangan UI halaman *Edit Schedule (Edit Date and Time)*

No.	UI - 09	
Nama	Edit Schedule (<i>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	 UI - 09.2 : UI Edit Date and Time untuk platform Blackberry
	UI - 09.3 : UI Edit Date and Time untuk platform iOS	 UI - 09.4 : UI Edit Date and Time untuk platform Java/Symbian
Keterangan		
 <pre> graph TD Start(()) -- "klik icon button Set" --> Picker[Menampilkan Date and Time Picker
+ entry / tanggal pengiriman pesan
+ entry / waktu pengiriman pesan] Picker -- "klik button Cancel" --> End((())) Picker -- "klik button Done" --> Decision{ } Decision -- "[menu edit date time schedule]" --> List[Menampilkan halaman list scheduled] Decision -- "[halaman edit all schedule]" --> EditAll[Menampilkan halaman Edit All Schedule
+ do / tampilkan tanggal dan waktu pengiriman pada Text field DateTime] List --> New[Menampilkan halaman New Schedule
+ do / tampilkan tanggal dan waktu pengiriman pada Text field DateTime] EditAll --> New </pre>		

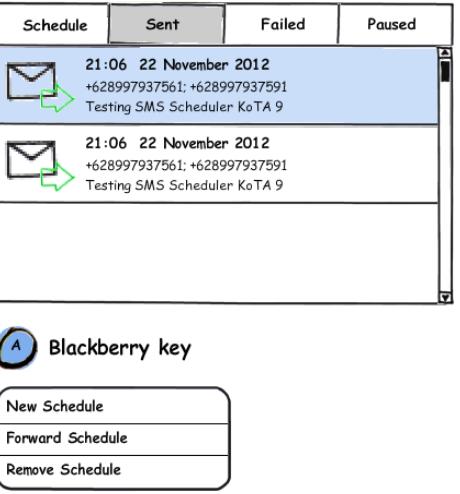
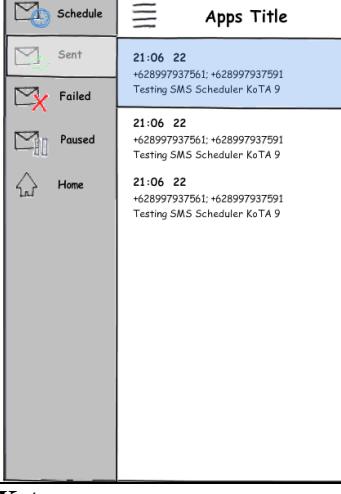
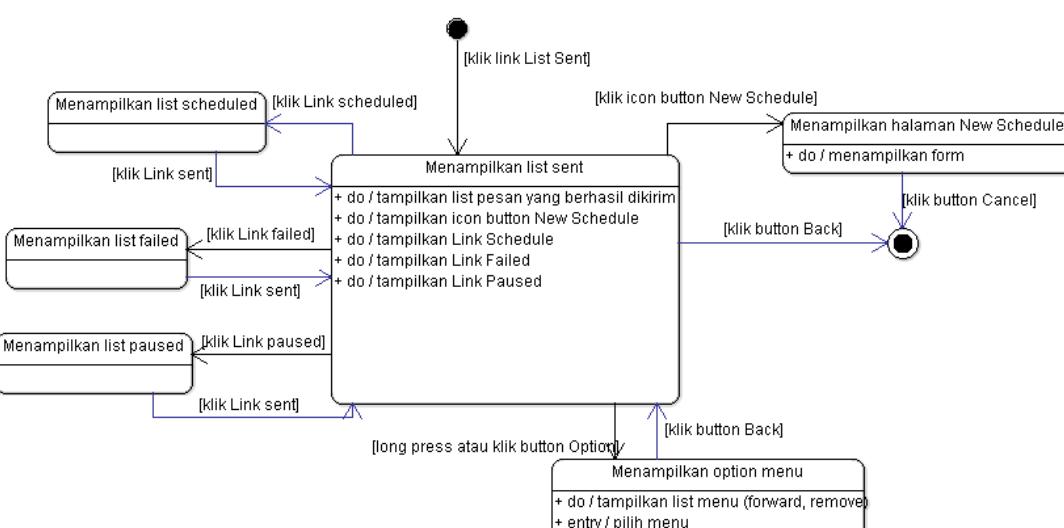
Tabel 130. Rancangan UI halaman menghapus *schedule*

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</p>			 <p>UI - 10.2 : UI menghapus <i>schedule</i> untuk platform Blackberry</p>
	 <p>UI - 10.3 : UI menghapus <i>schedule</i> untuk platform iOS</p>			 <p>UI - 10.4 : UI menghapus <i>schedule</i> untuk platform Java/Symbian</p>
Keterangan	<pre> graph TD Start(()) -- "[klik menu Remove schedule]" --> Step1[menampilkan halaman konfirmasi hapus + do / menampilkan alert box konfirmasi hapus] Step1 -- "[klik button Yes]" --> Step2[Menghapus schedule] Step2 --> Step3[Menampilkan halaman list scheduled] Step3 -- "[klik button No]" --> Step1 </pre>			

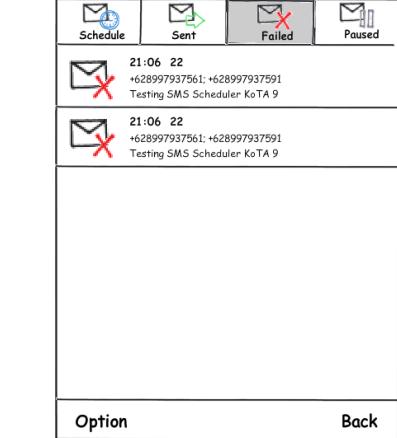
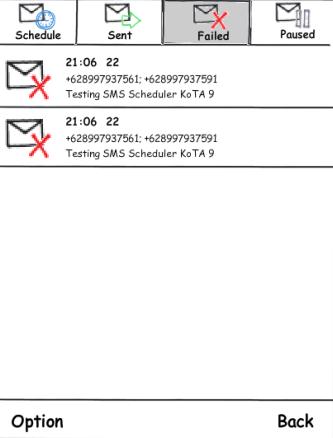
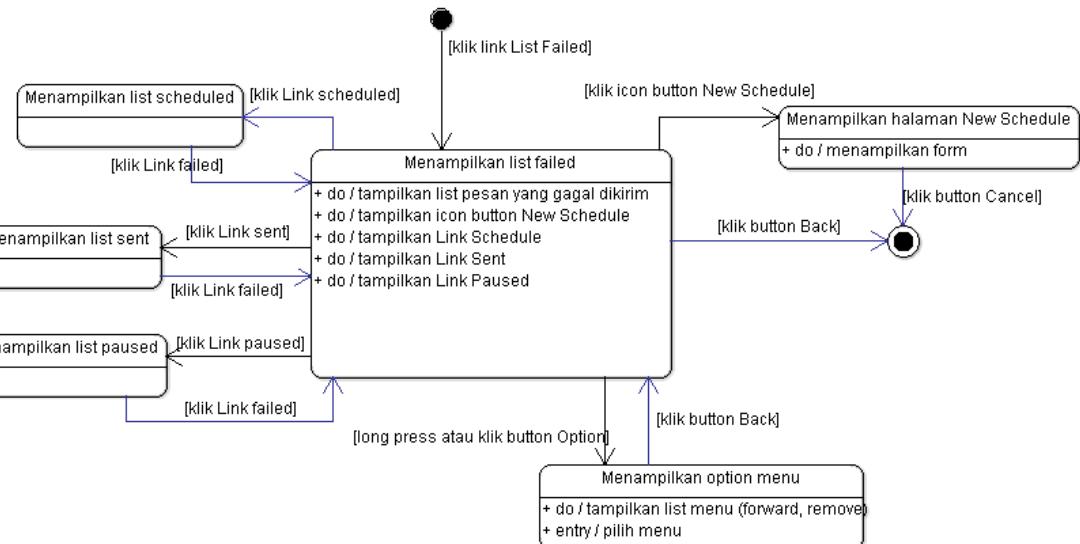
Tabel 131. Rancangan UI halaman kategori pesan (*List Schedule*)

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>UI - 11.3 : UI List Schedule untuk platform iOS</p>	 <p>UI - 11.4 : UI List Schedule untuk platform Java/Symbian</p>	
Keterangan :	 <pre> graph TD Start([Klik menu List Schedule]) --> LS[Menampilkan list scheduled] LS --> NS[Menampilkan halaman New Schedule] NS --> Option[Menampilkan option menu] Option --> Back1[long press atau klik button Option] Back1 --> LS LS --> Sent[Menampilkan list sent] LS --> Failed[Menampilkan list failed] LS --> Paused[Menampilkan list paused] Sent --> KLS1[Klik Link scheduled] Failed --> KLS2[Klik Link failed] Paused --> KLS3[Klik Link paused] KLS1 --> Edit[Menampilkan editing option] KLS2 --> Edit KLS3 --> Edit Edit --> Back2[long press atau klik button Back] Back2 --> LS Edit --> Cancel[Menampilkan halaman New Schedule] Cancel --> Back3[long press atau klik button Back] Back3 --> LS </pre>		

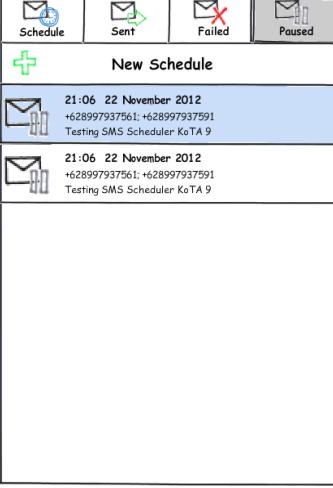
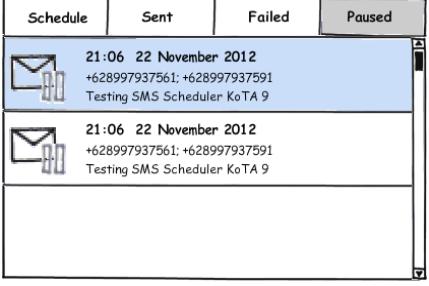
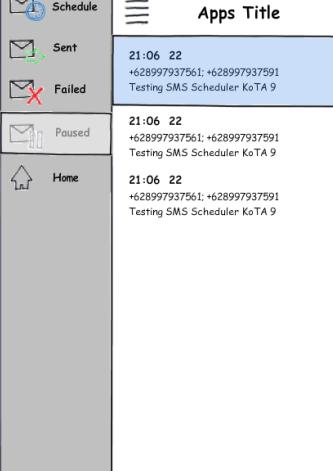
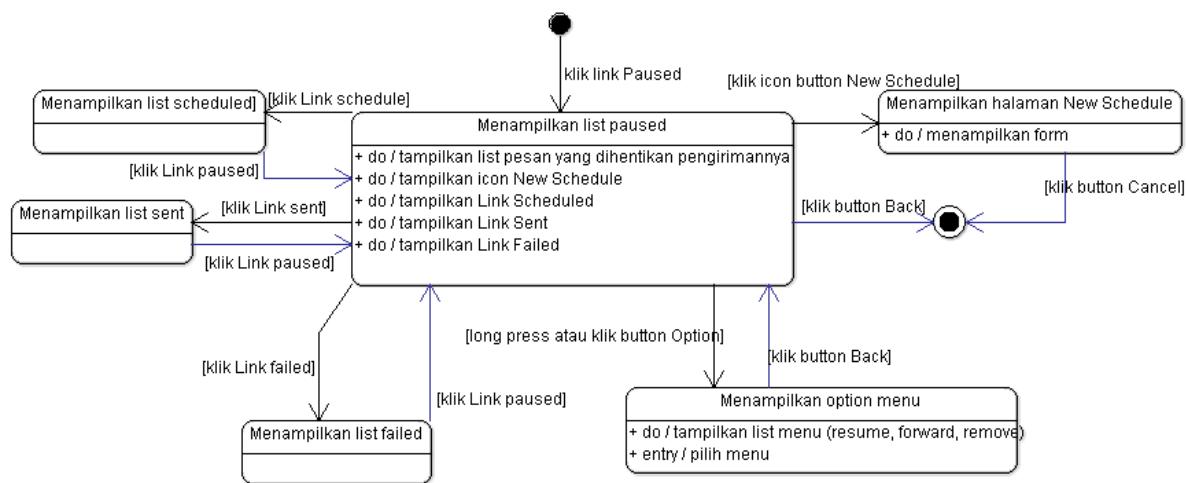
Tabel 132. Rancangan UI halaman kategori pesan (*List Sent*)

No.	UI - 12				
Nama	Kategori pesan (<i>List Sent</i>)				
Deskripsi	Halaman untuk menampilkan <i>list</i> dari pesan yang berhasil dikirim.				
Gambar	 <p>UI - 12.1 : UI <i>List Sent</i> untuk platform Android</p>				 <p>UI - 12.2 : UI <i>List Sent</i> untuk platform Blackberry</p>
	 <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>			
Keterangan	 <pre> graph TD Start(()) -- "[klik link List Sent]" --> LS[Menampilkan list sent] LS -- "[klik Link scheduled]" --> LS_Scheduled[Menampilkan list scheduled] LS -- "[klik Link failed]" --> LS_Failed[Menampilkan list failed] LS -- "[klik Link paused]" --> LS_Paused[Menampilkan list paused] LS -- "[klik icon button New Schedule]" --> NS[Menampilkan halaman New Schedule + do / menampilkan form] NS -- "[klik button Cancel]" --> End(()) LS -- "[klik button Back]" --> Back(()) Back -- "[long press atau klik button Option]" --> OM[Menampilkan option menu + do / tampilkan list menu (forward, remove) + entry / pilih menu] OM -- "[klik button Back]" --> Back </pre>				

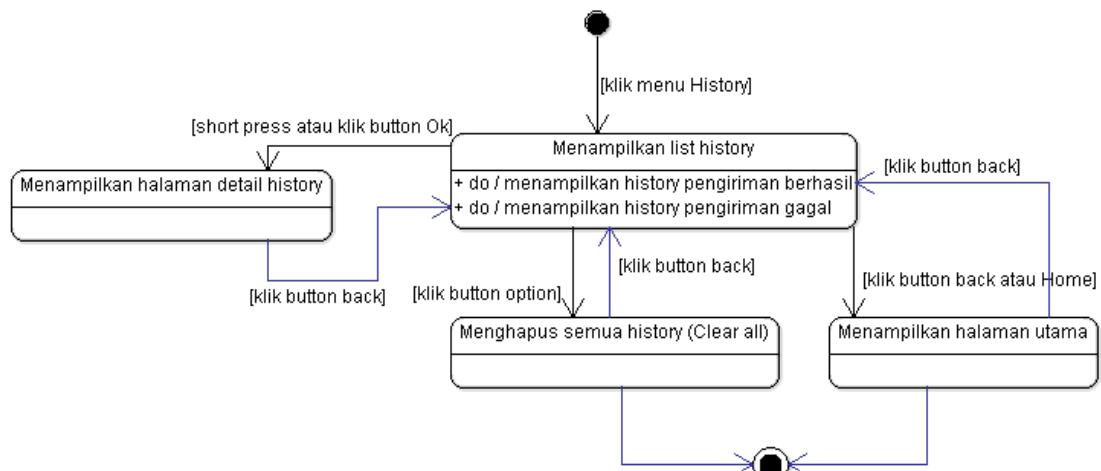
Tabel 133. Rancangan UI halaman kategori pesan (*List Failed*)

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>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>
Keterangan	 <pre> graph TD Start(()) -- "[klik link List Failed]" --> Main[Menampilkan list failed] Main -- "[klik Link failed]" --> Scheduled[Menampilkan list scheduled] Main -- "[klik Link failed]" --> Failed[Menampilkan list failed] Main -- "[klik Link failed]" --> Sent[Menampilkan list sent] Main -- "[klik Link failed]" --> Paused[Menampilkan list paused] Main -- "[klik icon button New Schedule]" --> NewSchedule[Menampilkan halaman New Schedule + do / menampilkan form] NewSchedule -- "[klik button Cancel]" --> Back(()) Back -- "[klik button Back]" --> Main Main -- "[long press atau klik button Option]" --> Option[Menampilkan option menu + do / tampilkan list menu (forward, remove) + entry / pilih menu] </pre>

Tabel 134. Rancangan UI halaman kategori pesan (*List Paused*)

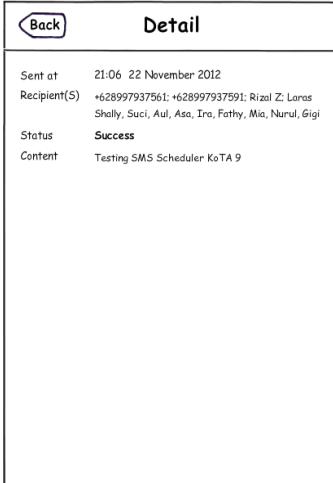
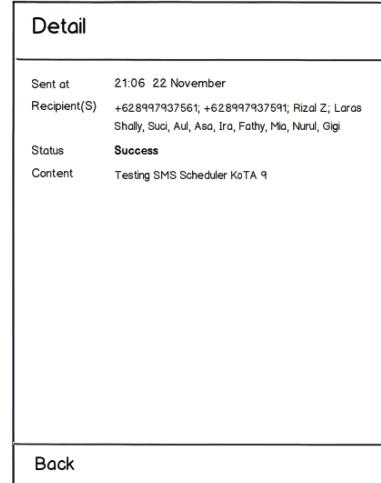
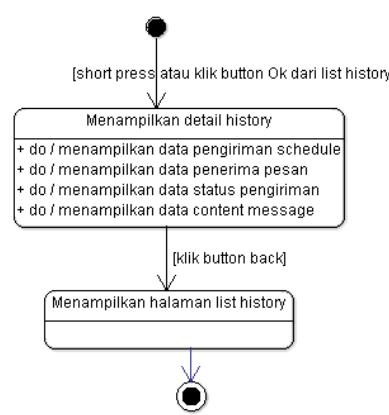
No.	UI - 14
Nama	Kategori pesan (<i>List Paused</i>)
Deskripsi	Halaman untuk menampilkan <i>list</i> dari <i>schedule-schedule</i> yang pengirimannya ditunda.
Gambar	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>UI - 14.1 : UI <i>List Paused</i> untuk platform Android</p> </div> <div style="text-align: center;">  <p>UI - 14.2 : UI <i>List Paused</i> untuk platform BlackBerry</p> </div> </div> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>UI - 14.3 : UI <i>List Paused</i> untuk platform iOS</p> </div> <div style="text-align: center;">  <p>UI - 14.4 : UI <i>List Paused</i> untuk platform Java/Symbian</p> </div> </div>
Keterangan	 <pre> graph TD Start(()) -- "klik link Paused" --> ListPaused[Menampilkan list paused] ListPaused -- "do / tampilkan list pesan yang dihentikan pengirimannya" --> ListPaused ListPaused -- "do / tampilkan icon New Schedule" --> NewSchedule[Menampilkan halaman New Schedule] ListPaused -- "do / tampilkan Link Scheduled" --> LinkScheduled[Menampilkan list scheduled] ListPaused -- "do / tampilkan Link Sent" --> LinkSent[Menampilkan list sent] ListPaused -- "do / tampilkan Link Failed" --> LinkFailed[Menampilkan list failed] NewSchedule -- "do / menampilkan form" --> Form[+ do / menampilkan form] Form -- "klik button Cancel" --> Back[Menampilkan halaman Back] Back -- "klik button Back" --> Start LinkScheduled -- "klik Link schedule" --> LinkSchedule[Menampilkan list scheduled] LinkSent -- "klik Link sent" --> LinkSent[Menampilkan list sent] LinkFailed -- "klik Link failed" --> LinkFailed[Menampilkan list failed] Option[Menampilkan option menu] -- "long press atau klik button Option" --> ListPaused Option -- "klik button Back" --> Back </pre>

Tabel 135. Rancangan UI halaman *history*

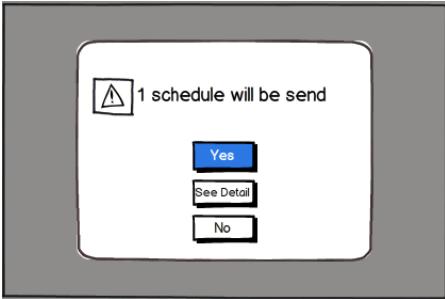
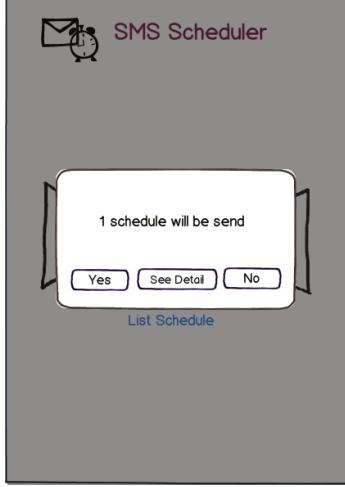
No.	UI - 15			
Nama	History			
Deskripsi	Halaman untuk menampilkan <i>list</i> dari pesan yang terkirim dan gagal terkirim.			
Gambar				
	 <p>A Option / more button on android Clear all</p>	<p>UI - 15.1 : UI History untuk platform Android</p>	 <p>A Blackberry key Clear all</p>	<p>UI - 15.2 : UI History untuk platform BlackBerry</p>
		<p>UI - 15.3 : UI History untuk platform iOS</p>		<p>UI - 15.4 : UI History untuk platform Java/Symbian</p>
Keterangan	 <pre> graph TD Start(()) --> ClickHistory[Click menu History] ClickHistory --> ShowHistory[Menampilkan list history] ShowHistory --> ViewDetails[Menampilkan halaman detail history] ViewDetails --> Back1[button back] ShowHistory --> ClearAll[do / menampilkan history pengiriman berhasil] ShowHistory --> ClearAllFailed[do / menampilkan history pengiriman gagal] ClearAll --> Back2[button back] ClearAllFailed --> Back2 Back2 --> MainScreen[Menampilkan halaman utama] Option[click button option] --> ClearAll </pre>			

Tabel 136. Rancangan UI halaman *detail history*

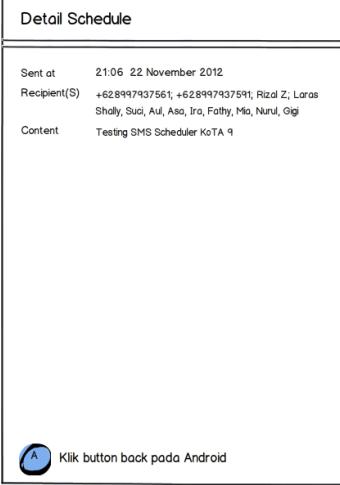
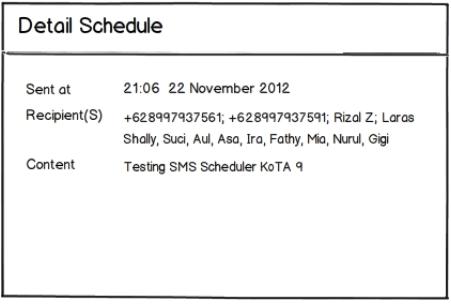
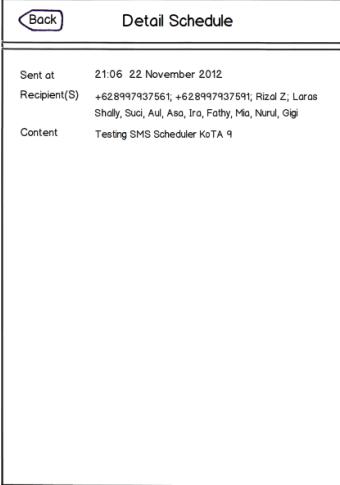
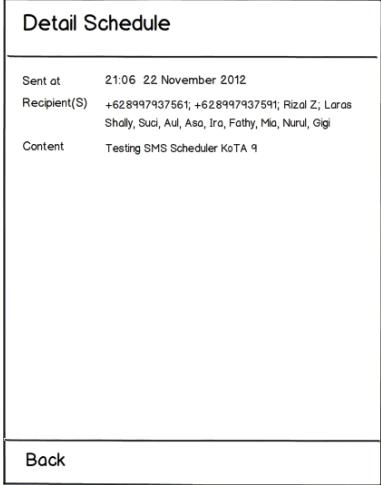
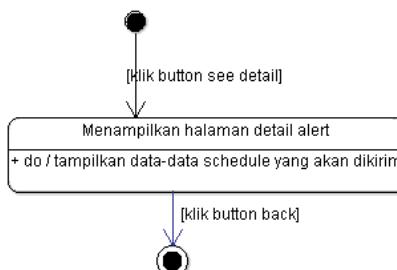
No.	UI - 16
Nama	Detail history

Deskripsi	Halaman untuk menampilkan <i>detail</i> dari <i>history</i> yang sebelumnya dipilih.		
Gambar			
	UI - 16.1 : UI Detail history untuk platform Android		UI - 16.2 : UI Detail history untuk platform Blackberry
	UI - 16.3 : UI Detail history untuk platform iOS		UI - 16.4 : UI Detail history untuk platform Java/Symbian
Keterangan	 <pre> graph TD Start(()) -- "[short press atau klik button Ok dari list history]" --> Detail[Menampilkan detail history
+ do / menampilkan data pengiriman schedule
+ do / menampilkan data penerima pesan
+ do / menampilkan data status pengiriman
+ do / menampilkan data content message] Detail -- "[klik button back]" --> List[Menampilkan halaman list history] List -- " " --> End(()) </pre>		

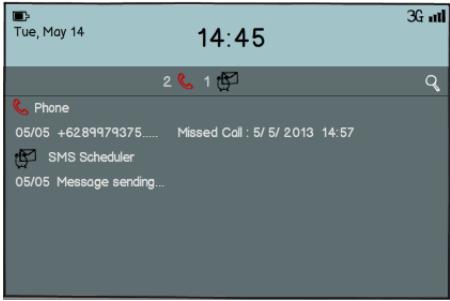
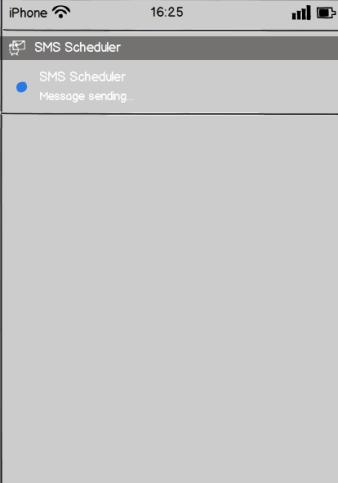
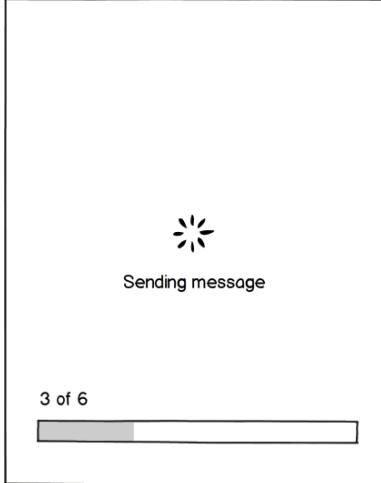
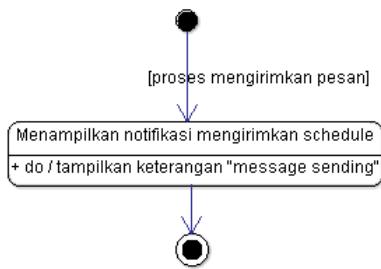
Tabel 137. Rancangan UI halaman Alert

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</p>
	 <p>UI - 17.2 : UI alert untuk platform Blackberry</p>
	 <p>UI - 17.3 : UI alert untuk platform iOS</p>
	 <p>UI - 17.4 : UI alert untuk platform Java/ Symbian</p>
Keterangan :	<pre> graph TD Start(()) --> A[Menampilkan alert box + do / tampilkan jumlah schedule yang akan dikirim] A -- "[klik button see detail]" --> B[Menampilkan halaman Detail Alert + do / tampilkan data schedule] A -- "[klik button Yes atau No]" --> C(()) C -- "[klik button back]" --> Start </pre>

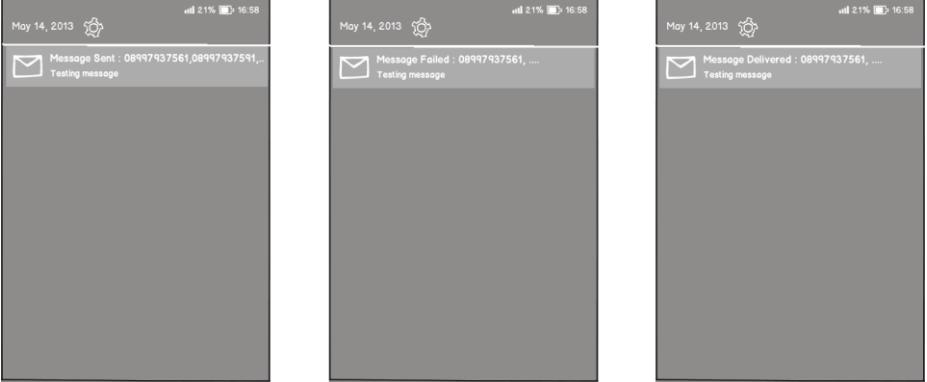
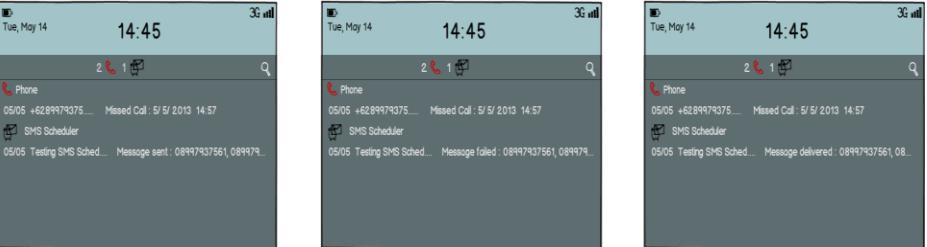
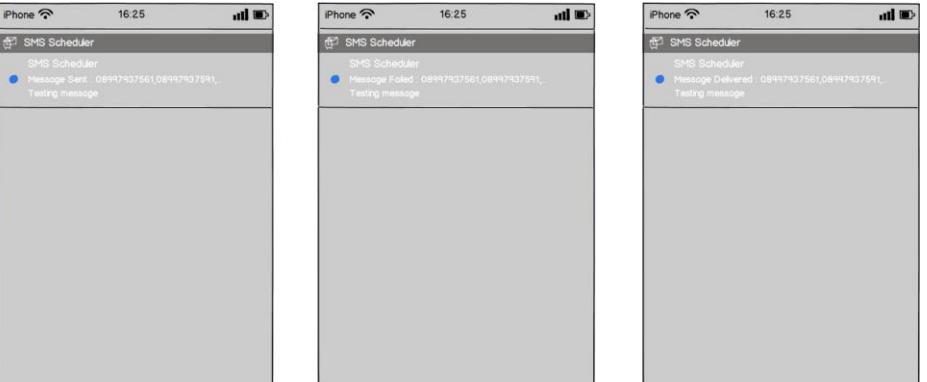
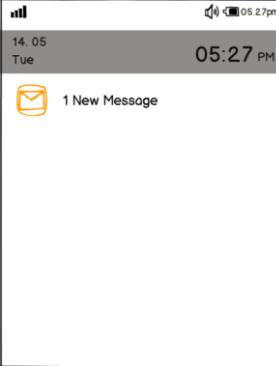
Tabel 138. Rancangan UI halaman Alert (*detail alert*)

No.	UI - 18		
Nama	<i>Alert (detail alert)</i>		
Deskripsi	Halaman yang menampilkan <i>detail schedule</i> yang akan dikirim.		
Gambar			
	 <p>Klik button back pada Android</p>	UI - 18.1 : UI alert (detail alert) untuk platform Android	 <p>UI - 18.2 : UI alert (detail alert) untuk platform BlackBerry</p>
		UI - 18.3 : UI alert (detail alert) untuk platform iOS	 <p>UI - 18.4 : UI alert (detail alert) untuk platform Java/Symbian</p>
Keterangan :	 <pre> graph TD A([Klik button see detail]) --> B[Menampilkan halaman detail alert
+ do / tampilkan data-data schedule yang akan dikirim] B --> C([Klik button back]) </pre>		

Tabel 139. 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>		
Keterangan :			

Tabel 140. 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 recipient	
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 141. Rancangan UI halaman *Notification (laporan pengiriman)* (Lanjutan)

No.	UI - 20
Nama	<i>Notification (laporan pengiriman)</i>
Keterangan :	

Tabel 142. 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	
<p>UI - 21.1 : UI <i>template</i> untuk <i>platform</i> Android</p>	<p>UI - 21.2 : UI <i>template</i> untuk <i>platform</i> Blackberry</p>
<p>UI - 21.3 : UI <i>template</i> untuk <i>platform</i> iOS</p>	<p>UI - 21.4 : UI <i>template</i> untuk <i>platform</i> Java/Symbian</p>

Tabel 143. Rancangan UI halaman *template* (Lanjutan)

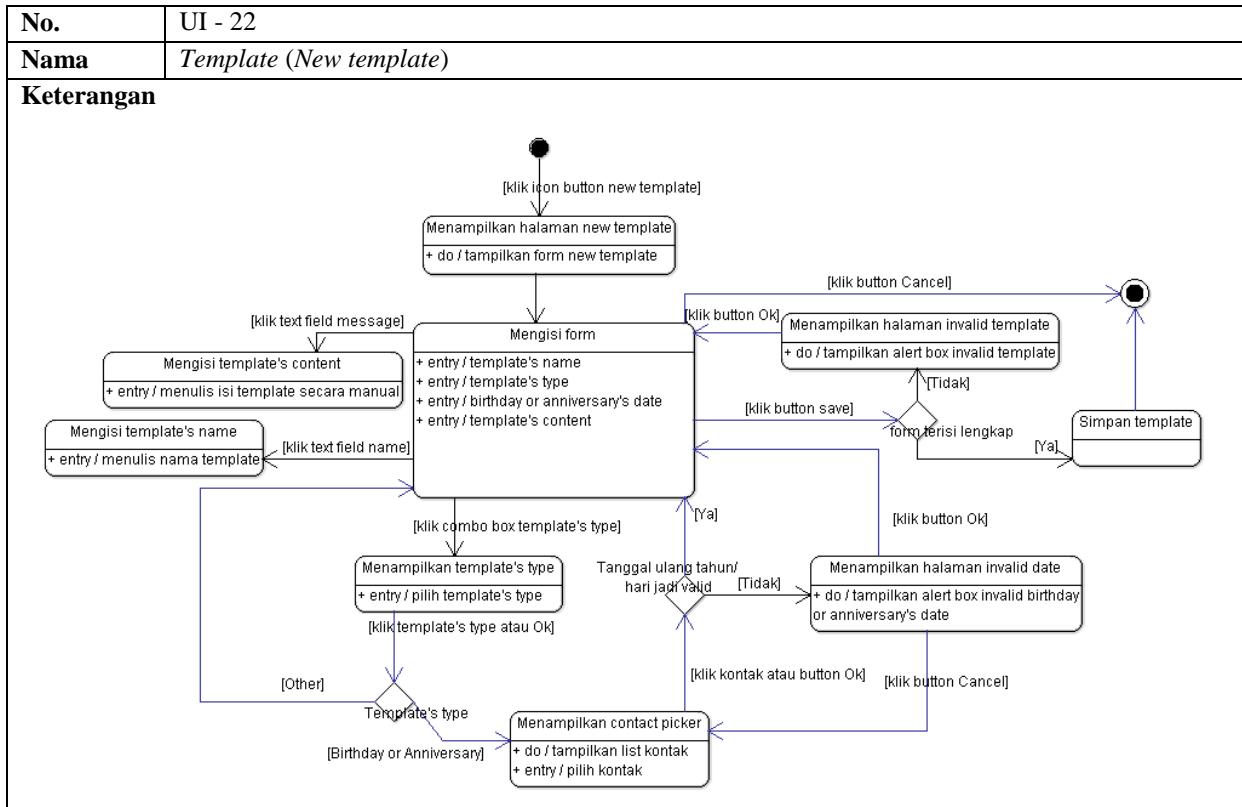
No.	UI - 21
Nama	<i>Template</i>
Keterangan	
<pre> graph TD Start(()) -- "[klik menu atau icon button template]" --> List[Menampilkan halaman list template
+ do / tampilkan template yang sudah dibuat
+ do / tampilkan icon button new template
+ entry / pilih template] List -- "[klik template atau button Done]" --> Message[Mengisi content message
+ entry / mengambil pesan dari template] Message -- "[klik button Back]" --> End(()) List -- "[klik icon button new template]" --> NewList[Menampilkan halaman new template
+ do / tampilkan form new template] </pre>	

Tabel 144. 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 style="text-align: center;">UI – 22.1 : UI new template untuk platform Android</p>	

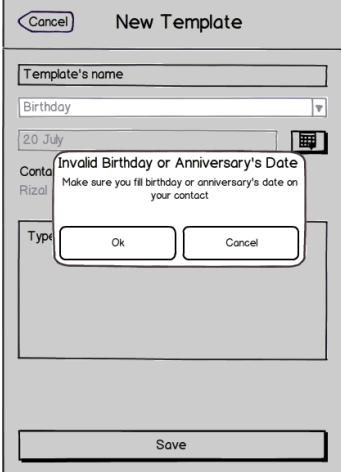
Tabel 145. Rancangan UI halaman Template (*New template*) (Lanjutan)

No.	UI - 22
Nama	<i>Template (New template)</i>
UI - 22.2 : UI new template untuk platform Blackberry	
	<p>Lanjutan dari layar pertama</p>
UI - 22.3 : UI new template untuk platform iOS	
UI - 22.4 : UI new template untuk platform Java/Symbian	

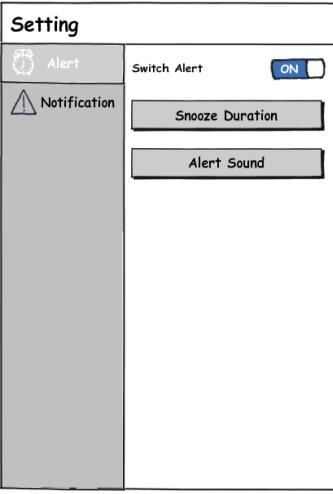
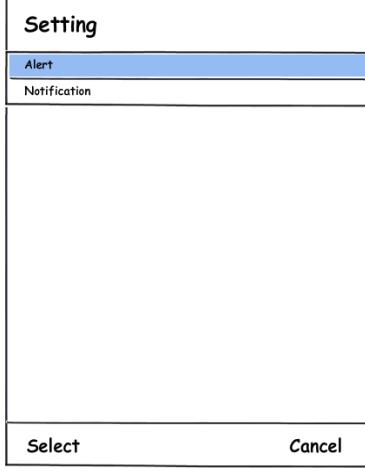
Tabel 146. Rancangan UI halaman Template (*New template*) (Lanjutan)Tabel 147. Rancangan UI halaman Template (*Invalid Date*)

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</p>	<p>UI - 23.2 : UI Set Data Count untuk platform BlackBerry</p>

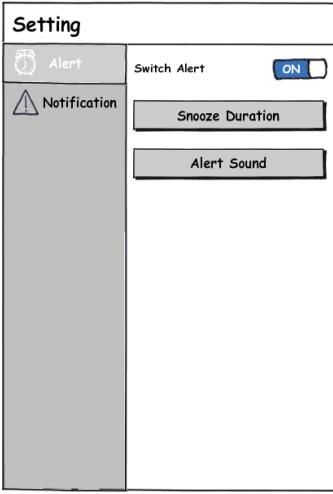
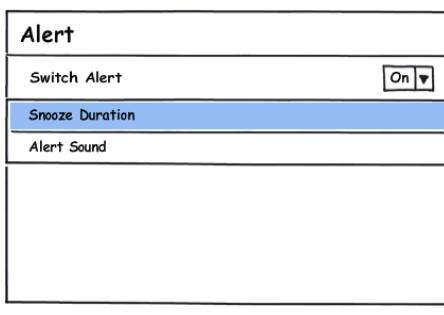
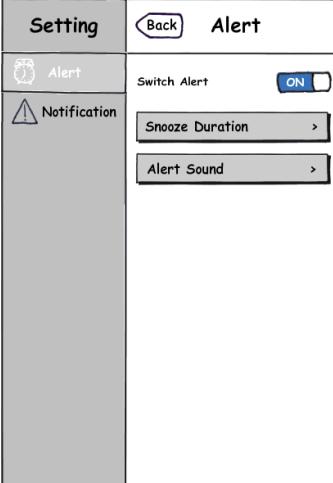
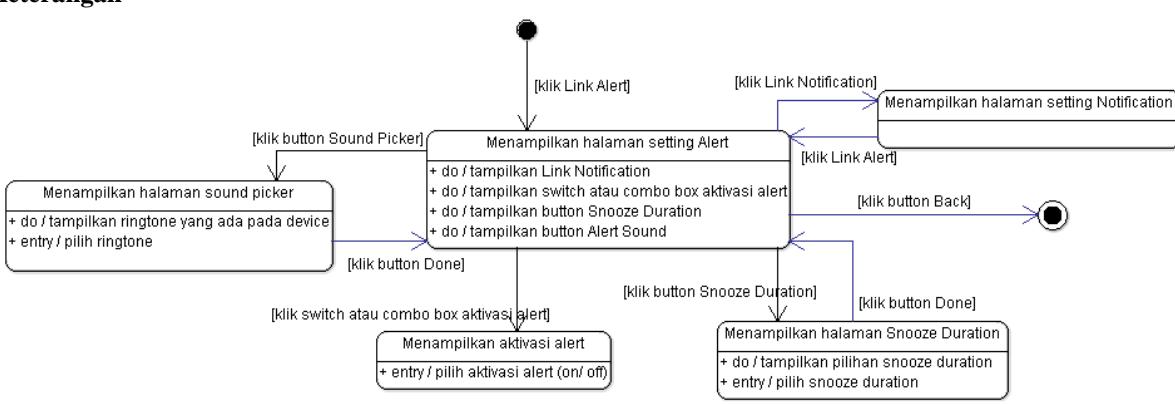
Tabel 148. Rancangan UI halaman *Template (Invalid Date)*

No.	UI - 23
Nama	<i>Template (Invalid Date)</i>
	 <p>UI - 23.3 : UI Set Data Count untuk platform iOS</p>
	 <p>UI - 23.4 : UI Set Data Count untuk platform Java/Symbian</p>
Keterangan	
<pre> [tanggal ulang tahun atau hari jadi tidak valid] ↓ Menampilkan halaman invalid date + do / tampilkan alert box invalid birthday or anniversary's date ↓ [klik button OK] ↓ Menampilkan halaman new template + do / tampilkan form new template ↓ [klik button Cancel] ↓ Menampilkan halaman contact picker + do / tampilkan list kontak + entry / pilih kontak ↓ [loop] </pre>	

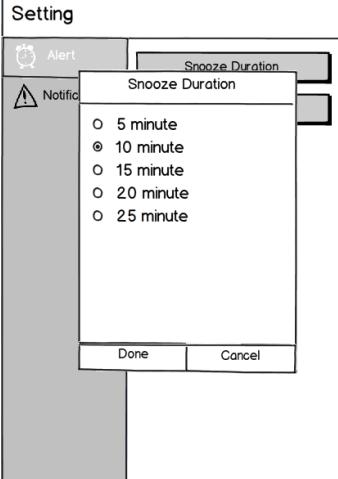
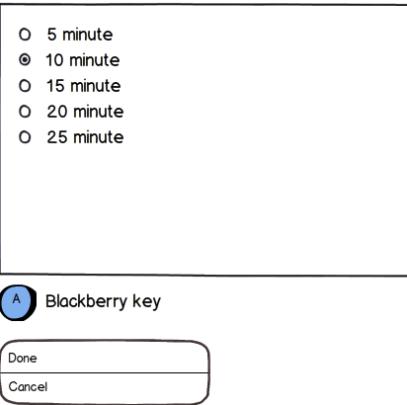
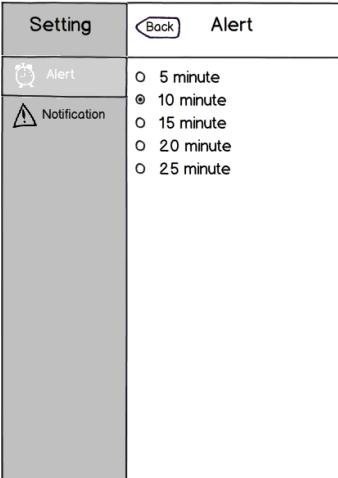
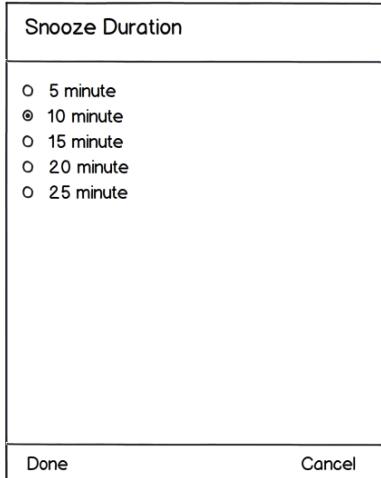
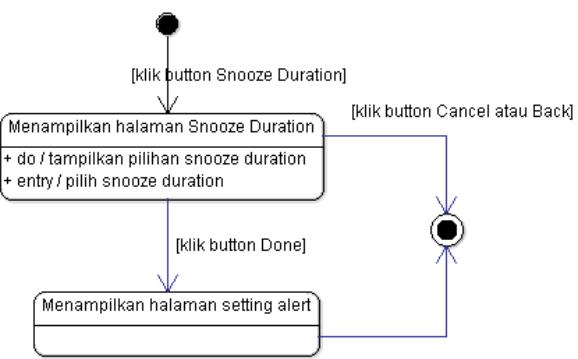
Tabel 149. Rancangan UI halaman *setting*

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 <i>platform</i> Android</p>  <p>UI - 24.2 : UI <i>setting</i> untuk <i>platform</i> Blackberry</p>
 <p>UI - 24.3 : UI <i>setting</i> untuk <i>platform</i> iOS</p>  <p>UI - 24.4 : UI <i>setting</i> untuk <i>platform</i> Java/Symbian</p>	
Keterangan	<pre> graph TD Start(()) -- "[klik menu Setting]" --> Decision{Pilih Link atau menu setting alert} Decision -- "[Ya]" --> Alert[Menampilkan halaman setting Alert] Alert --> End((())) Decision -- "[Tidak]" --> Notification[Menampilkan halaman setting Notification] </pre>

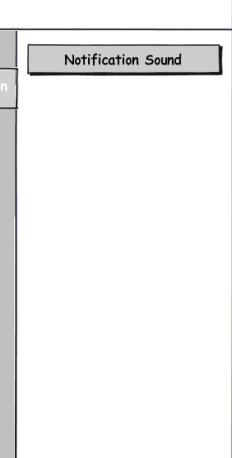
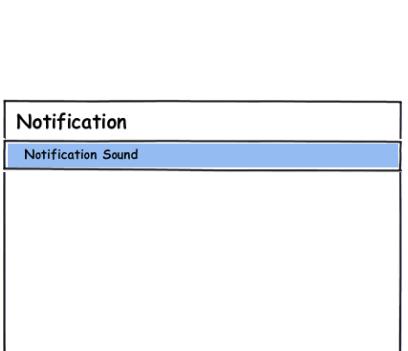
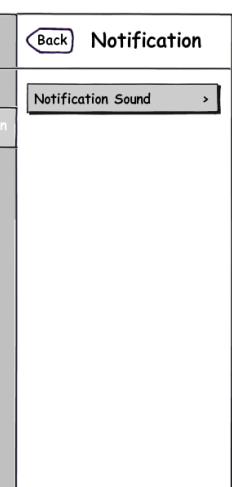
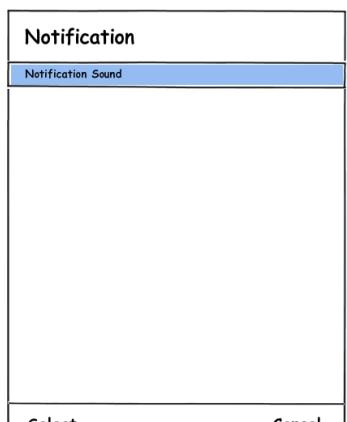
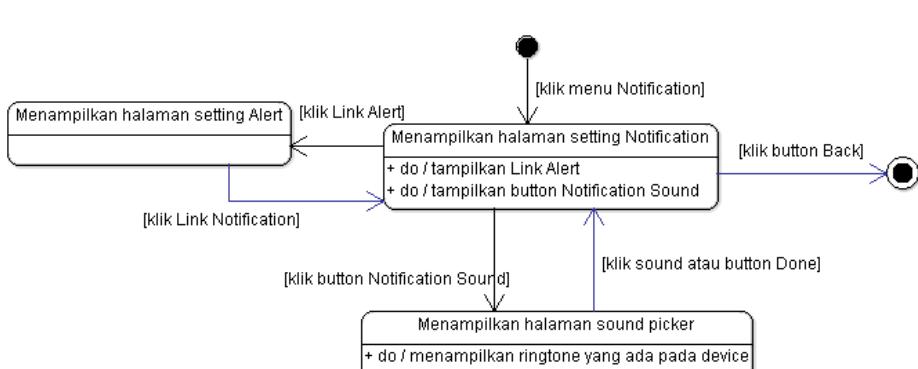
Tabel 150. Rancangan UI halaman *setting(alert)*

No.	UI - 25			
Nama	Setting(<i>alert</i>)			
Deskripsi	Halaman untuk melakukan pengaturan alert yang akan ditampilkan oleh aplikasi.			
Gambar	 <p>UI - 25.1 : UI setting (<i>alert</i>) untuk platform Android</p>		<p>UI - 25.2 : UI setting (<i>alert</i>) untuk platform Blackberry</p>	
	 <p>UI - 25.3 : UI setting (<i>alert</i>) untuk platform iOS</p>		<p>UI - 25.4 : UI setting (<i>alert</i>) untuk platform Java/Symbian</p>	
Keterangan	 <pre> graph TD Start(()) --> Main[Menampilkan halaman setting Alert] Main -- "[klik Link Alert]" --> SoundPicker[Menampilkan halaman sound picker] SoundPicker -- "[klik button Done]" --> Main SoundPicker -- "[klik button Sound Picker]" --> SoundPickerSub[+ do / tampilkan ringtone yang ada pada device + entry / pilih ringtone] SoundPickerSub -- "[klik button Done]" --> Main Main -- "[klik Link Notification]" --> Notification[Menampilkan halaman setting Notification] Notification -- "[klik Link Alert]" --> Main Notification -- "[klik button Back]" --> End(()) Main -- "[klik button Snooze Duration]" --> SnoozeDuration[Menampilkan halaman Snooze Duration + do / tampilkan pilihan snooze duration + entry / pilih snooze duration] SnoozeDuration -- "[klik button Done]" --> Main Main -- "[klik button Alert Sound]" --> AlertSound[Menampilkan aktivasi alert + entry / pilih aktivasi alert (on / off)] AlertSound -- "[klik button Done]" --> Main </pre>			

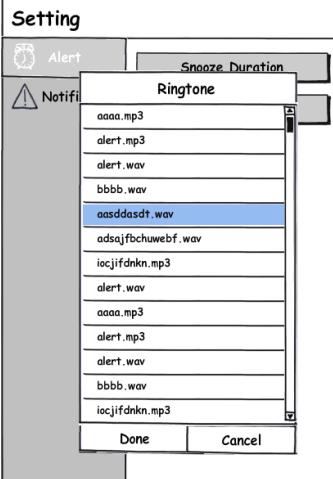
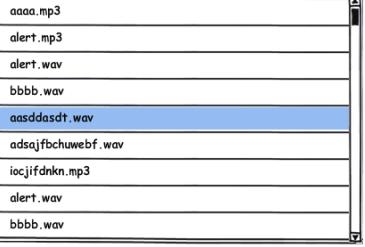
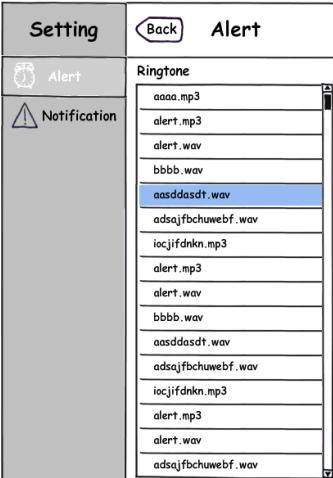
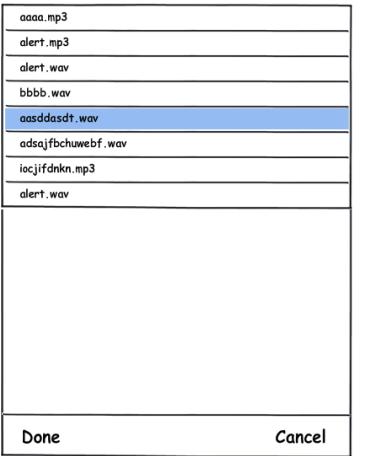
Tabel 151. Rancangan UI halaman *setting(Snooze duration)*

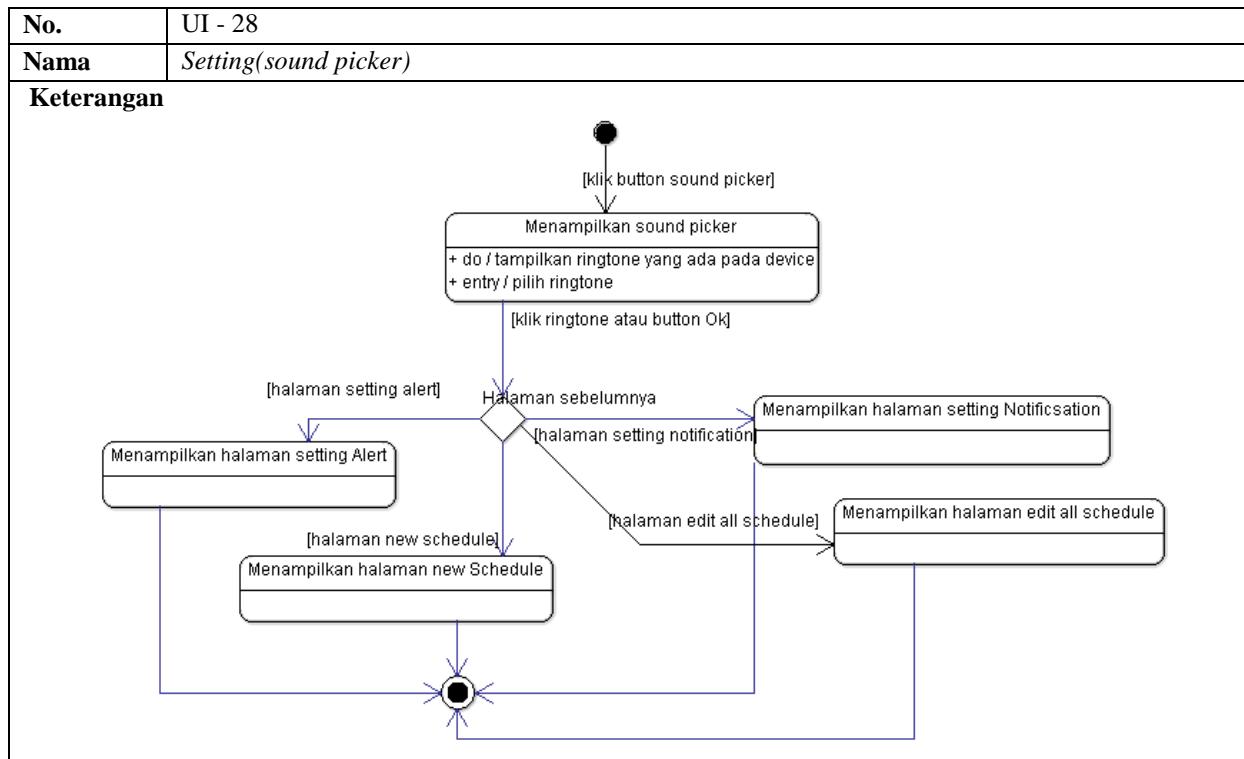
No.	UI - 26				
Nama	Setting(Snooze duration)				
Deskripsi	Halaman untuk melakukan pengaturan snooze duration alert.				
Gambar	 <p>UI - 26.1 : UI setting (Snooze duration) untuk platform Android</p>				 <p>UI - 26.2 : UI setting (Snooze duration) untuk platform BlackBerry</p>
	 <p>UI - 26.3 : UI setting (Snooze duration) untuk platform iOS</p>	 <p>UI - 26.4 : UI setting (Snooze duration) untuk platform Java/Symbian</p>			
Keterangan	 <pre> graph TD Start(()) -- "[klik button Snooze Duration]" --> SnoozeDuration[Menampilkan halaman Snooze Duration + do / tampilkan pilihan snooze duration + entry / pilih snooze duration] SnoozeDuration -- "[klik button Done]" --> AlertSettings[Menampilkan halaman setting alert] SnoozeDuration -- "[klik button Cancel atau Back]" --> Exit(()) </pre>				

Tabel 152. 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	 <pre> graph TD A[Menampilkan halaman setting Alert] -- "klik Link Alert" --> B[Menampilkan halaman setting Notification] B -- "do / tampilkan Link Alert" --> B B -- "do / tampilkan button Notification Sound" --> B B -- "klik button Back" --> C(()) B -- "klik Link Notification" --> A B -- "klik button Notification Sound" --> D[Menampilkan halaman sound picker] D -- "do / menampilkan ringtone yang ada pada device" --> D D -- "entry / pilih ringtone" --> D D -- "klik sound atau button Done" --> B </pre>

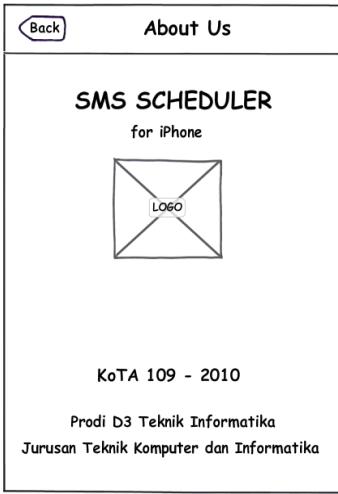
Tabel 153. Rancangan UI halaman *setting(sound picker)*

No.	UI - 28		
Nama	Setting(sound picker)		
Deskripsi	Halaman untuk memilih nada dering alert atau notification.		
Gambar	 <p>UI - 28.1 : UI setting (sound picker) untuk platform Android</p>	 <p>UI - 28.2 : UI setting (sound picker) untuk platform Blackberry</p>	
	 <p>UI - 28.3 : UI setting (sound picker) untuk platform iOS</p>	 <p>UI - 28.4 : UI setting (sound picker) untuk platform Java/Symbian</p>	

Tabel 154. Rancangan UI halaman *setting(sound picker)* (Lanjutan)Tabel 155. 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 platform Android</p> <p>UI - 29.2 : UI <i>about us</i> untuk platform Blackberry</p>

Tabel 156. Rancangan UI halaman *about us* (Lanjutan)

No.	UI - 29
Nama	<i>About us</i>
Gambar	 <p>UI - 29.3 : UI <i>about us</i> untuk platform iOS</p>  <p>UI - 29.4 : UI <i>about us</i> untuk platform Java/Symbian</p>
Keterangan	<pre> graph TD Start(()) --> A[Menampilkan halaman about us
+ do / tampilkan data untuk about us] A -- "[klik menu about us]" --> B[Menampilkan halaman utama] B -- "[klik button back atau Ok]" --> End((())) </pre>

BAB V

Implementasi

Tahap implementasi bertujuan untuk merepresentasikan rancangan yang telah dibuat ke dalam bentuk program dengan mengikuti dan menerapkan sesuai dengan yang dirancang pada tahap perancangan. Pada bab ini akan dideskripsikan mengenai implementasi untuk aplikasi SMS *Scheduler* yang dilakukan berdasarkan rancangan yang telah dibuat pada tahap sebelumnya, yaitu implementasi untuk hasil perancangan data, perancangan dari requirement yang telah terdefinisi (fitur sistem) dan perancangan *user interface*.

5.1 Batasan Implementasi

Dikarenakan pembuatan aplikasi ini dilakukan dengan metode *increment*, maka untuk saat ini implementasi aplikasi dibatasi hanya sampai implementasi pada *increment* kedua, yaitu sudah tersedianya fitur untuk membuat *schedule* baru dan kemudian aplikasi dapat mengirimkan pesan sesuai dengan *schedule* tersebut, serta menampilkan notifikasi ketika pesan berhasil dikirim maupun ketika pesan gagal dikirim.

5.2 Implementasi Data

Dari perancangan data yang telah dibuat pada subbab 4.2, terdapat beberapa kebutuhan data yang telah terimplementasi. Tabel 74 mendefinisikan kebutuhan data yang telah terimplementasi.

Tabel 157. Tabel status implementasi data

Nama Tabel	Dekripsi	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 character</i> .	√
<i>Defined_Character</i>	Tabel ini digunakan untuk menyimpan data tanggal,	√

Nama Tabel	Deksripsi	Status Implementasi
	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.	

Keterangan :

√ : sudah terimplementasi

x : tidak terimplementasi

5.3 Implementasi *Requirement*

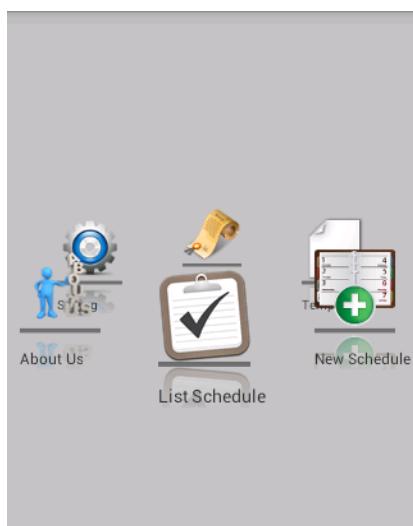
Pada bagian ini dijelaskan mengenai status implementasi dari *requirement* untuk aplikasi SMS *Scheduler* yang telah terdefinisi pada SRS aplikasi SMS *Scheduler* subbab 2.2, dan dijelaskan pada lampiran C.

5.4 Implementasi Fitur Aplikasi/ Sistem

Pada bagian ini akan dijelaskan mengenai hasil implementasi untuk rancangan fitur aplikasi/ sistem yang telah dibuat pada bab sebelumnya.

5.4.1 Fitur *New Schedule*

5.4.1.1 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

Gambar 29 . Screenshot Home Page

5.4.1.2 Screenshot New Schedule (Form New Schedule)

New Schedule

Recipient(s)

Date and time

Once

Stop after : 1

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* gambar 30.

Gambar 30 . Screenshot New Schedule (Form New Schedule)

5.4.1.3 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

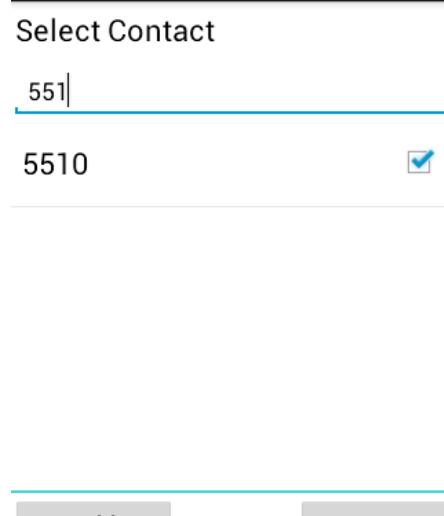
Select Contact

Search

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

Add CANCEL

(a) (b)

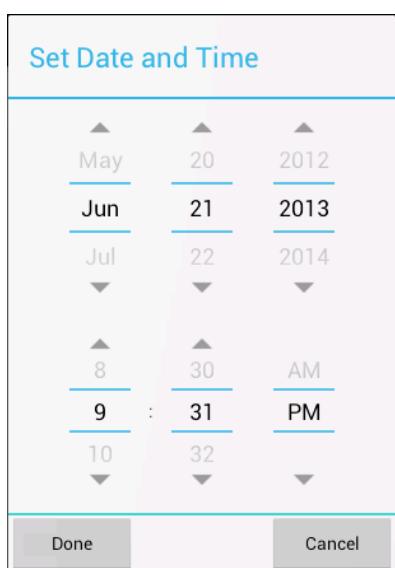


(c)

Gambar 31 . Screenshot New Schedule (Contact Picker)

Ketika *user* menginginkan kontak *recipient* diambil dari kontak yang ada pada *device*, maka akan ditampilkan halaman *contact picker* seperti yang digambarkan pada gambar 31. 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*.

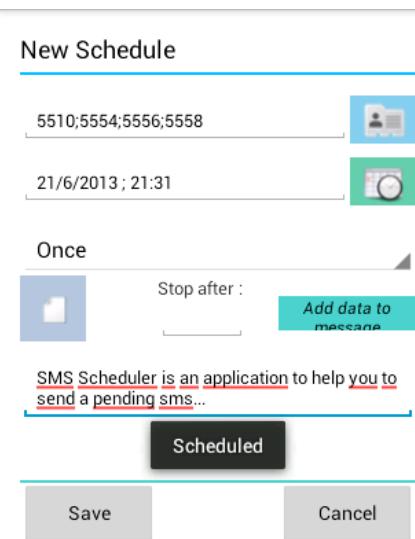
5.4.1.4 Screenshot New Schedule (Menampilkan Pengaturan Tanggal dan Waktu)



Untuk mengatur tanggal dan waktu pengiriman pesan, ditampilkan *date and time picker* seperti yang digambarkan pada gambar 32.

Gambar 32 . Screenshot New Schedule (Menampilkan Pengaturan Tanggal dan Waktu)

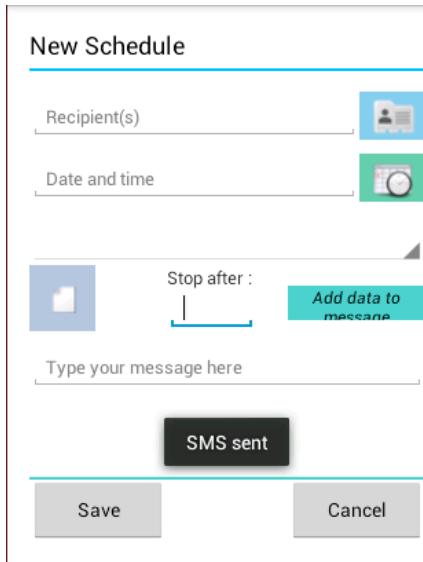
5.4.1.5 Screenshot Save Schedule



Screenshot pada gambar 33 menggambarkan permberitahuan pembuatan *schedule*, yaitu bahwa *schedule* telah berhasil dibuat.

Gambar 33 . Screenshot Save Schedule

5.4.1.6 Screenshot Notification



Screenshot pada gambar 34 menggambarkan permberitahuan pengiriman pesan (memberikan informasi bahwa pesan dari *schedule* yang dibuat telah berhasil dikirim atau gagal dikirim).

Gambar 34 . Screenshot Notification

BAB VI

Pengujian

Tahap pengujian bertujuan untuk mengetahui apakah aplikasi yang dibuat telah memenuhi requirement yang telah didefinisikan sebelumnya atau tidak. Pengujian dilakukan dengan metode *blackbox testing*.

6.1 Batasan Implementasi

Untuk saat ini, pengujian dilakukan terhadap fitur membuat *schedule*, mengirimkan *schedule* sampai dengan menampilkan notifikasi.

6.2 Perencanaan Pengujian

Dari gambaran umum pengujian (*test*) yang sebelumnya telah dibuat pada jadwal *increment* di Bab 1 (Metodologi Pengembangan Aplikasi), didapatkan *test case* yang dituliskan pada tabel 158 - 160.

Tabel 158. *Test Case-1*

No. Pengujian		1
Nama Pengujian		Membuat <i>schedule</i> baru
No	Masukan	Ekspektasi Keluaran
1	Klik menu <i>new schedule</i>	Aplikasi menampilkan halaman <i>new schedule</i> (form <i>new schedule</i>)
2	Klik button <i>contact</i>	Aplikasi menampilkan list <i>contact</i>
3	Klik button <i>datetime</i>	Aplikasi menampilkan date time picker
4	Klik <i>combo box frequency</i>	Aplikasi menampilkan pilihan <i>frequency</i>
5	Klik button <i>template</i>	Aplikasi menampilkan halaman <i>template</i> (list <i>template</i>)
6	Klik <i>enter</i>	Focus kursor berpindah ke <i>edittext</i> atau button selanjutnya.
7	User menginputkan satu atau lebih nomor recipient secara manual dari keyboard/ keypad device	Aplikasi menampung nomor recipient yang diinput user
8	User menginputkan satu atau lebih nomor recipient secara manual dari keyboard/ keypad device, dimana inputannya menggunakan karakter “+” dan “;”	Aplikasi menampung nomor recipient yang diinput user, termasuk karakter “+” dan “;”
9	User mengambil satu atau lebih nomor recipient dari kontak device	Aplikasi menampung nomor recipient yang diinput user
10	User menginputkan waktu pengiriman yang sama dengan waktu sistem saat itu	Aplikasi menampilkan pesan <i>error</i> bahwa waktu yang diinputkan sudah lampau
11	User menginputkan waktu pengiriman yang sudah lampau	Aplikasi menampilkan pesan <i>error</i> bahwa waktu yang diinputkan sudah lampau
12	User memilih <i>template</i>	Aplikasi menampilkan isi <i>template</i> pada <i>field content message</i>
13	User tidak mengisi semua <i>field</i> pada <i>form</i>	Aplikasi menampilkan pesan <i>error</i> bahwa form masih belum lengkap
14	Klik button <i>save</i>	Aplikasi menjalankan proses validasi dan

No. Pengujian	1	
Nama Pengujian	Membuat <i>schedule</i> baru	
No	Masukan	
15	Klik button <i>cancel</i>	Ekspektasi Keluaran
		penyimpanan <i>schedule</i> ke <i>database</i>
		Aplikasi kembali ke halaman utama

Tabel 159. *Test Case-2*

No. Pengujian	2	
Nama Pengujian	Mengirimkan pesan	
No	Masukan	
1	User membuat <i>schedule</i>	Ekspektasi Keluaran
		Aplikasi mengirimkan pesan sesuai dengan <i>schedule</i> yang tersimpan di <i>database</i>
2	User membuat beberapa <i>schedule</i> berbeda dengan waktu pengiriman yang sama	Aplikasi mengirimkan pesan sesuai dengan <i>schedule</i> yang tersimpan di <i>database</i> berdasarkan urutan disimpannya (konsep FIFO)

Tabel 160. *Test Case-3*

No. Pengujian	3	
Nama Pengujian	Menampilkan notifikasi	
No	Masukan	
1	User membuat <i>schedule</i>	Ekspektasi Keluaran
		Aplikasi menampilkan notifikasi pesan berhasil dikirim
2	User membuat <i>schedule</i>	Aplikasi menampilkan notifikasi pesan gagal dikirim

6.3 Hasil Pengujian

Berdasarkan *test case* yang terdapat pada sub bab perencanaan pengujian, maka berikut ini merupakan hasil pengujian yang telah dilakukan pada setiap *test case*.

Tabel 161. Hasil pengujian

No. Test Case	No	Masukan	Keluaran	Status Pengujian
Membuat <i>schedule</i> baru (Test case-1)	1	Klik menu <i>new schedule</i>	Aplikasi menampilkan halaman new <i>schedule</i> (form new <i>schedule</i>)	Berhasil
	2	Klik button <i>contact</i>	Aplikasi menampilkan contact picker	Berhasil
	3	Klik button <i>datetime</i>	Aplikasi menampilkan date time picker	Berhasil
	4	Klik combo box <i>frequency</i>	Aplikasi menampilkan pilihan frequency	Berhasil
	5	Klik button <i>template</i>	Aplikasi menampilkan halaman template (list template)	Belum dilakukan pengujian
	6	Klik <i>enter</i>	Focus kurSOR berpindah ke edittext atau button selanjutnya.	Belum dilakukan pengujian
	7	User menginputkan satu atau lebih nomor recipient secara manual dari keyboard/ keypad device	Aplikasi menampung nomor recipient yang dipilih user	Berhasil
	8	User menginputkan satu atau lebih nomor recipient	Aplikasi menampung nomor recipient yang diinput user,	Berhasil

No. Test Case	No	Masukan	Keluaran	Status Pengujian
		secara manual dari keyboard/ keypad device, dimana inputannya menggunakan karakter “+” dan “;”	termasuk karakter “+” dan “;”	
	9	User mengambil satu atau lebih nomor recipient dari kontak device	Aplikasi menampung nomor recipient yang diinput user	Berhasil
	10	User menginputkan waktu pengiriman yang sama dengan waktu sistem saat itu	Aplikasi menampilkan pesan <i>error</i> bahwa waktu yang diinputkan sudah lampau	Berhasil
	11	User menginputkan waktu pengiriman yang sudah lampau	Aplikasi menampilkan pesan <i>error</i> bahwa waktu yang diinputkan sudah lampau	Berhasil
	12	<i>User memilih template</i>	Aplikasi menampilkan isi <i>template</i> pada <i>field content message</i>	Belum dilakukan pengujian
	13	<i>User tidak mengisi semua field pada form</i>	Aplikasi menampilkan pesan <i>error</i> bahwa form masih belum lengkap	Belum dilakukan pengujian
	14	Klik <i>button save</i>	Aplikasi menjalankan proses validasi dan penyimpanan <i>schedule</i> ke <i>database</i>	Berhasil
	15	Klik <i>button cancel</i>	Aplikasi kembali ke halaman utama	Berhasil
Mengirimkan <i>schedule</i> (Test case-2)	1	<i>User membuat schedule</i>	Aplikasi mengirimkan <i>schedule</i> sesuai waktu yang ditentukan user	Berhasil
	2	User membuat beberapa <i>schedule</i> berbeda dengan waktu pengiriman yang sama	Aplikasi mengirimkan <i>schedule</i> sesuai dengan konsep FIFO	Berhasil
Menampilkan notifikasi (Test case-3)	1	<i>User membuat schedule</i>	Aplikasi menampilkan notifikasi pesan berhasil dikirim	Berhasil
	2	<i>User membuat schedule</i>	Aplikasi menampilkan notifikasi pesan gagal dikirim	Berhasil

DAFTAR PUSTAKA

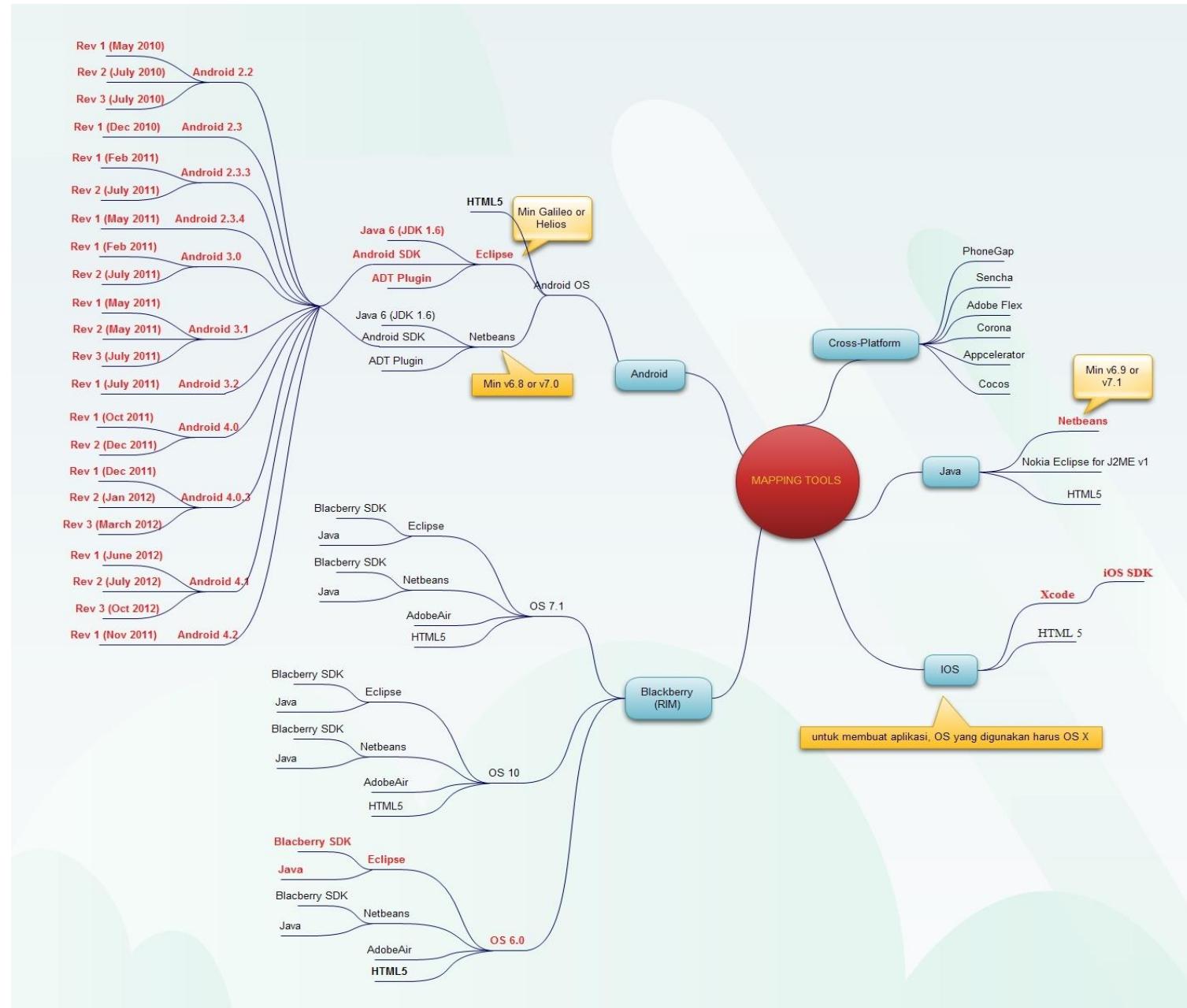
- Alleman, Glen B. (2005). What is the Purpose of a Schedule ?. Diakses pada tanggal 11 Juni 2013 dari http://herdingcats.typepad.com/my_weblog/2005/11/what_is_the_pur.html.
- Charland, Ade. (2011). *Mobile Application Development : Web vs Native*. Diakses pada tanggal 24 Desember 2012 dari <http://www.creativebrain.web.id/media.php?action=readnews&id=315&title=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_tiandi.staff.gunadarma.ac.id/Downloads/files/20647/1_Pengenalan_Java.pdf.
- 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.
- SmartCode. (2009). Timed SMS Scheduler 1.0. Diakses pada tanggal 31 Mei 2013 dari <http://timed-sms-scheduler.smartcode.com/info.html>.
- 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 Requirement	Deksripsi	Status Implementasi
[REQ F - 01]	Aplikasi menyediakan fitur untuk melakukan pengelolaan <i>schedule</i> .	
[REQ F - 01.1]	Aplikasi menyediakan fitur untuk membuat jadwal 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 - 01.2]	Aplikasi menyediakan fitur untuk menghapus jadwal 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 - 01.3]	Aplikasi menyediakan fitur untuk mengubah/ mengedit jadwal 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 menerima input nomor <i>recipient</i> secara manual.	
[REQ UPD - 03]	Aplikasi dapat menerima input nomor <i>recipient</i> dari nomor kontak pada <i>device</i> .	
[REQ UPD - 04]	Aplikasi dapat menerima input nomor <i>recipient</i> dari grup kontak pada <i>device</i> .	
[REQ UPD - 05]	Aplikasi dapat menggunakan <i>multiple recipient</i> .	
[REQ UPD - 06]	Aplikasi dapat memvalidasi <i>input</i> waktu pengiriman.	
[REQ UPD - 07]	Aplikasi dapat menampilkan pesan <i>error</i> jika waktu pengiriman yang diinputkan adalah waktu lampau.	
[REQ UPD - 08]	Aplikasi dapat menyediakan pilihan frekuensi pengiriman.	
[REQ UPD - 09]	Aplikasi dapat menggunakan <i>template</i> yang sudah dibuat.	
[REQ UPD - 10]	Aplikasi dapat menyimpan <i>schedule</i> yang diubah ke <i>database</i> .	

Nomor Requirement	Dekripsi	Status Implementasi
[REQ UPD - 11]	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 - 01.4]	Aplikasi menyediakan fitur untuk menunda jadwal pengiriman pesan.	
[REQ PSD - 01]	Aplikasi dapat Menghentikan sementara <i>schedule</i> pengiriman pesan.	
[REQ PSD - 02]	Aplikasi dapat melanjutkan (<i>resume</i>) <i>schedule</i> pengiriman pesan.	
[REQ PSD - 03]	Aplikasi dapat melakukan validasi waktu untuk <i>schedule</i> pengiriman pesan yang di- <i>resume</i> .	
[REQ PSD - 04]	Aplikasi dapat menampilkan pesan <i>error</i> jika proses <i>resume</i> gagal.	
[REQ F - 01.5]	Aplikasi menyediakan fitur untuk dapat mengelompokkan jadwal pengiriman pesan sesuai dengan kategorinya.	
[REQ FOL - 01]	Aplikasi dapat mengambil pesan untuk kategori <i>schedule</i> (pesan yang sedang <i>dischedulekan</i> dan belum dikirim).	
[REQ FOL - 02]	Aplikasi dapat mengambil pesan untuk 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).	
[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 - 02]	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 - 03]	Aplikasi menyediakan fitur <i>alert/ pop up</i> .	
[REQ ALR - 01]	Aplikasi dapat menampilkan <i>pop up</i> beberapa waktu sebelum pengiriman <i>schedule</i> , sesuai <i>setting</i> waktu yang ditentukan <i>user</i> .	
[REQ ALR - 02]	Aplikasi dapat menampilkan <i>alert/ pop up</i> secara berkala setiap 5 menit sekali jika <i>user</i> tidak menanggapi <i>alert</i> yang muncul.	

Nomor Requirement	Dekripsi	Status Implementasi
[REQ ALR - 03]	Aplikasi mampu menyertakan pilihan nada yang ada di ponsel <i>user</i> ketika <i>pop up</i> muncul.	
[REQ ALR - 04]	Aplikasi dapat menampilkan <i>detail</i> dari <i>schedule</i> yang akan dikirim.	
[REQ ALR - 05]	Aplikasi dapat menghentikan/ membatalkan pengiriman <i>schedule</i> .	
[REQ ALR - 06]	Aplikasi dapat meneruskan proses pengiriman <i>schedule</i> .	
[REQ F - 04]	Aplikasi menyediakan fitur notifikasi jadwal	
[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 - 05]	Aplikasi menyediakan fitur <i>setting</i> .	
[REQ F - 05.1]	Aplikasi menyediakan fitur untuk mengatur pengaturan alert sesuai dengan keinginan <i>user</i> .	
[REQ SA - 01]	Aplikasi menyediakan pilihan untuk mengaktifkan <i>alert</i> dan untuk menon-aktifkan <i>alert</i> .	
[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 - 05.2]	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.	
[REQ F - 06]	Aplikasi menyediakan fitur <i>typical message/ template</i>	
[REQ F - 07]	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> .	

Nomor <i>Requirement</i>	Deksripsi	Status Implementasi
[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 TMP - 08]	Aplikasi dapat menampilkan <i>form</i> pembuatan <i>template</i> baru.	

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 bantuannya 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